8 Physical Simulation

8.0 Introduction

Over recent centuries the scientific approach has revealed various systems of laws to describe the dynamics of objects, electromagnetic phenomena, atomic forces and cosmological models for the universe. Each domain has its own set of laws, and for this chapter we will restrict the examples to some of the simple behaviours of objects related to our own human scale. In particular, we will investigate the motion of objects in a gravitational field, collisions with other objects, and other simple dynamic systems.

It was mentioned at the start of Chapter 7 that traditional animators rely upon a mixture of visual memory and drawing skills to simulate various physical behaviours. When a line test reveals that the animation is not correct, it is redrawn and retested until it satisfies the animator. Unfortunately, such techniques cannot be used in real-time computer graphics. If we want realism, we must rely upon deterministic procedures that encode knowledge of the physical world. Such procedures may be based upon either empirical laws or techniques that have been 'tweaked' to mimic a particular physical behaviour. Both approaches have their advantages and disadvantages: on the one hand, numerical techniques can be made very realistic and efficient, but may have limited application; whereas empirical laws accurately describe physical behaviour and enjoy a wide domain of applications, but may require significant levels of computation. In the circumstances, we must be prepared to use the best techniques for the right application and use our experience to decide how and when to make this choice.

When considering the motion of objects we cannot ignore Newton's contribution to this domain. Newton's laws of motion provide us with the primary behaviours of theoretical particles having mass but no size. These laws, however, can be employed to describe the motion of very large objects without introducing significant errors.

Newton's first law of motion states that 'the momentum of a particle is constant when there are no external forces'. This law reminds us that an object moves in a straight line unless disturbed by some force. In reality, the universe is awash with force fields, be they gravitational, electric, magnetic or atomic, and in general it is only when we move away from the earth that objects begin to obey this law.

Newton's second law of motion states that 'a particle of mass m, subjected to a force F, moves with acceleration F/m'. This law is also called the equation of motion of a particle, and reminds us that work has to be done whenever an object is accelerated.

Newton's third law of motion states that 'If a particle exerts a force on a second particle, the second particle exerts an equal reactive force in the opposite direction'. This law is useful when considering collisions between objects. Let us now examine how these laws can help us simulate some simple physical behaviours.

8.1 Objects falling in a gravitational field

When an object is moved away from the earth's surface it is attracted back through a gravitational force. This force is called the object's *weight* and is equal to mg, where m is the object's mass, and g is the acceleration due to gravity. Generally, g equals $9.81\,\mathrm{ms}^{-2}$, but is 0.3% smaller at the equator and 0.2% larger at the poles. As the vertical y-axis represents positive offsets from the origin, upward velocities are defined positive and downward velocities negative.

In general, if a particle has an initial vertical velocity of v and acceleration a, the distance d travelled after time t is given by:

$$d = vt + \tfrac{1}{2}at^2 \tag{8.1}$$

If the particle has an initial vertical velocity of v_0 and is acceleration under the influence of gravity, the distance d becomes:

$$d = v_0 t - \tfrac{1}{2}gt^2 \tag{8.2}$$

In Figure 8.1 a particle is positioned at point P_0, at a distance h_0 above the ground plane. If the particle is allowed to fall under the action of gravity, after time t it will be at P_1 having fallen a distance d:

$$d = -\tfrac{1}{2}gt^2 \tag{8.3}$$

Its height h_1 above the ground plane is given by:

$$h_1 = h_0 + d = h_0 - \tfrac{1}{2}gt^2 \tag{8.4}$$

Calculating h_1 after any time t is trivial. But say we want to identify the point in time when the particle collides with the floor – this requires further analysis.

If the status of the particle is sampled at various time intervals, the point

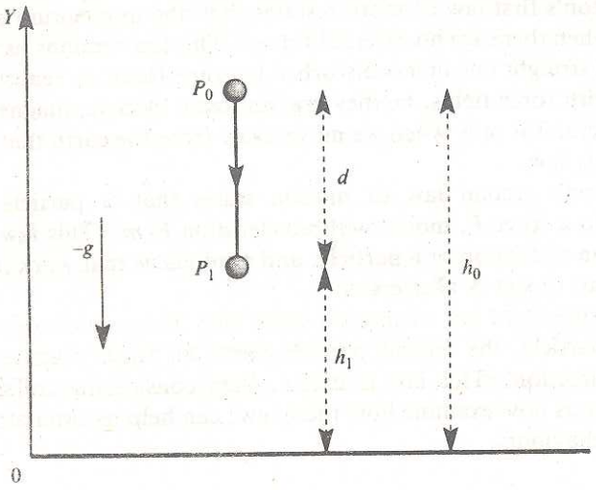

Figure 8.1 If a particle P is dropped from a height h_0, after time t it will have travelled $\frac{1}{2}gt^2$ downwards. Therefore, its new height h_1 is $h_0 - \frac{1}{2}gt^2$

of collision with the floor is detected when the particle's height is zero. Because of the sampling process we will not automatically discover this condition. It is much more likely that the particle's height becomes negative and the point of intersection has to be computed. This can be implemented by maintaining a record of the particle's velocity and height at the previous sample.

Let the following parameters be used for the simulation:

h is the particle's height above the ground plane
v is the particle's velocity
t is the time at the sample point

When the status of the particle is sampled, it will produce values h_{now} and v_{now} at t_{now}, while the previous values will be h_{prev} and v_{prev} at t_{prev}. The values of h_{now}, v_{now}, and t_{now} become t_{prev}, h_{prev} and v_{prev} when the next sample is made. Figure 8.2(a) shows this situation. If $t = (t_{now} - t_{prev})$ the velocity v_{now} is given by:

$$v_{now} = v_{prev} - gt \tag{8.5}$$

while the new height h_{now} is given by:

$$h_{now} = h_{prev} + v_{prev}t - \tfrac{1}{2}gt^2 \tag{8.6}$$

If $h_{now} > 0$, the particle is still falling. If $h_{now} = 0$, the particle has just hit the floor, and with a perfect elastic collision will bounce upwards with its velocity reversed (see Figure 8.2b). If $h_{now} < 0$, the particle has passed through the floor as shown in Figure 8.2(c). When this occurs we can take various actions:

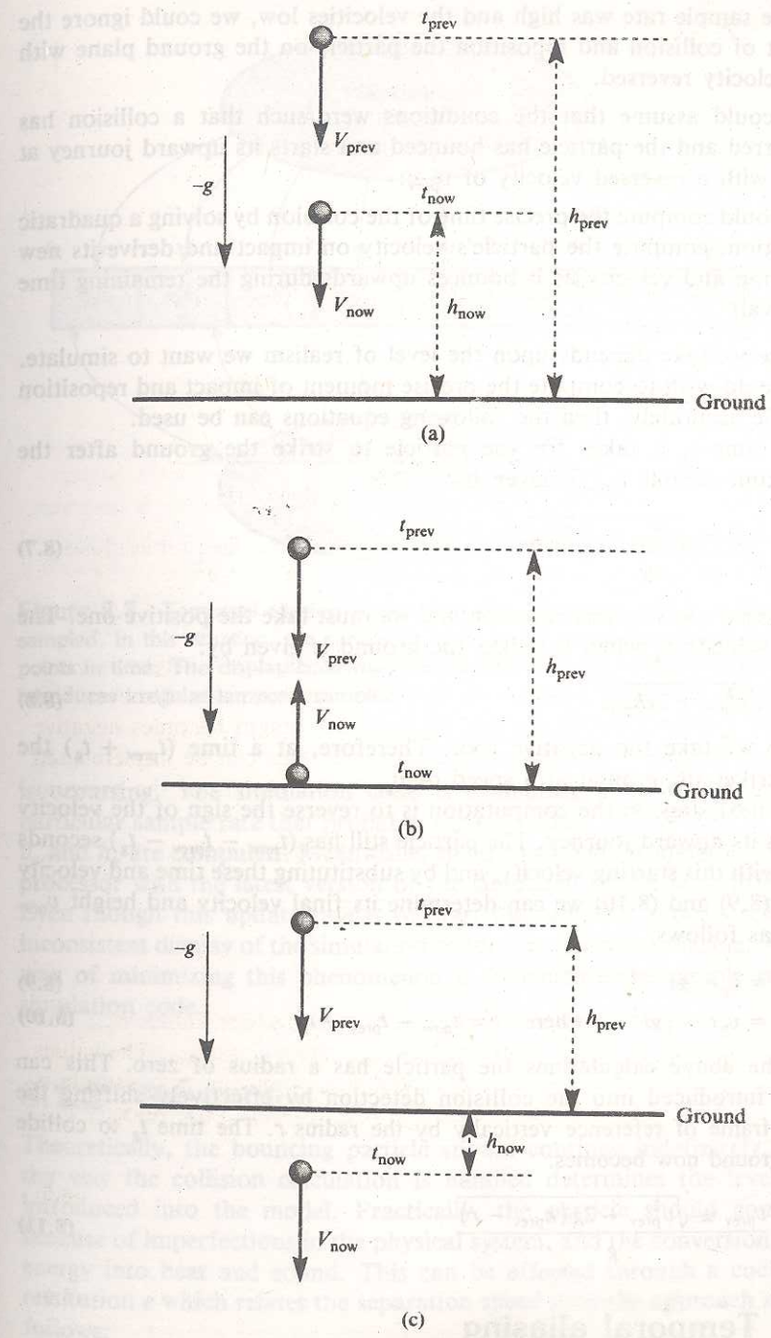

Figure 8.2 (a) The particle is still falling, therefore $h_{now} > 0$. (b) The particle has hit the ground, therefore $h_{now} = 0$. (c) The particle has effectively intersected the ground plane, therefore $h_{now} < 0$.

