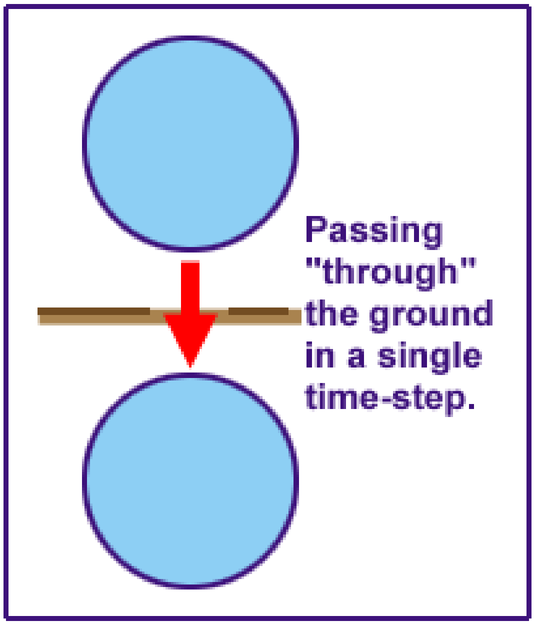
High-Speed Bouncing (and Collisions)

Prepared by: Teton Multimedia (www.TetonMultimedia.com)

Batting an air-filled balloon around the kitchen is easy. The balloon moves slowly, and timing your swing couldn't be much less demanding. But hitting a 90 mile-per-hour fastball zipping over home plate is relatively difficult. The same is true when building computer models of fast-moving objects. Timing must be relatively exact, and there is little room for error.



As noted in "Bouncing Ball Part II (The Ground)," if a ball is moving very fast, it can pass completely through an object in a single time-step. This means that a collision will go undetected, and the animation will be completely "wrong." By limiting the time-step size, we can prevent this type of problem.

In Director, the time-step size can be set using the "PuppetTempo" command. An alternate and interestingly visual method is to define several frame labels with corresponding FrameTempos. Then simply select the frame label that suits the animation velocity. This method is demonstrated in the accompanying Director movie.

The maximum permissible time-step size is derived from the maximum distance the ball is permitted to travel in a single time-step. Generally, some fraction of the radius is used, such as 0.5 or 0.2. This gives us the following equation.

$$r * frac > v * \Delta t \quad (1.1)$$

Rewriting this equation gives us

$$\Delta t < r * frac / v \quad (1.2)$$

Because the FrameTempo is the reciprocal of the time-step size, we then arrive at

$$\text{FrameTempo} > v / (r * frac) \quad (1.3)$$

Note that the inequality has changed directions, and the FrameTempo must be *greater than* the calculated minimum. (A high tempo corresponds to a small time-step size.)

Maximum Velocity

Next, it is common to limit an object's velocity to some "reasonable" maximum. First, consider that a typical refresh rate of a monitor is around 75 Hz, which means that it refreshes 75 times a second. If the FrameTempo is much higher than this, then many calculations will not be "seen" by the monitor, and consequently they won't be seen by the observer. Animations will appear choppy, collisions will occur without being "seen," and results may not appear satisfactory to the human observer. However, this is something to "twiddle" with in the context of *your* animation. Often you can run at 200 fps or more and everything will seem just fine.

Rewriting equation 1.3, we arrive at the equation for the maximum velocity (V_{max})

$$v_{max} = FT_{max} * r * frac \quad (1.4)$$

Where FT_{max} is the maximum frame tempo we feel like permitting. Then, within our animation, all calculated velocities are restricted by this maximum.

Preventing "Double Hits"

Next is an interesting situation that can affect collision detection at any speed (not just at high speed). In our simple bouncing example, when a collision is detected, the sign of the velocity is reversed (multiplied by –1). This produces the "bounce." However, this isn't always sufficient. Consider a time-step when a collision "almost" occurs, followed by a time-step in which a collision *does* occur (the ball overlaps the ground), followed by yet another time-step in which the ball doesn't quite escape from the ground. This can happen if the collision is not completely elastic (the ball slows down as a consequence of the bounce) or if V_{max} testing is performed only when collisions occur. In either case, the ball exits the collision at a lower speed than it enters the collision.

The possible result: two successive time-steps in which there is overlap. The ball's velocity will change signs twice in succession, pointing it back *into* the object with which it collided. Then it will probably detect another collision, and another and another, and the ball will appear "stuck" as it rattles back and forth from one time-step to the next. Consequently, your collision detection must prevent collisions in adjacent time-steps (unless colliding with multiple objects).

The accompanying Director movie illustrates these principles. It also suggests features to deactivate to observe the result.