ON FLEXIBLE BODY APPROXIMATIONS OF RIGID BODY DYNAMICS

Dr. John McDonald

School of CTI, DePaul University,
243 S. Wabash Ave
Chicago, IL 60706 USA
jmcdonald@cs.depaul.edu http://mcdonald.cs.depaul.edu

ABSTRACT

This paper demonstrates that techniques in flexible body dynamics can yield surprising results when applied to rigid bodies. The discussion presents a technique for constructing rigid bodies from collections of masses and springs, and demonstrates that the simulation calculates many features of rigid body dynamics such as the behavior of the center of mass and the moments of inertia, free of charge. Moreover, complex rotational features such as the precession of a spinning top and the behavior of a gyroscope arise, again without the need of extending the model in any way.

Keywords: Physically based modeling, animation, rigid body dynamics, particle systems, mass-spring systems

1. INTRODUCTION

Rigid body dynamics has long been a key element of physically based modeling for computer animation, and has been used to solve a wide range of problems. Since many animations concern the interactions of essentially rigid bodies, the simulation of rigid body dynamics will continue to be essential to animators.

This paper demonstrates that an alternative to traditional rigid body dynamics can yield unexpected benefits for physical simulation. In particular, these techniques can eliminate the need for a range of complicated physical calculations required in the setup of a rigid body simulation, and can yield many subtle properties of rigid body motion with no extra work on the part of the animator.

The models presented here are flexibly bodies consisting of a set of particles connected by springs that are stiff enough that internal vibrations in the objects are below the threshold of onscreen visibility. Stiff springs can cause serious instability problems in the numerical integrator used to drive the simulation. However, for simple to moderately complex models, this stiffness does not pose a serious problem as today's PC's are fast enough to run the simulations at real or even faster than real-time speeds, with time steps small enough that the simulation is stable.

Flexible body techniques give several key benefits to a simulation or animation.

- Object modifications are easy without excess recalculations or code changes
- Added physical properties of objects are included in the simulation without extra design work
- Techniques can serve as a conceptual bridge between the subjects of interacting particles and rigid body dynamics

2. RIGID BODIES IN ANIMATION

Rigid body dynamics is one of the foundations of classical mechanics for good reason. Its techniques can be applied over a wide range of problems and are very effective at solving large-scale dynamics problems while ignoring the small vibrations and distortions inside the objects themselves.

The techniques for calculating rigid body motion generally follow the pattern:

- Calculate the inertia tensors for each object
- Find the current position and orientation for each object
- Calculate the forces and torques exerted on each object
- Calculate the induced accelerations, both linear and rotational
- Use some technique (e.g. fourth order Runge-Kutta) to numerically integrate the resulting differential equations.

It is important to note that both the rigid theory and the flexible body theory require the use of a numerical integrator. As we will see, the prerequisite knowledge for the flexible body theory is generally no more than that of the rigid theory and is, in many cases, surprisingly less.

Traditional methods require a deep understanding of the physics of rigid bodies, including topics such as

- Inertia tensors
- Angular momentum and angular velocity
- Induced forces, such as the torque caused by gravitational pull on a rotating object

A significant amount of time is necessary to develop these topics in a classroom, and the basic set-up of a solution must be altered if the properties of the objects change in significant ways. For example, altering the body's shape or mass distribution requires the recalculation of the inertia tensors. Hence the rigid body techniques often lack flexibility conducive to experimentation.

This is not to say that students of physically based modeling should ignore a deep understanding of physical laws. Rather, these techniques may be used to reinforce intuition while trying to teach students the more subtle consequences of rotational dynamics. Further, the flexible body techniques can be a powerful tool for experimentation.

Animators require a large body of simulation techniques, and the techniques presented here form a valuable addition to existing methods.

3. REVIEW OF RIGID BODY DYNAMICS

Let *B* be a rigid body moving freely in three-dimensional space, and let *c* be its center of mass. The state of *B* can be completely described, at any point in time, by the position of its center of mass *c* and by the orientation of *B*. Orientation is defined as a rotation of *B* from some reference coordinate frame, and can be described by a set of Euler angles, by a quaternion, etc. No deformations in the body itself are considered.