(1) If the sample rate was high and the velocities low, we could ignore the point of collision and reposition the particle on the ground plane with its velocity reversed.

(2) We could assume that the conditions were such that a collision has occurred and the particle has bounced and starts its upward journey at h_{prev} with a reversed velocity of v_{prev}.

(3) We could compute the precise time of the collision by solving a quadratic equation, compute the particle's velocity on impact and derive its new position and velocity as it bounces upwards during the remaining time interval!

The choice we take depends upon the level of realism we want to simulate.

If we do wish to compute the precise moment of impact and reposition the particle accurately, then the following equations can be used.

The time t_g it takes for the particle to strike the ground after the previous time sample t_{prev} is given by:

$$t_g = \frac{v_{prev} \pm \sqrt{v_{prev}^2 + 2gh_{prev}}}{g} \tag{8.7}$$

There are two roots to this equation and we must take the positive one. The particle's velocity v_g when it strikes the ground is given by:

$$v_g = \sqrt{v_{prev}^2 + 2gh_{prev}} \tag{8.8}$$

This time we take the negative root. Therefore, at a time $(t_{prev} + t_g)$ the particle strikes the ground at a speed of v_g.

The next stage in the computation is to reverse the sign of the velocity as it starts its upward journey. The particle still has $(t_{now} - t_{prev} - t_g)$ seconds to travel with this starting velocity, and by substituting these time and velocity values in (8.9) and (8.10) we can determine its final velocity and height v_{now} and h_{now} as follows:

$$v_{now} = v_g - gt \tag{8.9}$$

$$h_{now} = v_g t - \tfrac{1}{2}gt^2 \quad \text{where} \quad t = t_{now} - t_{prev} - t_g \tag{8.10}$$

In the above calculations the particle has a radius of zero. This can easily be introduced into the collision detection by effectively shifting the particle's frame of reference vertically by the radius r. The time t_g to collide with the ground now becomes:

$$t_g = \frac{v_{prev} \pm \sqrt{v_{prev}^2 + 2g(h_{prev} - r)}}{g} \tag{8.11}$$

8.1.1 Temporal aliasing

The above example raises a fundamental problem associated with all discrete sampling systems, namely, aliasing. Figure 8.3 shows diagrammatically what

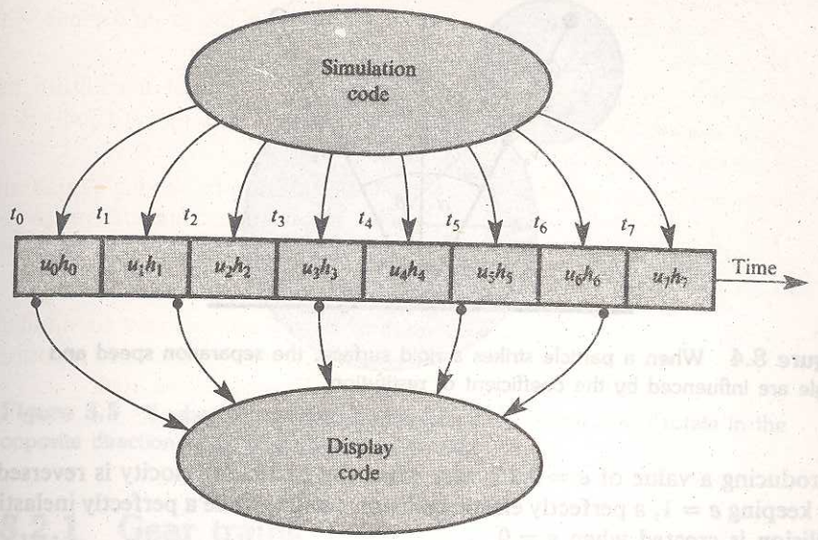

Figure 8.3 Temporal aliasing arises when any continuous system is discretely sampled. In this situation, the simulation code computes parameters and different points in time. The display code may sample this data on a regular basis, which introduces irregular temporal sampling.

is occurring. The simulation code is evaluating a numerical model at a particular sample rate that probably varies in time. At any time t_n, values of v_n and h_n are computed. Meanwhile, the display code is updating the display processor with the latest version of the simulated model at a different rate. Even though this update rate is regular, the sampling process results in an inconsistent display of the simulation producing temporal aliasing. A practical way of minimizing this phenomenon is to increase the sample rate of the simulation code.

8.1.2 Restitution

Theoretically, the bouncing particle should continue indefinitely, however, the way the collision calculation is handled determines the level of error introduced into the model. Practically, the particle should come to rest because of imperfections in the physical system, and the conversion of kinetic energy into heat and sound. This can be effected through a coefficient of restitution e which relates the separation speed v_s to the approach speed v_a as follows:

$$v_s = ev_a \quad \text{where} \quad 0 \leqslant e \leqslant 1 \tag{8.12}$$

In the bouncing particle scenario we could attenuate the bouncing cycle by

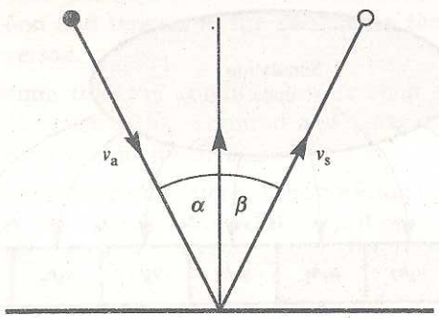

Figure 8.4 When a particle strikes a rigid surface, the separation speed and angle are influenced by the coefficient of restitution.

introducing a value of $e = 0.95$, say, when the particle's velocity is reversed. By keeping $e = 1$, a perfectly elastic collision occurs, while a perfectly inelastic collision is created when $e = 0$.

Figure 8.4 shows a particle striking a smooth rigid surface with an approach speed v_a at an angle α, and a separation speed of v_s at an angle of β. The component of its velocity tangential to the surface remains constant, whereas the magnitude of the normal component is influenced by the coefficient of restitution as follows:

$$v_s \sin\beta = v_a \sin\alpha \quad \text{(tangential component)} \tag{8.13}$$

$$v_s \cos\beta = e v_a \cos\alpha \quad \text{(normal component)} \tag{8.14}$$

Therefore:

$$e \tan\beta = \tan\alpha \tag{8.15}$$

and:

$$v_s^2(\sin^2\beta + \cos^2\beta) = v_a^2(\sin^2\alpha + e^2\cos^2\alpha) \tag{8.16}$$

Therefore:

$$v_s = v_a\sqrt{\sin^2\alpha + e^2\cos^2\alpha} \tag{8.17}$$

and:

$$\beta = \tan^{-1}((\tan\alpha)/e) \tag{8.18}$$

8.2 Rotating wheels

In Chapter 7 we saw how rotational movements are achieved using matrix techniques, so let us now examine some of the practical issues of simulating rotating elements in a VE.

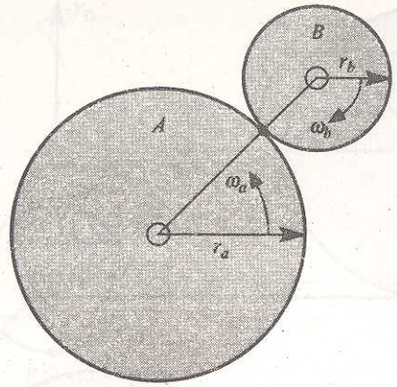

Figure 8.5 If wheel A is rotating at ω_a[rad s^{-1}], wheel B will rotate in the opposite direction at a rate proportional to r_a/r_b.

8.2.1 Gear trains

Consider the problem of animating the rotational speed of a wheel in contact with some driving wheel. Figure 8.5 illustrates two wheels A and B, with radii r_a and r_b, and rotational rates ω_a and ω_b rad s^{-1} respectively. At the point of contact P, the angular speed of wheel A is $\omega_a r_a$, which must be the same for wheel B, that is if we assume zero slip between the two wheels. However, as wheel A is rotating anticlockwise, wheel B is driven in the opposite direction, therefore the following relationship holds:

$$\omega_b r_b = -\omega_a r_a \tag{8.19}$$

and:

$$\omega_b = \frac{-r_a\omega_a}{r_b}\ [\text{rad s}^{-1}] \tag{8.20}$$

If t seconds is the elapsed time from some reference time, wheel A will rotate $\omega_a t$ radians and wheel B will rotate $-\omega_a t r_a/r_b$ radians which can then be substituted into the relevant matrix used to rotate wheel B.