Choosing Euler angles for our representation, we can see that the state of a rigid body *B,* at some point in time, is a six dimensional Euclidean vector, consisting of three Euclidean coordinates (x, y, z) and three Euler angles (r_x, r_y, r_z).

The velocity and acceleration of the center of mass are treated just as in the case of a particle. In addition, we have the angular acceleration and velocities. Denote angular velocity by ω, and angular acceleration simply by $d\omega/dt$. See Figure 1.

A force *F* applied to *B* at a point on the surface of the object has two effects.

1. It causes the body's center of mass to accelerate as if the object were a point mass centered at *c*.
2. It applies a torque to the object, causing an angular acceleration of the body about *c*, given by the formula

$$\tau = r \times F \qquad (1)$$

From this, we can calculate the effect on the angular momentum using the rotational analogue to Newton's second law of dynamics:

$$\tau = \frac{d}{dt}(I \cdot \omega) = I \frac{d\omega}{dt}, \qquad (2)$$

where ω is the angular velocity of the object, and *I* is the inertia tensor of the object. Recall that the terms of *I* are usually calculated with triple integrals over the volume of *B*. For a more detailed discussion see [Mario70] or [Feynm63].

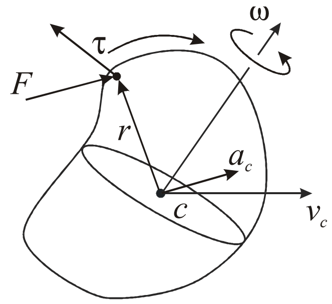


Figure 1: Force and a rigid body

The inertia tensor can be simplified by a careful choice of principal axes of rotation, analogous to the diagonalization of a linear transformation[Mario70]. Inertia tensors remain, however, a challenging calculation for irregularly shaped objects.

Rigid body techniques use equation (2) and Newton's second law, together with a numerical integrator, to update the position, orientation, velocity and angular velocity at each time step.

Introduced into this mix are also constraint and collision calculations that make the object react to other objects in the scene. For example, a billiard ball *B* might collide with another billiard ball, hit the edge of the table. The table also exerts a repulsive force to keep the ball from falling through to the floor.

The techniques for handling collisions and constraints are varied, ranging from collision forces, which repel objects as they come together, to impulse based calculations, which calculate the accelerations, both linear and rotational, caused by rigid collisions [Baraf93][Mirti95].

4. INTERACTING PARTICLES AND MASS-SPRING SYSTEMS

Another face of physically based modeling is the simulation of flexible bodies and surfaces. This field has seen its most recent successes in the simulation of cloth, e.g. clothing, flags, etc.[Provo95] [Baraf98]

Many of the techniques for flexible body dynamics extend the notion of a particle system; a set of non-interacting bodies affected by external forces. To model deformable surfaces or objects, we add interaction forces between the particles. These forces come in many forms, from the simple linear spring forces discussed here, to the more complicated inter-atomic forces such as the Lennard-Jones potential law. [Szeli92] [McDon99]

Except for collision considerations, this discussion will primarily concern linear force laws. They are simple and yield the features that we require from the simulation. The discussion will assume that the reader is basically familiar with numerical integration. For a more complete discussion, see [Press92] or [Burde96].

4.1 Linear Springs

The basic unit of many flexible body simulations is the damped linear harmonic oscillator. Recall that an equilibrium length, a spring coefficient (called the Hooke's law constant), and a damping coefficient define such a spring.

Let p_1 and p_2 be particles in space and let s be a damped spring between p_1 and p_2. Suppose that s has equilibrium length L_0, spring constant k, and damping coefficient c. Let $\mathbf{u} = p_2 - p_1$ be the vector from p_1 to p_2. Then the force the spring exerts on particle p_i is given by

$$m\mathbf{a}_i = \mathbf{F}_i = (-1)^{i-1}\left(k(L-L_0) - c\frac{d|L|}{dt}\right)\mathbf{u} \quad (3)$$

The $(-1)^{i-1}$ term simply reflects the fact that the force is equal and opposite in its effect on the two particles.[Feynm63]

If L is in meters, m is in kg, and t is in seconds then the units for the spring constant k are kg/s^2, and the units for c are kg/s. If desired, we can look up realistic values for various materials, but often experimentation is used to find suitable values.

Mass-spring systems are commonly used to simulate such systems as swinging ropes and bungee cords. In the case of a swinging rope, we link a series of particles of small mass (.1kg) together with springs, and give the particle at the end of the rope a large mass (70kg) to simulate a weight hung on the end.

For the rope simulation to be convincing, the rope should not "stretch" much under the weight of the object. Thus the rope's springs must have a high spring constant k, on the order of 10^5.

It is essential that the springs have a measure of damping in them; otherwise the rope will vibrate more like a swinging metallic wire than a rope. A damping constant of order 10^1 or 10^2 kg/s will give realistic motion in many situations

4.1 Numerical integration of mass-spring systems

The problems associated to stiff springs and numerical integration are well known[Baraf98], and much work has been done to try to avoid using stiff springs in models. This is one of the reasons that more traditional rigid body techniques have been so successful.

Recently Baraff and Witkin have applied more sophisticated integration methods such as Inverse Euler to counteract this stiffness, though at the cost of increased computation for each time step. [Baraf99][Kang00]

Certainly, one should eventually apply a more sophisticated integrator, but we shall demonstrate that even ordinary fourth-order Runge-Kutta does an acceptable job at handling these models, and does so with very little code! Again, this can significantly benefit introductory physically based modeling courses, because the algorithm is relatively simple and the model is straightforward.

To summarize the problem with stiff springs and integration, consider that a spring s with constant k has a fundamental frequency f at which it vibrates. For a single spring oscillating with a total mass of m attached, this period can be calculated as

$$\rho = 2\pi\sqrt{\frac{m}{k}} \quad (4)$$

Certainly, the time step must be smaller than this period and depending on the chosen integration technique, it usually must be much smaller. For fourth order Runge-Kutta, taking h between $\rho/5$ and $\rho/10$ is often safe. For a time step of h=.0005, this gives us the ability to take k/m as high as about 3×10^6, which is stiff enough for many applications.

It is important to remember this instability when building a model. The more the model is subdivided, the smaller the mass of each particle in the model, and thus the higher the fundamental frequency of the springs involved. A simulation must strike a careful balance between complexity and stability.

Fortunately, ρ varies as $\sqrt{m/k}$ for the harmonic oscillator. Thus reducing the time step by a factor of 2, often allows an increase in the spring constant to mass ratio by a factor of four.

The above time step guidelines are only estimates and depend on a great many factors including the other forces present in the system and on the distribution of springs in the model. For more sophisticated error estimates see [Press92] or [Burde96].

These experimental guidelines do, however work for simple situations. If a time step fails, the usual solution applies. Lower the time step until the simulation is stable, or use an adaptive time step integration method [Press92]. If the stable time step is so small as to make the simulation unfeasible, then other techniques must be sought.

5. SIMULATING RIGID BODIES WITH FLEXIBLE MODELS

Most objects are not truly rigid at all, they are made up of constantly vibrating atoms and can exhibit quite large deformations in-spite of being essentially rigid. We will see that internal forces and small vibrations in the system can have a highly non-trivial effect on the macroscopic behavior of the system. Therefore it can be beneficial to include such interactions in our model, even crudely.

So, consider these objects as very stiff flexible bodies, instead of rigid ones. Their atoms are in flux, but they are held together so tightly that we usually cannot see the vibrations and deformation that accompany movement and collisions.

To model a rigid body, take a collection of widely spaced particles and connect them by a set of springs that are stiff enough so that any vibrations are below the threshold of visibility on-screen. Thus, the model forms a rather crude approximation to the internal forces in a rigid body. The surprise is that this crude approximation is good enough to model many properties of linear and rotational dynamics!

For simplicity, begin by considering an irregular pendulum hanging from one fixed point, but otherwise free to swing about this focus. We will consider the fixed point to be completely rigid for translation and frictionless for rotation. For this discussion an irregular pendulum is one that is *not* described as an idealized point mass suspended by a rigid mass-less rod, but rather a more realistic rigid body swinging about a fixed focus.

5.1 Irregular Pendula

One of the simplest non-trivial rigid bodies is the cube. It is one of the first examples studied in a course on rigid body dynamics, because its inertia tensor and principle axes are easy to calculate.