8.2.2 Combined linear and rotational motion

If a wheel or ball rolls across a surface, there is a simple relationship linking the linear and rotational speeds. Figure 8.6 shows a wheel of radius r whose centre is travelling at a speed v. The linear distance d travelled after t seconds is given by:

$$d = vt \tag{8.21}$$

The angle θ turned by the wheel is:

(Note that the angle of rotation θ is negative, because the wheel rotates clockwise relative to the OCS), which simplifies to:

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.26)$$

and the two matrices concatenate to:

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & d \\ -\sin\theta & \cos\theta & 0 & r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.27)$$

For instance, say $v = 2\pi r$, $(t_1 - t_0) = 1$ and $r = 1$. The distance travelled is $d = 2\pi$, and the rotated angle is 360°. Substituting these values in (8.27) produces:

$$\begin{bmatrix} 1 & 0 & 0 & 2\pi \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.28)$$

which confirms that a point $P(0,-1,0)$ is transformed to $P'(2\pi,0,0)$ after 1 second, as shown in Figure 8.7(b).

8.2.3 A steerable wheel

Let us now consider simulating a steerable wheel that can roll about the ground plane under the control of some external agent. This will probably be the VO who is controlling the wheel's speed and direction through some interface. The system samples the wheel's linear speed v, and steering angle ϕ, at some frequency, which hold until the next sample is made.

The wheel is modelled as shown in Figure 8.8(a) with its centre at the origin of the OCS, and rotates about the Z_O-axis. Three matrices are needed to position the wheel in the VE: the first matrix rotates the wheel about the Z_O-axis of the OCS by the accumulated roll angle α (Figure 8.8b); the second matrix rotates the wheel about the Y_O-axis by the current steering angle θ (Figure 8.8c); and the third matrix translates the wheel (x,r,z) in the VE. At time t_0 the wheel is located in the VE as shown in Figure 8.9, with the following conditions:

r is the wheel's radius

α is the accumulated angle rolled by the wheel

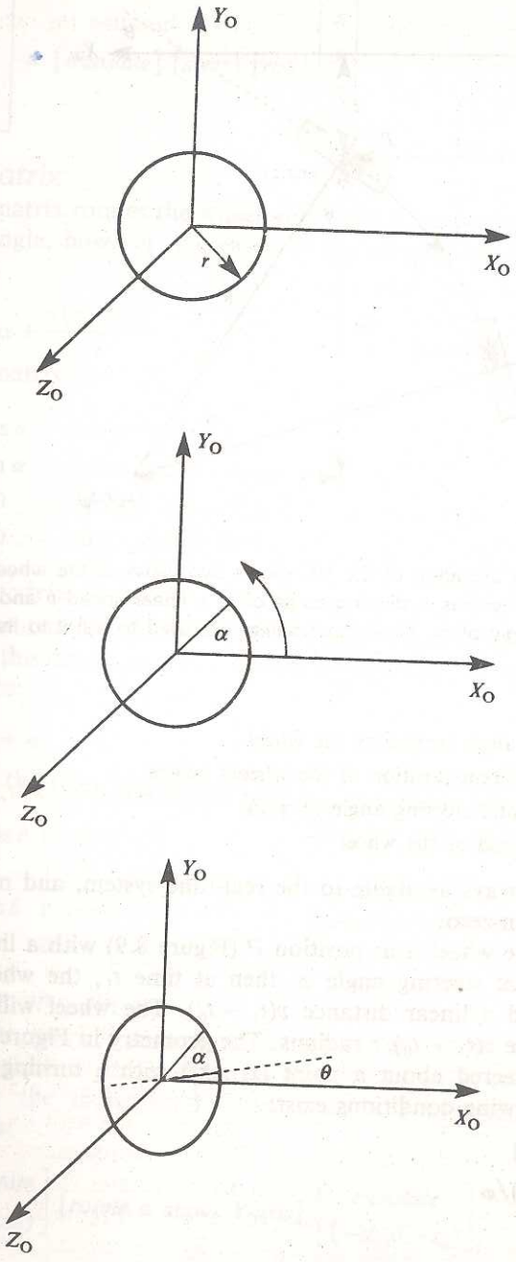

Figure 8.8 (a) A wheel, radius r, is modelled with its centre at the origin of the OCS. (b) It is rotated α about the Z_O axis to simulate rolling. (c) It is then rotated θ about the Y_O axis to simulate the steering angle.

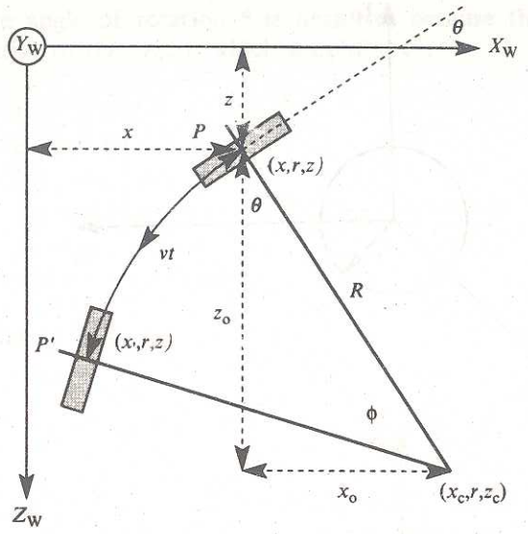

Figure 8.9 This plan elevation of the VE shows two views of the wheel at P and P'. Initially the wheel has a steering angle of θ, a linear speed v and an incremental steering angle of ϕ. These parameters are used to roll it to its next position at P'

θ is the steering angle turned by the wheel
$P(x,r,z)$ is the current position of the wheel's centre
ϕ is the incremental steering angle ($\phi \neq 0$)
v is the linear speed of the wheel

Both ϕ and v are always available to the real-time system, and note that ϕ is assumed to be non-zero.

If at time t_0 the wheel is at position P (Figure 8.9) with a linear speed v and an incremental steering angle ϕ, then at time t_1, the wheel will be at P', having rolled a linear distance $v(t_1 - t_0)$. The wheel will therefore roll through an angle $v(t_1 - t_0)/r$ radians. The geometry in Figure 8,9 shows that the wheel is steered about a point (x_c,r,z_c) with a turning radius R. Therefore, the following conditions exist:

$$R\phi = v(t_1 - t_0) \tag{8.29}$$
$$R = v(t_1 - t_0)/\phi \tag{8.30}$$
$$x_o = R\sin\theta \tag{8.31}$$
$$z_o = R\cos\theta \tag{8.32}$$
$$x_c = x + x_o = x + R\sin\theta \tag{8.33}$$
$$z_c = z + z_o = z + R\cos\theta \tag{8.34}$$

We now have sufficient information to develop the three matrices that will be applied to every wheel vertex in the following order:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = [translate]\,[steer]\,[roll] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{8.35}$$

Roll matrix

The roll matrix rotates the wheel about the Z_O-axis of the OCS, by an angle α. This angle, however, increases by the rolled angle between each update, therefore:

$$\alpha \leftarrow \alpha + \frac{v(t_1 - t_0)}{r} \tag{8.36}$$

The roll matrix is:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.37}$$

Steer matrix

The steer matrix rotates the wheel about the Y_O-axis of OCS by an angle θ. However, this angle increases by the incremental steering angle ϕ between each update:

$$\theta \leftarrow \theta + \phi \tag{8.38}$$

Therefore, the steer matrix becomes:

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.39}$$

Translate matrix

The translate matrix translates the wheel into the VE by the new position of the wheel, which is obtained by rotating the previous position (x,r,z) about $(x_c,0,z_c)$ by the incremental steering angle ϕ. In matrix terms, this is expressed as:

$$\begin{bmatrix} translate \\ (x_c,r,z_c) \end{bmatrix} [rotate\ \phi\ about\ Y\text{-}axis] \begin{bmatrix} translate \\ (-x_c,0,-z_c) \end{bmatrix} \tag{8.40}$$

$$\begin{bmatrix} 1 & 0 & 0 & x_c \\ 0 & 1 & 0 & r \\ 0 & 0 & 1 & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.41}$$

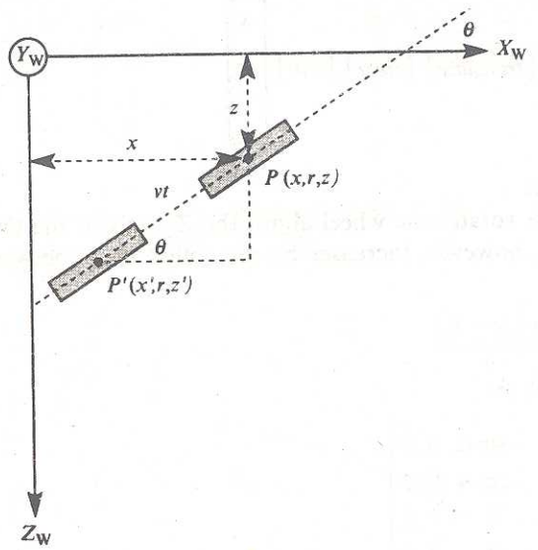

Figure 8.10 This plan elevation of the VE shows two views of the wheel at P and P'. The wheel has a steering angle of θ, a linear speed v and an incremental steering angle of zero. Its new position at P' depends only upon v, t and θ.