Let B be a cube situated in some orientation in space. Treat this cube as a pendulum, by choosing one of the vertices v of the cube as the focus of the pendulum. Then suspend the cube from this focus. Allow, the cube to swing around v freely with a full three degrees of freedom (three Euler angles), but otherwise treat v as a perfectly rigid frictionless joint.

5.1.1 The Rigid Body Theory

Considering this system as a constrained rigid body problem has several subtleties. For a swinging pendulum it is, at first, tempting to place the origin of rotation at the focus of the pendulum, instead of at the center of mass, to eliminate all but rotational motion from the equations.

While we can calculate the moments of inertia of B in such a system, calculating the torque induced by the uniform gravity field is non-trivial, because our center of rotation is not at B's center of mass. It is therefore more convenient to place the cube's origin at the center of mass, and to choose principal axes through each face of the cube. [Mario71]

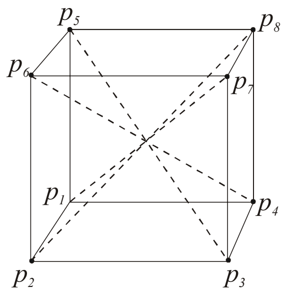
These choices diagonalize the inertia tensor, and in fact, yield identical moments of inertia for each of the axes, namely $1/6Mb^2$, where M is the mass of the cube and b is the length of one of its sides. With this setup, though, we need to apply a constraint force at the focus of the pendulum.

Several different methods have been applied to constrain such systems while keeping the torques and rotations centered at c, where they are easiest to calculate [Barze88][Witki97]. Many techniques use spring-like penalty forces, to enforce the constraint, which can have the same stability problems as the present mass-spring model.

With the flexible body technique, we eliminate the physical calculations necessary to set up such problems, and we can easily simulate swinging pendula of a wide array of shapes, given a nice triangulation. (Too much subdivision in any one region of the body will yield very small masses and destabilize the system.)

5.1.2 The mass-spring model

The flexible model is quite elementary, consisting of 8 masses and 28 springs, and requires only the spring force law and the chosen integrator for simulation.



Figrue2: The Mass-Spring System for a Cube

Place eight particles p_i, $i = 1 \ldots 8$, at the eight corners of the unit cube. Place 28 identical springs between the following pairs of particles:

(p_0, p_1), (p_1, p_2), (p_2, p_3), (p_3, p_0) – Top Square
(p_4, p_5), (p_5, p_6), (p_6, p_7), (p_7, p_4) – Bottom Sqr
(p_0, p_4), (p_1, p_5), (p_2, p_6), (p_3, p_7) – Vertical
(p_0, p_6), (p_1, p_7), (p_2, p_4), (p_3, p_5) – Inner Support
(p_0, p_5), (p_4, p_1), (p_1, p_6), (p_5, p_2) – Face Support
(p_2, p_7), (p_6, p_3), (p_3, p_4), (p_7, p_0) – Face Support
(p_0, p_2), (p_1, p_3), (p_4, p_6), (p_5, p_7) – Face Support

Set the mass of each particle to 1/8kg, so that the total mass of the cube is 1kg. For each spring set the spring constant to 10^5 kg/s^2, and the damping constant to 10 kg/s.

Figure 2 shows the cube and the first 16 springs in the model. The last 12 springs span each face, giving the cube the structural integrity needed to look rigid. To finish the simulation, fix the point p_0 and apply a constant gravitational acceleration.

Note that we do not need to calculate constraint forces for this model at the focus of the pendulum.

Since the focus is itself a particle, we can simply zero its velocity and acceleration at each time step. The springs then calculate for us the constraint force exerted on the rest of the body.

This model extends quite easily to more complicated shapes with nothing more than a calculation of a triangulation of the body. This is no great burden since, to render the object itself, we would need to triangulate its surface, and then, triangulating the interior is trivial. For a convex object we can put a particle at the center of mass, and create springs radiating from the center to each vertex. For non-convex objects, we can divide the object up into a collection of convex objects and proceed as before. Once we have built the basic mass-spring model, we can add extra springs for structural integrity as needed

5.2 Rotations, Gyroscopes and Tops

Now, consider a rotating rigid body. Here, we see an astonishing benefit to the mass-spring construction of a rigid body. Sophisticated features of rotational motion arise quite free of charge. Again, the springs inside the model are taking care of a great deal of physical bookkeeping for us. Our model doesn't have to be extended at all to include these features.

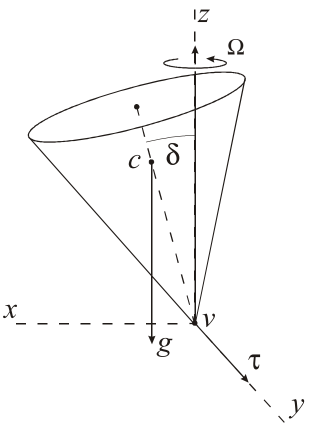


Figure 3: Rigid Body Dynamics of a Symmetric Top

This problem has, of course, been completely solved by other means.[Mario70] No attempt is made here to improve on the classical theory. It is, however, highly instructive to see how the features of the rotational motion of the top arise from simple simulations of the internal forces in the spinning object.

5.2.1 The Rigid Theory of Rotating Tops

The following is a brief review of the forces and torques involved in a rotating gyroscope or top. A complete discussion of this theory may be found in [Mario70][Feynm63].

Let B be a rigid body, and let c be its center of mass. Suppose that B is fixed at one point v on its surface, and rotating about an axis $\boldsymbol{a}$ through v. Suppose that the axis $\boldsymbol{a}$ lies at some positive angle δ from vertical. See figure 3.

The force of gravity pulls down on the center of mass c of B, and induces a torque $\boldsymbol{\tau}$ that rotates B away from the vertical axis. This angular acceleration coupled with the angular momentum of B about $\boldsymbol{a}$ causes the top to precess about the vertical axis according to the formula

$$\tau = \Omega \times a \,. \tag{5}$$

where Ω is the angular velocity (precession rate) about the vertical axis.

5.2.2 A "Flexible" Top

Let T be a circular conical top or gyroscope of mass M, inverted so that it is standing on its vertex. The fixed point of rotation will be the vertex of the cone. A conical shape is easy to build and it is nicely symmetrical. Note, however, that such a top could take many different shapes with these techniques.

As indicated in figure 4, we will place one particle v at the vertex of the cone, and one particle p at the top of our cone on the axis of symmetry.

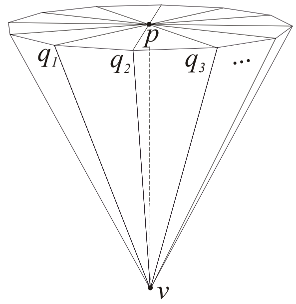


Figure 4: The Mass Spring Model for the Top

Place an array of n (say n = 12) particles $q_1 \ldots q_n$ in a circle about the edge of the top face of the cone, and link them with springs to their immediate neighbors, to p and to v. Lastly, place a spring along the symmetry axis of the cone connecting p and v.

Give each of the springs a Hooke's law constant of 10^6, and give each of the q_i a mass of $M/2n$. Thus, the ring on the top edge to the gyroscope contains half of the mass of the object. Give p and v each mass $M/4$ to account for the rest of T's mass. Remember to adjust the integration step size appropriately if n is large, giving each q_i a small mass.

To complete the setup of the simulation,

1. Tilt the top at an angle δ from vertical. Remember that a vertical spinning top just spins on its axis.
2. Give the particles an initial velocity to make the top spin about its axis.

Let the top be initially oriented vertically, and choose an angular velocity ω. Choose a cylindrical coordinate system so that the coordinates of q_1 are

(0, r, h), where r and h are the radius and height of the top, respectively. Then the coordinates of q_i are

$$q_i = \left(\frac{i\pi}{6}, r, h\right). \tag{6}$$

To spin the top, apply a tangential velocity to each of the q_i. We do not need to affect the vertex v, as it is an elementary particle and contains no orientation information itself. So, give each q_i a velocity of

$$v_i = (-2\pi\omega\sin\theta,\ 2\pi\omega\cos\theta,\ 0) \tag{7}$$

to give an angular velocity of ω. Then apply a rotation of δ about the y-axis to the entire system (including the v_i), to tilt the system into place.