The above geometric reasoning holds only for non-zero values of ϕ. If the controlling agent sets this input value to zero, it implies that the wheel should continue forward in the current direction defined by θ. For this to occur, the wheel is still subjected to the roll and steer matrices, but the translation matrix is modified. Figure 8.10 shows the relevant geometry. If at time t_0 the wheel is at $P(x,r,z)$, travelling at a speed v with a steering angle θ, then at time t_1 the wheel will be at $P'(x',r,z')$, having rolled a linear distance $v(t_1 - t_0)$. Therefore:

$$x' = x - v(t_1 - t_0)\cos\theta \tag{8.42}$$

$$z' = z + v(t_1 - t_0)\sin\theta \tag{8.43}$$

These values are then substituted into the translate matrix (8.41) as before.

There are some interesting features of this example worth exploring. To begin with, the above matrices provide the basis for developing a steerable vehicle that can be driven inside a VE. The vehicle could be controlled by an agent in the real world, with the vehicle's movements observed on a computer display. The vehicle could even be controlled by a VO immersed in the VE using a control device similar to that used for radio-controlled model cars. The VO could be positioned inside the vehicle and experience the changing views of the VE as the vehicle moves about. The latter scenario could form the basis for a VR-based car simulator.

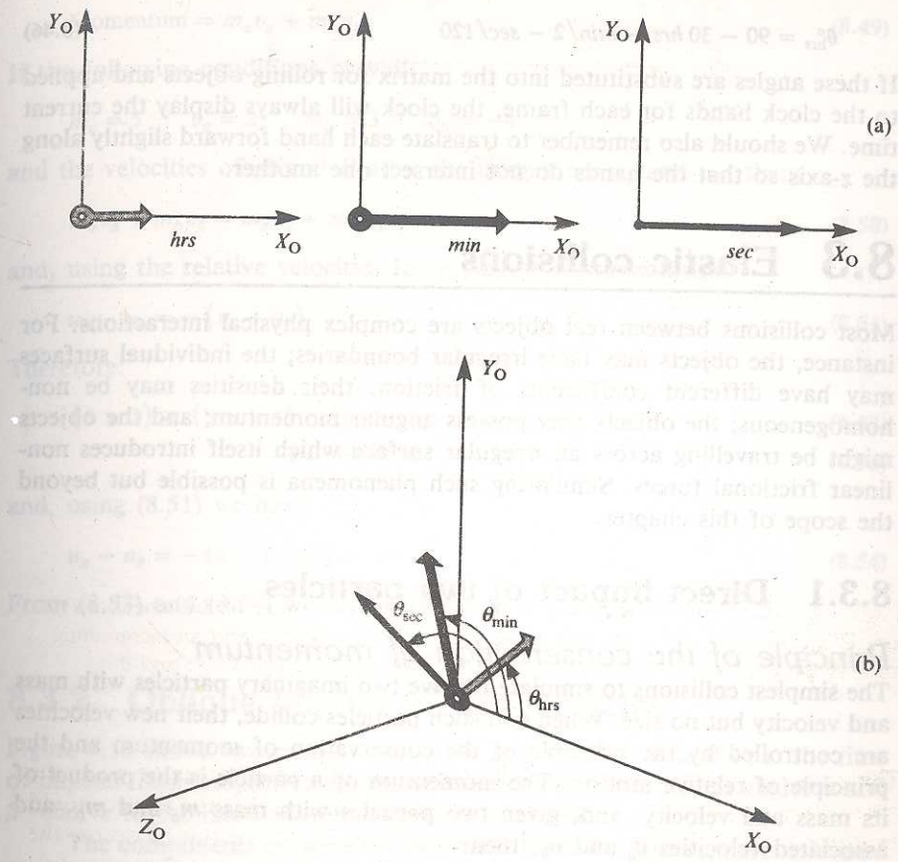

Figure 8.11 (a) The hour, minute and second hands. (b) The three angles θ_{sec}, θ_{min} and θ_{hrs} are used to rotate the individual hands about the Z_O-axis.

8.2.4 Clocks

Consider the problem of animating a clock to show the current time. Basically, we need to develop the matrices to rotate the clock's hour, minute and second hands. These hands are modelled as shown in Figure 8.11. For this example we will assume that the current time is available in the form of (hrs, min, sec), which will be used to derive $(\theta^\circ_{hrs}, \theta^\circ_{min}, \theta^\circ_{sec})$. As the second hand is initially pointing at the '15 second' position, θ°_{sec} is given by:

$$\theta^\circ_{sec} = 90 - 6\,sec \tag{8.44}$$

As the minute hand is pointing at the '15 minute' position, θ°_{min} is given by:

$$\theta^\circ_{min} = 90 - 6\,min - sec/10 \tag{8.45}$$

Similarly, θ°_{hrs} is given by:

$$\theta^{\circ}_{hrs} = 90 - 30\,hrs - min/2 - sec/120 \tag{8.46}$$

If these angles are substituted into the matrix for rolling objects and applied to the clock hands for each frame, the clock will always display the current time. We should also remember to translate each hand forward slightly along the z-axis so that the hands do not intersect one another.

8.3 Elastic collisions

Most collisions between real objects are complex physical interactions. For instance, the objects may have irregular boundaries; the individual surfaces may have different coefficients of friction; their densities may be non-homogeneous; the objects may possess angular momentum; and the objects might be travelling across an irregular surface which itself introduces non-linear frictional forces. Simulating such phenomena is possible but beyond the scope of this chapter.

8.3.1 Direct impact of two particles

Principle of the conservation of momentum

The simplest collisions to simulate involve two imaginary particles with mass and velocity but no size. When two such particles collide, their new velocities are controlled by the principle of the conservation of momentum and the principle of relative motion. The momentum of a particle is the product of its mass and velocity, and, given two particles with mass m_a and m_b, and associated velocities v_a and v_b, then:

$$m_a v_a + m_b v_b = \text{constant} \tag{8.47}$$

The principle of relative motion

The principle of relative motion states that the relative velocity after an impact equals the relative velocity before the impact multiplied by the coefficient of restitution. Therefore, if the respective velocities of the two particles after the collision are u_a and u_b, then:

$$u_a - u_b = -e(v_a - v_b) \tag{8.48}$$

where e is the coefficient of restitution.

When two bodies collide, the interaction is assumed to take place during a short time interval. The forces generated during the collision are assumed to be much larger than any external forces such as gravity and wind resistance. Furthermore, any two particles are assumed to move in the same straight line connecting their centres. Such collisions are called 'direct impact'.

To illustrate the principle of the conservation of momentum and relative velocity, consider the case of particle A with mass m_a and velocity v_a striking particle B with mass m_b and velocity v_b, travelling in opposite directions. Their total momentum is given by:

$$\text{Momentum} = m_a v_a + m_b v_b \tag{8.49}$$

If the following conditions prevail:

$$m_a = 5 \quad v_a = 2 \quad m_b = 3 \quad v_b = -4$$

and the velocities of A and B after the impact are u_a and u_b, then:

$$m_a v_a + m_b v_b = m_a u_a + m_b u_b \tag{8.50}$$

and, using the relative velocities for a perfect elastic collision:

$$u_a - u_b = -(v_a - v_b) \tag{8.51}$$

Therefore:

$$(5 \times 2) + (3 \times -4) = 5u_a + 3u_b \tag{8.52}$$

$$-2 = 5u_a + 3u_b \tag{8.53}$$

and, using (8.51) we have:

$$u_a - u_b = -(2 - (-4)) = -6 \tag{8.54}$$

From (8.53) and (8.54) we discover that $u_a = -2.5$, and $u_b = 3.5$.

8.3.2 Oblique impact of two particles

Figure 8.12 shows two particles A and B positioned at c_a and c_b at the time of impact. Their respective approach velocities are v_a and v_b at angles α and β relative to the line connecting their two centres.

The components of velocity perpendicular to the line $c_a c_b$ are unaltered by the collision, and are given by $v_a \sin\alpha$ and $v_b \sin\beta$ respectively. The components of velocity parallel to the line $c_a c_b$ are subject to the conservation of momentum because the particles' masses are accelerated. Therefore:

$$m_a v_a + m_b v_b = m_a u_a \cos\alpha + m_b u_b \cos\beta \tag{8.55}$$

By Newton's law of relative motion,

$$u_a - u_b = -e(v_a \cos\alpha - v_b \cos\beta) \tag{8.56}$$

where e is the coefficient of restitution. Substituting (8.56) in (8.55) we have:

$$u_a = \frac{(m_a - em_b)v_a \cos\alpha + m_b v_b \cos\beta(1 + e)}{m_a + m_b} \tag{8.57}$$

$$u_b = \frac{(m_b - em_a)v_b \cos\beta + m_a v_a \cos\alpha(1 + e)}{m_a + m_b} \tag{8.58}$$

The final velocity of each particle and its direction of motion is found from the perpendicular and parallel components.