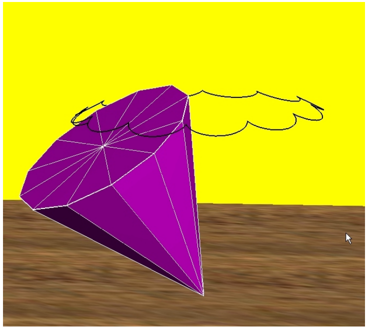


Figure 5: The Precession of a Top

A constant gravitational acceleration applied to each point and numerical integration finishes the simulation. Figure 5 shows a snapshot of the top spinning along with a trace of the particle v. The figure clearly shows the precession and nutation of the top, agreeing with the form of the path that one would expect from rigid dynamics. [Mario70]

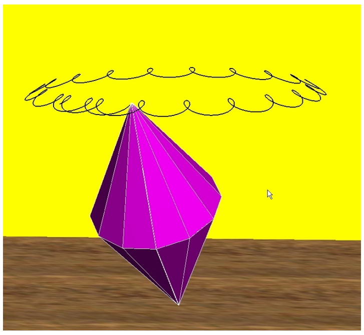


Figure 6: Top Started with Initial Precession

In the preceding example we started the simulation with the top's center of mass stationary. The form of the trace varies depending on whether we start the top with an initial precession about the vertical axis. Also, as stated previously, this technique is flexible enough to allow rapid changes to the top's shape. Figure 6 shows an elongated top, started with an initial precession opposite the precession it will get from its own spinning. This trace also agrees with the form predicted by the rigid theory.

Not only can we change the shape of the top, but also we can easily change the mass distribution to create an unbalanced top. Figure 7 shows the results of doubling the mass of one of the particles on the outer ring of the top.

6. CONSTRAINT FORCES AND COLLISION

The examples above have limited our object to rotational motion, as the object was always held fixed at a point on its surface. This was to demonstrate the special nature of the rotational motion of the body.

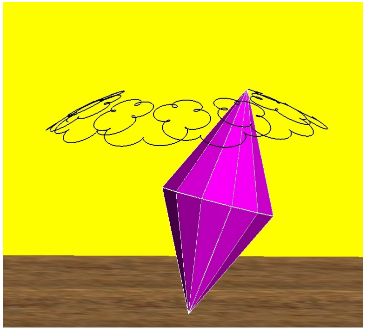


Figure 7: An Unbalanced Top

Now, turn to the more general case where the object is free to move in space subject to one or more constraints. This discussion will consider only simple constraints, limited to collisions of the object with walls, floors and other flat surfaces. The same technique would also apply if the surface were defined analytically and were reasonably smooth.

6.1 Floors and Walls

Now, consider releasing the fixed point of our object and let the object interact with the floors and walls of a virtual room. We just need to model the collisions of the object with these surfaces. Fortunately, since the surfaces are flat, we only need to consider collisions of the individual particles in the object with the surface.

If the object were to interact with a dramatically more complex surface, one with many small hills and valleys, then we would need to use a more sophisticated collision detection and response algorithm.

Realistic friction calculations require knowledge of a normal collision force that the surface exerts on the object. Therefore, it is convenient to use a collision force to model the interaction.

To model the collision, apply an inverse square force that begins some small distance Δ above the surface. We normalize the force so that it is 0 at Δ. Thus, the force is described by the following equation

$$F_n = \max\left(\frac{c}{h^2} - \frac{c}{\Delta^2},\ 0\right) \tag{8}$$

The problem with such a repulsion force, is that it can cause stability problems in the numerical integrator.[Baraf98] After all, if the time step is large enough that the particle can go from above Δ to below the surface in a single time step, we certainly have a problem. This trouble is partially mitigated by the fact that we already have a fairly low time step, due to spring stiffness.

Unfortunately, this does not entirely prevent such an occurrence. Moreover, we have to consider the fact that the force itself causes a stability problem since it is unbounded as we approach the surface. Therefore, we will prevent the particle from passing closer than a distance of Δ/10 to the surface. See figure 8. If a particle passes this barrier, we will simply move it back to a distance of Δ/10, and simultaneously set the component of the velocity normal to the surface to 0.