If particle B is at rest, the final velocities are given by:

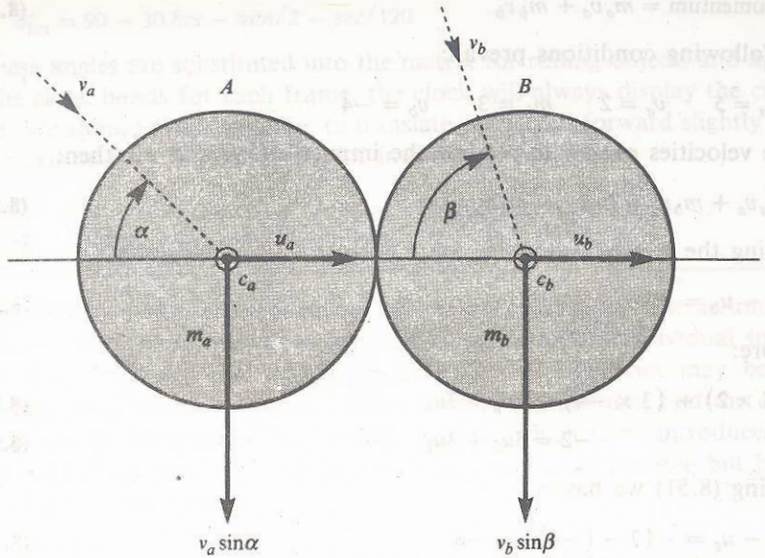

Figure 8.12 When two particles A and B collide with velocities v_a and v_b, at angles α and β to the line joining their two centres, the components of velocity perpendicular to the line are unaltered. The tangential components are determined by applying the principle of the conservation of momentum.

$$u_a = \frac{(m_a - em_b)v_a \cos\alpha}{m_a + m_b} \tag{8.59}$$

$$u_b = \frac{m_a v_a \cos\alpha (1 + e)}{m_a + m_b} \tag{8.60}$$

Furthermore, its velocity after impact must be directed parallel to the line $c_a c_b$ as its perpendicular velocity component was zero.

If the new direction of the colliding particle A is defined in terms of an angle β relative to the line $c_a c_b$, then:

$$\tan\theta = \frac{v_a \sin\alpha}{u_a} = \frac{(m_a + m_b)\tan\alpha}{m_a - em_b} \tag{8.61}$$

If the two particles have the same mass, then:

$$\tan\theta = \frac{2\tan\alpha}{1 - e} \tag{8.62}$$

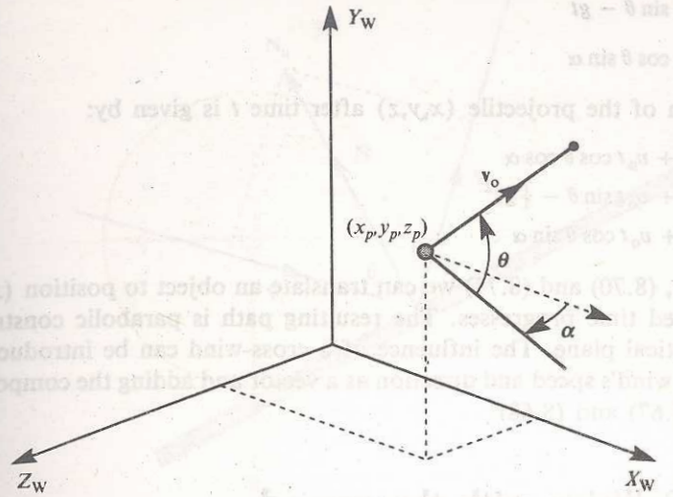

Figure 8.13 The source of the projectiles is (x_p, y_p, z_p), and two angles α and θ are used to position the exit vector.

8.4 Projectiles

Shells and bullets are the first things we associate with projectiles, but any object projected into the air is a projectile, be it a coin, stone or table. An inflated balloon, however, cannot be considered as a projectile as its buoyancy and air resistance are significant factors influencing its motion. For the following analysis it will be assumed that the frictional force due to air resistance is insignificant, and that the earth's gravitational force is constant at all heights.

8.4.1 The motion of a projectile

In order to develop a general solution to the trajectory of a projectile, let us employ the scenario shown in Figure 8.13, where an object is projected from the position (x_p, y_p, z_p) with velocity v_o and orientation angles θ and α.

The initial velocity components are given by:

$$v_x = v_o \cos\theta \cos\alpha \tag{8.63}$$

$$v_y = v_o \sin\theta \tag{8.64}$$

$$v_z = v_o \cos\theta \sin\alpha \tag{8.65}$$

After time t after projection, the velocity components are given by:

$$v_x = v_o \cos\theta \cos\alpha \tag{8.66}$$

$$v_y = v_o \sin\theta - gt \tag{8.67}$$

$$v_z = v_o \cos\theta \sin\alpha \tag{8.68}$$

The position of the projectile (x,y,z) after time t is given by:

$$x = x_p + v_o t \cos\theta \cos\alpha \tag{8.69}$$

$$y = y_p + v_o t \sin\theta - \tfrac{1}{2} g t^2 \tag{8.70}$$

$$z = z_p + v_o t \cos\theta \sin\alpha \tag{8.71}$$

Using (8.69), (8.70) and (8.71) we can translate an object to position (x,y,z) as the elapsed time progresses. The resulting path is parabolic constrained within a vertical plane. The influence of a cross-wind can be introduced by defining the wind's speed and direction as a vector and adding the components to (8.66), (8.67) and (8.68).

8.4.2 Collision with the ground

Earlier in Section 8.1 we investigated the problem of detecting when a falling object struck the ground. It was shown that different strategies could be adopted depending upon the level of accuracy required. The same applies in this situation. If we maintain the status of the projectile at each sample, we can identify the precise moment in time when the projectile's height becomes zero. At this point we can determine the impact velocity. Rather than repeat this analysis again, we will continue and consider the behaviour of the projectile if it bounces off the surface.

As the ground plane is a specific case, an arbitrarily oriented surface will be used to derive the bounce vector. Figure 8.14 shows a surface whose orientation is specified by its normal vector N. The incident unit vector is $\mathbf{V}_i$ and the bounce unit vector is $\mathbf{V}_b$. If restitution is ignored, the angle of incidence θ equals the angle of bounce. From Figure 8.14 it can be seen that the surface normal vector $\mathbf{N}_u$ is given by:

$$\mathbf{N}_u = -\mathbf{V}_i + \mathbf{V}_b \tag{8.72}$$

$$-\mathbf{V}_i \cdot \mathbf{V}_b = \cos 2\theta = 2\cos^2\theta - 1 \tag{8.73}$$

$$\frac{|\mathbf{N}_u|}{2} = |-\mathbf{V}_i| \cos\theta \tag{8.74}$$

but as $\mathbf{V}_i$ is a unit vector:

$$|\mathbf{N}_u| = 2\cos\theta \tag{8.75}$$

Using the dot product definition:

$$-\mathbf{V}_i \cdot \mathbf{N} = \cos\theta \tag{8.76}$$

Therefore:

$$|\mathbf{N}_u| = 2(-\mathbf{V}_i \cdot \mathbf{N}) \tag{8.77}$$

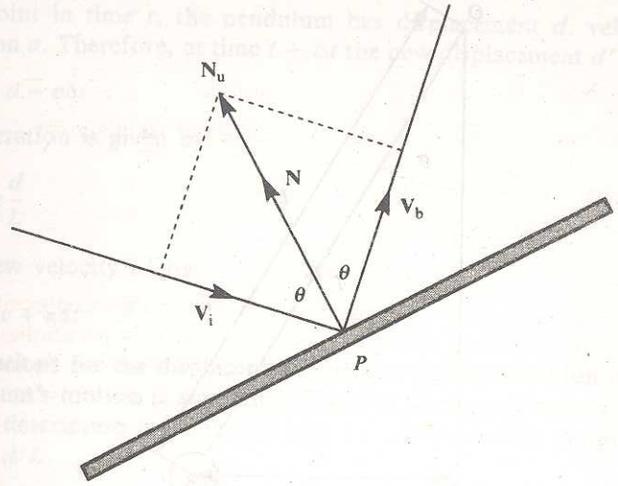

Figure 8.14 The relationship between the incident unit vector $\mathbf{V}_i$, the unit surface normal vector $\mathbf{N}$ and the unit bounce vector $\mathbf{V}_b$.

but:

$$\mathbf{N} = \frac{\mathbf{N}_u}{|\mathbf{N}_u|} = \frac{-\mathbf{V}_i + \mathbf{V}_b}{2(-\mathbf{V}_i \cdot \mathbf{N})} \tag{8.78}$$

and:

$$2(-\mathbf{V}_i \cdot \mathbf{N})\mathbf{N} = -\mathbf{V}_i + \mathbf{V}_b \tag{8.79}$$

Therefore, the bounce vector is given by:

$$\mathbf{V}_b = \mathbf{V}_i + 2(-\mathbf{V}_i \cdot \mathbf{N})\mathbf{N} \tag{8.80}$$

It is left to the reader to implement this geometry and introduce the effect of restitution.