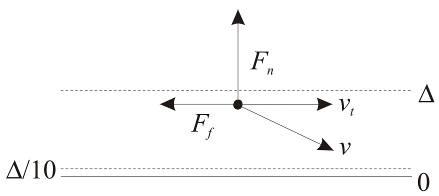


Figure 8: The Repulsion and Friction Forces

Techniques somewhat similar to this have been used to calculate inter-particle collisions in cloth. [McDon99] The relative benefits of this technique become clear when considering the inclusion of friction forces in the model.

6.2 Friction, Inelastic Collisions

Collisions of the object with the above surface resemble an object falling on a sheet of ice, as the system contains no friction or dissipative forces other than those in the springs themselves. The result is that, if we drop an irregularly shaped object on the floor, its center of mass will bounce straight up and down. There are no lateral forces on the object to cause a deviation of the center of mass.

Also, if we unbind the vertex of the top, the vertex will initially slide out from underneath the top and the top will settle down to spinning in place about its center of mass. This is a powerful demonstration of the fact that this simulation exhibits the physical laws concerning the motion of the center of mass.

There are two types of dissipation to consider for collisions with surfaces: sliding friction and collision elasticity.

6.2.1 The Friction Force

Consider the usual definition of friction in an introductory physics course. This simplification gives good results and is easy to calculate. Let v be the velocity of the particle as it strikes or slides on the surface. Since the repulsive force does not restrict the particle to sliding tangentially, there may be some normal component to v. Let v_t be the component of v tangential to S.

Friction is a force tangent to the surface, in the direction of v_t, and proportional to the normal repulsive force exerted by the surface on the object.

$$F_f = \mu |F_n| \frac{v_t}{|v_t|}, \quad v_t \neq 0 \tag{9}$$

Luckily, we've already calculated F_n in the above collision response algorithm. Thus, we can model friction precisely as stated, resulting in a tangential friction force as in figure 8.

For stability, it is important that the total effect of this force in a single time step be less than the current tangential component of the velocity. Therefore, we put a limit on the magnitude of the friction force generated by this algorithm. If the force becomes larger than this magnitude, we will simply scale the force to be equal to that maximum. Taking $|v_t|/2h$ works nicely.

6.2.2 Collision Elasticity

Note that we are already partially modeling such dissipation with the damping terms in the springs' forces. We are not, however, taking into account the dissipation occurring in the material of the floor. This is more of a problem, because we are only modeling the surface as an inverse square force.

As an approximation, introduce a term into the surface's force law, which is analogous to the dissipative forces inside the springs themselves. Consider the normal component v_N of the particle's velocity as it strikes the surface. As long as the particle is in contact with the surface (i.e. as long as there is a normal repulsive force being applied by the surface) we will apply a force counter to v_N and proportional to the magnitude of v_N.

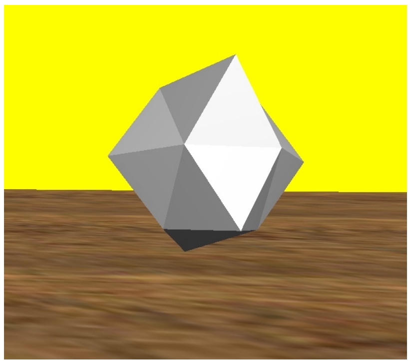


Figure 9: Irregular Body Falling to the Floor

This approximation produces quite realistic inelastic collisions without adding much complexity. To test both the friction and collision forces, consider two examples. The top, in this situation, simply spins in place about its center of mass, and slowly sinks to the ground due to the energy lost to friction. An irregularly shaped object falling to the floor provides a better example of the dynamics of the collisions. See Figure 9.

7. PERFORMANCE

These simulations were performance-tested on an Intel Pentium III-800 and on an AMD Athalon-800, on models with varying complexity. A benchmark of the spinning top, with 37 springs and 14 masses, gives a good indication of the performance of this algorithm on these machines.

On the Pentium III, the algorithm ran about 14,000 iterations/sec, while the Athalon achieved nearly 18,000 iterations/sec. The Pentium and Athalon are, therefore, fast enough to support a model nearly seven and ten times the complexity, respectively, in real time with a step size of .0005.