8.5 Simple pendulums

It can be shown that a simple pendulum exhibits approximate simple harmonic motion if it consists of a heavy concentrated mass suspended by an inextensible cord and restricted to a swing of up to ±14°. To simulate this behaviour dynamically we must analyse the forces shown in Figure 8.15. In this figure we see that the mass of the pendulum, m, is supported by a cord of length L. If the mass is displaced from its upright position by an angle θ radians, a restoring force $m \sin\theta$ will attempt to re-establish equilibrium. The acceleration a produced by this restoring force is given by:

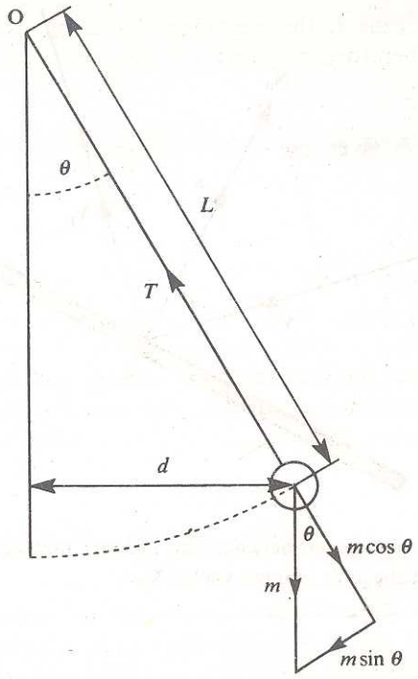

Figure 8.15 When a pendulum is displaced by an angle θ from the vertical position, a restoring force $m \sin\theta$ attempts to re-establish equilibrium. The future position of the pendulum is determined by computing its velocity and acceleration from this force.

$$a = \frac{m \sin\theta}{m/g} \tag{8.81}$$

$$a = g \sin\theta \tag{8.82}$$

If the angle of swing does not exceed ±14°, $\sin\theta$ is approximately equal to θ radians. Therefore, we can write:

$$a = g\theta \tag{8.83}$$

But for these small angles:

$$\theta = \frac{d}{L} \tag{8.84}$$

Therefore:

$$a = g\frac{d}{L} \tag{8.85}$$

At any point in time t, the pendulum has displacement d, velocity v and acceleration a. Therefore, at time $t + \Delta t$ the new displacement d' is given by:

$$d' = d - v\Delta t \tag{8.86}$$

The acceleration is given by:

$$a = g\frac{d}{L} \tag{8.87}$$

and the new velocity v' by:

$$v' = v + a\Delta t \tag{8.88}$$

If the equations for the displacement, velocity and acceleration are iterated, the pendulum's motion is simulated. The pendulum is animated by rotating the object description through and angle θ relative to the upright position, where $\theta = d/L$.

8.6 Springs

The motion of an object attached to a spring or a piece of elastic can be simulated by analysing the dynamic forces acting upon the object. Figure 8.16 shows a spring of length l hanging from a rigid fixing. If an object of mass m is attached to the free end of the spring, it will bounce up and down with simple harmonic motion. Two laws are used to describe the active forces, namely Newton's second law of motion, and Hooke's law. Hooke's law states that the tension T in a spring is given by:

$$T = \lambda\frac{e}{l} \tag{8.89}$$

where λ is the modulus of the spring and e is the extension of the spring. While the object is moving up and down, its motion is controlled by:

$$ma = mg - T \tag{8.90}$$

where m is the mass of the object, a is the acceleration of the object and g is the acceleration due to gravity.

If at any time t, the mass has position y, velocity v and acceleration a, at a time $t + \Delta t$ its new position y' relative to the top of the spring is given by:

$$y' = y + v\Delta t \tag{8.91}$$

The extension e is given by:

$$e = y - l \tag{8.92}$$

and if this is substituted into (8.93):

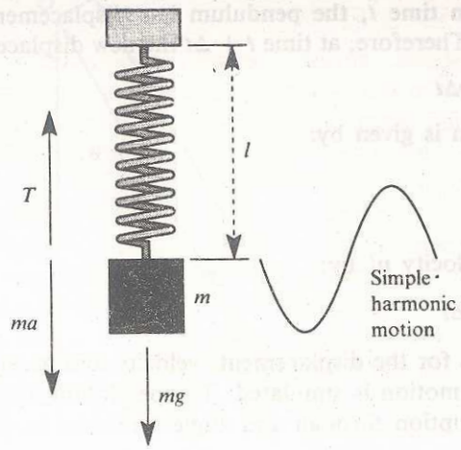

Figure 8.16 If a mass m is attached to a light spring it will oscillate with simple harmonic motion.

$$ma = mg - \lambda\frac{e}{l} \tag{8.93}$$

the acceleration a is equal to:

$$a = g - \lambda\frac{e}{lm} \tag{8.94}$$

and the new velocity v' of the mass is equal to:

$$v' = v + a\Delta t \tag{8.95}$$

If this value of v' is substituted back into (8.91) to develop a new value for y, and continuously repeated, the trace of y is simple harmonic. The value of y can then be substituted into a matrix to subject an object to this motion. Perhaps one of the most useful features of this approach is that the model can be dynamically modified while it is running. Any of the above parameters can be interactively changed, resulting in an almost instantaneous change in the motion.

8.6.1 Restitution

As there is no loss in this model, the simple harmonic motion continues indefinitely. If the motion is required to decay to zero, a restitution term k_d can be introduced in (8.95) as follows:

$$v' = k_d v + a\Delta t \qquad 0 \leqslant k_d \leqslant 1 \tag{8.96}$$

When $k_d = 1$, the motion continues unchecked, but when $k_d < 1$, the motion decays away.

8.6.2 Moving springs

Figure 8.17(a) shows a spring of length l connected to a wheel which can travel over a surface. On top of the spring is a mass m. If the spring's restitution is less than unity, when the mass is placed on the spring it will oscillate and settle down to some state of equilibrium. The mass will oscillate only if it is subjected to some external force.

One way of introducing a force is to roll the wheel over an undulating terrain as shown in Figure 8.17(b). As the wheel rises and falls, the spring is compressed and stretched, modulating the spring tension transmitted to the mass. The extra spring compression h is the difference between the terrain height sampled at $t + \Delta t$ and t. Initially h is set to zero and is introduced into the spring extension equation as follows:

$$e = y - l - h \tag{8.97}$$

where y is the height of the spring before the mass was added.

The new set of equations then become:

$$y' = y + v\Delta t$$
$$e = y - l - h$$
$$a = g - \lambda\frac{e}{lm} \tag{8.98}$$
$$v' = k_d v + a\Delta t$$
$$h' = h_{new} - h$$

Where h_{new} is the next value of h sampled at $t + \Delta t$. When this algorithm is implemented, it produces very realistic motion.

8.6.3 Elastic structures

We saw in Section 7.4 how free-form deformations (FFDs) are used by animators to form flexible objects. But if a sense of elasticity has to be introduced into the animation, the control points of the FFDs must be moved with suitable dynamics. We can use the dynamic model of a spring to create this effect, but the ideas developed below have also been applied to model flexible objects such as flags, carpets, curtains and tablecloths (Miller, 1984; Scanlon, 1990; Terzopoulos and Witkin, 1988; Weil, 1986).

Fabrics can be modelled as a matrix of mass points connected together in a mesh, where each mass is connected to a neighbouring mass using springs, as shown in Figure 8.18. The forces acting on the springs can be related to the masses they act upon using Newton's second law of motion.

To illustrate the technique we will analyse the 2D case, and it will be left to the reader to extend the ideas to the 3D domain, with reference to the papers cited.

Consider the dynamic simulation of an elastic thread fixed at both ends.

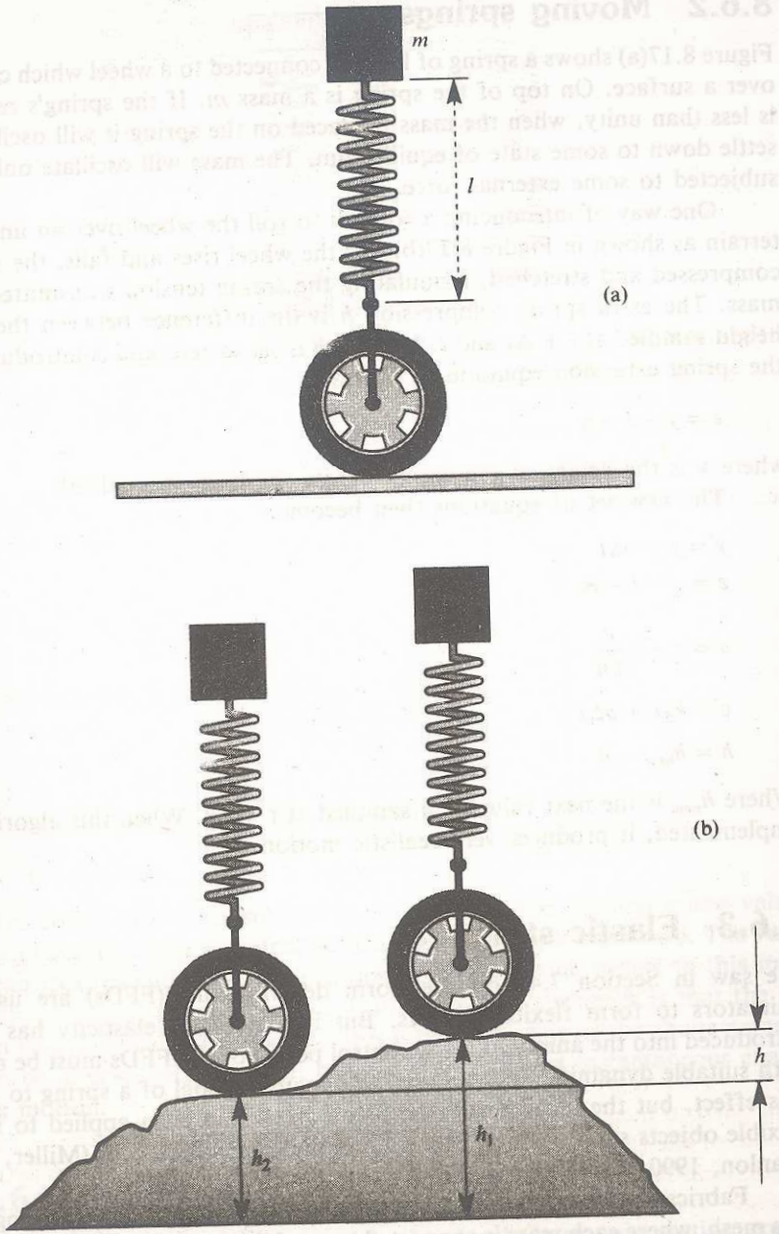

Figure 8.17 (a) If a mass m is attached to a light spring it will move with simple harmonic motion and settle down to a state of equilibrium. (b) As the wheel rises and falls, the mass attached to the free end of the spring will subject the spring to extra forces caused by accelerating the mass.

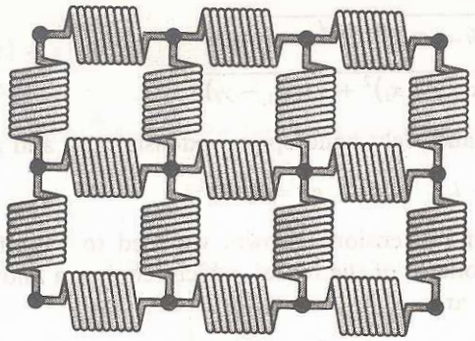

Figure 8.18 Fabric can be represented as a mesh of mass points connected to one another by stiff springs. The forces acting through these springs can be computed and used to move the mass point positions to simulate the behaviour of material under the forces of gravity.

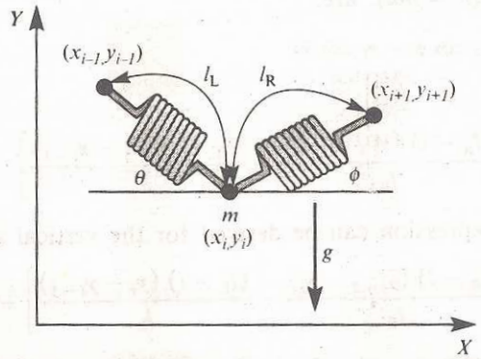

Figure 8.19 Given three mass points falling under the action of gravity (g), their changing positions in space can be determined by resolving the horizontal and vertical force components.

This can be represented approximately by a sequence of mass points m connected to their neighbours with springs of length l. Now imagine that the middle of the thread has been raised and allowed to fall under the action of gravity. Hooke's law implies that the spring's extension is proportional to the applied force. It is these forces that need to be calculated, but first the spring extension has to be computed.

Figure 8.19 shows the positions of the three mass points whose positions are specified by:

$$(x_{i-1},y_{i-1}) \quad (x_i,y_i) \quad \text{and} \quad (x_{i+1},y_{i+1}) \tag{8.99}$$

The left- and right-hand lengths l_L and l_R will be:

$$l_L = \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2} \tag{8.100}$$

$$l_R = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{8.101}$$

and the left- and right-hand spring extensions e_L and e_R will be:

$$e_L = l_L - l \quad \text{and} \quad e_R = l_R - l \tag{8.102}$$

With the spring extensions known, we need to identify the horizontal and vertical components of the forces, which requires a knowledge of the spring's angles. These are:

$$\cos\theta = (x_i - x_{i-1})/l_L \qquad \sin\theta = (y_i - y_{i-1})/l_L \tag{8.103}$$

and

$$\cos\phi = (x_{i+1} - x_i)/l_R \qquad \sin\phi = (y_{i+1} - y_i)/l_R \tag{8.104}$$

As the force exerted by a spring is proportional to its extension and stiffness S, the horizontal forces acting on the ith mass, using Newton's second law of motion ($F = ma$), are:

$$ma_h = S(e_R \cos\phi - e_L \cos\theta) \tag{8.105}$$

therefore:

$$a_h = \frac{S}{m}\left[\frac{(l_R - l)(x_{i+1} - x_i)}{l_R} - \frac{(l_L - l)(x_i - x_{i-1})}{l_L}\right] \tag{8.106}$$

and a similar expression can be derived for the vertical acceleration:

$$a_v = \frac{S}{m}\left[\frac{(l_R - l)(y_{i+1} - y_i)}{l_R} - \frac{(l_L - l)(y_i - y_{i-1})}{l_L}\right] - g \tag{8.107}$$

(8.106) and (8.107) describe the horizontal and vertical accelerations of the central mass point, and we now require to determine values of the x- and y-coordinates that satisfy them. Euler's method for numerical integration is useful for solving this type of problem, and assumes that the mass points move small discrete distances in equally small time intervals.

Acceleration is a measure of the rate of change of velocity, which in turn is a measure of the rate of change of displacement with time. On an incremental basis, the horizontal acceleration can be defined in terms of the horizontal velocity as follows:

$$a_h = \frac{v_h(t + \Delta t) - v_h(t)}{\Delta t} \tag{8.108}$$

where t is any point in time, and Δt is a small increment in t. The velocity at time $t + \Delta t$ is therefore:

$$v_h(t + \Delta t) = v_h(t) + \Delta t a_h \tag{8.109}$$

but the velocity can also be defined as:

$$v_h(t) = \frac{x(t + \Delta t) - x(t)}{\Delta t} \tag{8.110}$$

therefore the new horizontal position for the mass point (which is what we require) is given by:

$$x(t + \Delta t) = x(t) + \Delta t v_h(t) \tag{8.111}$$

At t_0, when $t = 0$, the horizontal velocity and acceleration are zero:

$$v_h(t_0) = 0 \quad \text{and} \quad a_h(t_0) = 0 \tag{8.112}$$

Therefore, we evaluate (8.106) relating horizontal forces with acceleration to discover the initial acceleration. This is substituted into (8.109) using some suitable value of Δt and an initial velocity $v_0(0)$ of zero. The new velocity $v_h(t + \Delta t)$ is substituted in (8.111) to reveal the new position of the mid mass point. A similar process is applied to (8.107) to derive the vertical displacement. When both the x- and y-displacements are known, the position of the mid mass point is updated and the cycle repeated. Values of the stiffness S, the gravitational acceleration g, and the mass m will have to be selected to create the desired movement.

When this algorithm is implemented, we discover that the central mass bounces around with a realistic sense of elasticity. In fact, as there is no opportunity for energy loss, the system refuses to decay; however, this can be introduced by a restitution term in (8.109), similar to that used in (8.96).

8.7 Flight dynamics of an aircraft

Hopefully, it requires little convincing that simulating the dynamic behaviour of a 100 ton aircraft taking off, flying and landing is a non-trivial exercise. Nevertheless, it is possible, and is a central component of a flight simulator. However, in this example we will not attempt to reproduce an actual flight simulator model, but will develop a simple flight model that could be used to fly an aircraft within a VE.

The numerical models used to simulate the forces acting upon the frame of an aircraft are complex and address the effects of the craft's aerodynamic characteristics, engines, the atmospheric environment, undercarriage and hydraulic/mechanical control system used by the pilot to fly the aircraft (Rolfe, 1991).