8. CONCLUSION AND FUTURE WORK

It is surprising that such large scale, and rather crude, approximations to the inertial distributions and flexibility forces at work in rigid bodies could give rise to the subtle features of rotational dynamics observed here.

An attempt should be made to significantly improve the numerical integration techniques used. As stability is the main barrier, inverse methods such as the Inverse Euler or Rosenbrock methods may be appropriate. However, since these involve large and expensive linear system inversions, their usefulness in real-time animations is uncertain.

This technique presents an approximation of the ideal rigid body case, and the spring forces were chosen accordingly. An effort should be made to relate the spring forces to the more realistic internal forces present in a real top, made from wood, metal, plastic, etc. It would also be helpful to perform a comparison of such models to more traditional simulations of rigid rotational dynamics.

Another main addition to these techniques will be the introduction of object-object collisions, rather than simply object-surface. This will increase the complexity of the simulation significantly, but will also increase their usefulness to production animation techniques.

REFERENCES

[Baraf98] Baraff, D, Witkin, A: Large Steps in Cloth Simulation, *Computer Graphics* (Proc. SIGGRAPH), 1998, pp. 43-54.

[Baraf93] Issues in Computing Contact Forces for Non-penetrating Rigid Body Simulation. Computer Graphics (Proc. SIGGRAPH), 1993, pp. 292-352.

[Baraf92] Baraff, D, Witkin, A: Dynamic Simulation of Non-Penetrating Flexible Bodies, *Computer Graphics,* (Proc. SIGGRAPH), 1992, pp. 303-308.

[Baraf91] Baraff, D: Coping with Friction for Non-Penetrating Rigid Body Simulation. Computer Graphics (Proc. SIGGRAPH), 1991, pp. 31-40.

[Barzel88] Barzel, R: A Modeling System Based on Dynamic Constraints, *Computer Graphics,* Vol 22, No 4, August 1988.

[Bolin95] Bolino, P, Courchesne, M, Magnenat-Thalmann, N: Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects, *Computer Graphics* (Proc. SIGGRAPH), 1995, pp. 137-144.

[Burde96] Burden, R, Faires, D: Numerical Analysis, Sixth Edition, *Brooks & Cole Publishing, 1996*

[Feyn63] Feynman, R, Leighton, R, Sands, M: Lectures on Physics, Vol. 1, *Addison Wesley*, 1963.

[Humme97] Hummel, A, Girod, B: Fast Dynamic Simulation of Flexible and Ridig Bodies with Kinematic Constraints, *ACM VRST,* 1997, pp. 125-131.

[Kang00] Kang, Y, Choi, J, Cho, H, Lee, D, Park, C: Real-Time Animation Technique for Flexible and Thin Objects, *Conference Proceedings*, WSCG 2000, pp. 322-329

[Mario70] Marion, J: Classical Dynamics of Particles and Systems, Second Ed, *Academic Press,* 1970.

[McDon99] McDonald J: Methods for Preventing Cloth Self-Intersection, *Conference Sketches and Applications,* SIGGRAPH, 1999.

[Mirti95] Mirtich, B, Canny, J: Impulse-based simulation of rigid bodies. *Symp. on Interactive 3D Graphics*, 1995, pp. 181-217

[Press92] Press, W, Teukolsky, S, Vetterling, W, Flannery, B: Numerical Recipes in C, The Art of Scientific Computing, *Cambridge University Press,* 1992.

[Provo95] Provot, X: Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior, *Proc. Graphics Interface*, 1995, pp 147-154.

[Sauer98] Sauer, J, Schömer, Elmar: A Constraint-Based Approach to Rigid Body Dynamics for Virtual Reality Applications, *ACM VRST*, 1998, pp. 153-161.

[Szeli92] Szeliski, R, Tonnesen, D: Surface Modeling with Oriented Particle Systems, *Computer Graphics,* (Proceedings SIGGRAPH '92), 1992, pp. 185-194.

[Witki97] Witkin, A, Physically Based Modeling: Principles and Practice. SIGGRAPH '97, Course Notes, 1997.