The aerodynamic characteristics are affected by the vehicle's weight, speed, altitude, centre of gravity, angle of attack, flight surface geometry and ground proximity effects. The forces transmitted through the undercarriage during take-off and landing are determined by the mechanical, hydraulic and pneumatic assemblies associated with these structures. Apart from simulating these effects, the mathematical model may even include the dynamic behaviour of brakes, tyres and the nose-wheel steering system.

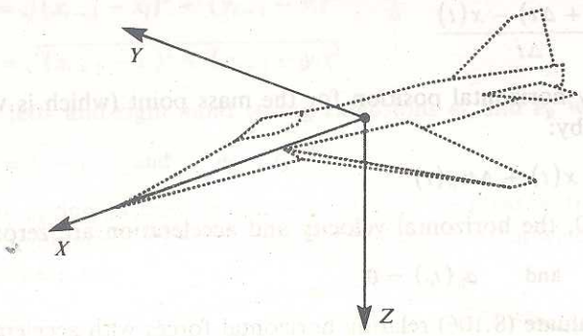

Figure 8.20 The traditional method of representing the axial system for an aircraft is a right-handed system with the Z-axis pointing downwards.

Different types of aeroengines possess different dynamic characteristics. They have varying thrust profiles and capacities; they burn fuel at different rates; they have differing temperature/efficiency characteristics and differing response patterns.

The atmospheric model reflects the variation of air density and temperature with height. Wind speed must be included as it can significantly affect a plane's true ground speed, as well as modify the craft's behaviour during landing and take-off. Wind shear, which creates highly dangerous flying conditions, must be modelled, and even the turbulence caused by large planes must be simulated as it can seriously affect the aerodynamics of planes following in their wake.

The mathematical techniques used to simulate such a complex system can be only an approximation of reality. But our understanding of these numerical simulation models is such that their accuracy and fidelity allow them to play a significant role in flight simulators. As this level of detail is relevant only to flight simulation, let us examine a simple first-order aircraft model.

8.7.1 A simple aircraft model

Figure 8.20 shows the traditional method of representing the axial frame of reference for an aircraft. It is right-handed, with the z-axis directed downwards. For the sake of consistency the axial system will be modified to conform with the examples in the rest of the book.

A first-order model for approximating an aircraft assumes that it is a point without mass, travelling with speed v in its body-fixed z-axis, and velocities v_x and v_y in the x- and y-axes respectively, as shown in Figure 8.21(a). The heading ϕ is specified by the angle between the z-axes of the aircraft and the WCS, as shown in Figure 8.21(b). For the moment, the aircraft is not allowed to yaw, pitch or roll.

Given the aircraft's position (x,y,z), speed v, acceleration a, and heading

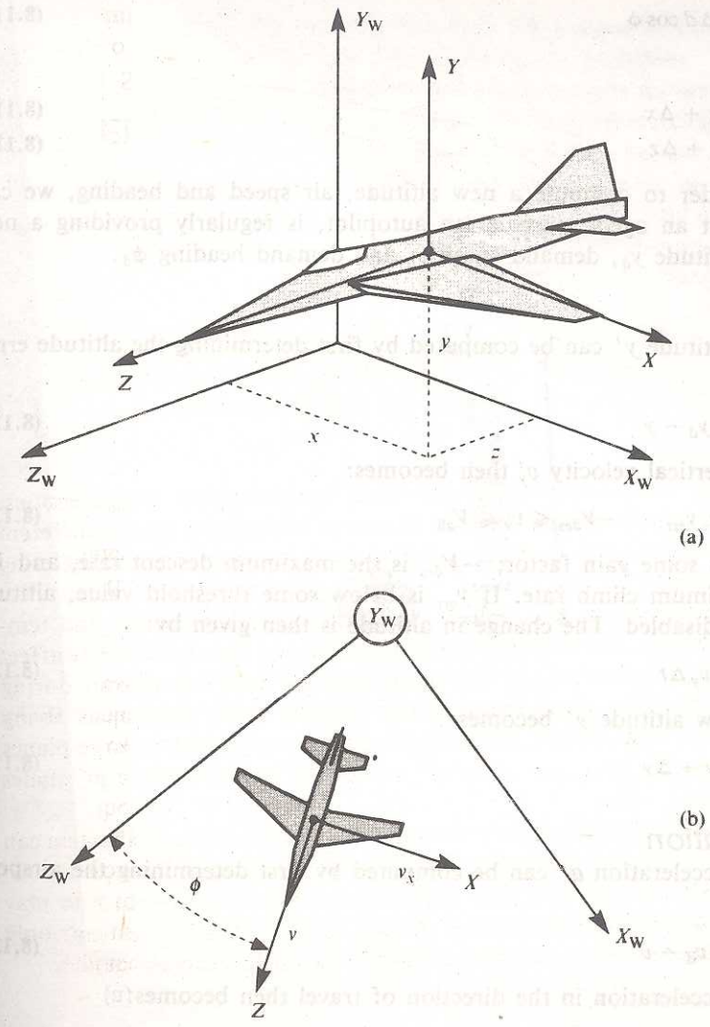

Figure 8.21 (a) A first-order flight model for an aircraft assumes that the craft is a point without mass, with instantaneous velocites v, v_x and v_y. (b) This plan elevation shows the definition of the heading angle ϕ.

ϕ, its new position (x',y',z'), after time Δt can be computed as follows. The incremental distance Δd moved forward is given by:

$$\Delta d = v\Delta t + \tfrac{1}{2} a\Delta t^2 \qquad (8.113)$$

The corresponding increments Δx and Δz in the x- and z-axes are given by:

$$\Delta x = \Delta d \sin\phi \qquad (8.114)$$

$$\Delta z = \Delta d \cos \phi \tag{8.115}$$

then:

$$x' = x + \Delta x \tag{8.116}$$

$$z' = z + \Delta z \tag{8.117}$$

In order to compute a new altitude, air speed and heading, we can assume that an agent, such as an autopilot, is regularly providing a new demand altitude y_d, demand speed v_d and demand heading ϕ_d.

Altitude

The new altitude y' can be computed by first determining the altitude error y_{err}:

$$y_{err} = y_d - y \tag{8.118}$$

The new vertical velocity v_y' then becomes:

$$v_y' = k_v y_{err} \qquad -V_{des} \leqslant v_y' \leqslant V_{cli} \tag{8.119}$$

where k_v is some gain factor; $-V_{des}$ is the maximum descent rate, and V_{cli} is the maximum climb rate. If y_{err} is below some threshold value, altitude acquire is disabled. The change in altitude is then given by:

$$\Delta y = v_y \Delta t \tag{8.120}$$

and the new altitude y' becomes:

$$y' = y + \Delta y \tag{8.121}$$

Acceleration

The new acceleration a' can be computed by first determining the airspeed error v_{err}:

$$v_{err} = v_d - v \tag{8.122}$$

The new acceleration in the direction of travel then becomes:

$$a' = k_a v_{err} \qquad -A_{dec} \leqslant a' \leqslant A_{acc} \tag{8.123}$$

where k_a is some gain factor, $-A_{dec}$ is the maximum deceleration permitted, and A_{acc} is the maximum acceleration rate permitted. If v_{err} is below some threshold value, airspeed acquire is disabled. The new airspeed is given by:

$$v' = v + a \Delta t \tag{8.124}$$

Heading

Finally, we come to the new heading ϕ'. Again we establish the error between the current heading and the demand heading

$$\phi_{err} = \phi_d - \phi \tag{8.125}$$

The new lateral velocity v_x' becomes:

$$v_x' = k_\phi \phi_{err} \qquad -V \leqslant v_x' \leqslant V \tag{8.126}$$

where k_ϕ is some gain factor and V is the maximum lateral velocity, given by $V = v \tan(M \Delta t)$, where M is the maximum turn rate (3° per second for standard turns). The change in heading is then given by:

$$\Delta\phi = \tan^{-1}\left(\frac{v_x}{v}\right) \tag{8.127}$$

and the new heading ϕ' becomes:

$$\phi' = \phi + \Delta\phi \tag{8.128}$$

After time Δt the aircraft has position (x',y',z'), speed v', acceleration a' and heading ϕ'. These become the new values of (x,y,z), v, a and ϕ respectively for the next time period Δt.

This flying model makes many assumptions. To begin with, the aircraft has no mass, it cannot rotate about any of its axes, and there is nothing in the model to simulate the action of landing and taking-off.

One simple mechanism for introducing a banking angle is to roll the aircraft about its z-axis by an angle proportional to the heading angle error ϕ_{err}. It is left to the reader to develop the model further by introducing pitch, yaw, mass and gravity.

In reality, the Euler equations of a rigid body moving in a vacuum are used as a starting point for a flight simulator model. These are sensitive to the aircraft's mass, the moments of inertia about the centre of gravity and the forces acting along the axes. Readers wishing to pursue this approach are forewarned of their relative complexity, and the axial notation used, which is different to that used above.