CHAPTER 5

Particle Choreography

In building a particle system simulation the primary intent is always to create an interesting visual effect, treating the system as a whole to achieve a coordinated bulk motion. To do this, we will need to think like animators, as well as engineers, since our desired goal goes beyond the technical problems of creating, simulating, and rendering particles. We need to also be able to choreograph the particles to create the motion we want. Figure 5.1 shows a frame from a highly choreographed sequence, where the particles are always under the control of the animator.

In his seminal paper *Particle Animation and Rendering Using Data Parallel Computation*, Sims [1990] laid the foundations of high-performance particle system simulation in a parallel computing environment, but the main emphasis of his paper is on the aesthetics of a particle simulation—he has a lot to say about choreography. A key idea that he introduced, informing work with particles to this day, is that we can think of the choreography of particles in terms of a set of operators affecting their motion. These fall into the categories of *initialization operators*, *acceleration operators*, and *velocity operators*. And, he adds to these the idea of *bounce*, i.e., particle collision response.

In Chapter 4 we covered the mechanics of creating and running a particle system, including initialization and bounce. In this chapter, we focus on the aspects of the choreography of particles involving the use of acceleration and velocity operators. We also look at methods for steering particles around objects to avoid collisions, and for creating wind fields.

5.1 ACCELERATION OPERATORS

An acceleration operator implements a choreography method that causes a change in the particle's velocity. Velocity changes can be created by summing forces, which when divided by mass yield acceleration, together with directly applied accelerations to yield a net acceleration. We will call the additive acceleration operators $\mathbf{a}^{+\mathrm{op}}$. These include the gravitational acceleration and air-resistance forces we studied in Chapter 3. Another possibility is to apply an operator to modify the velocity

T&F Cat #K23004 — K23004_C005 — page 65 — 8/23/2016 — 15:24

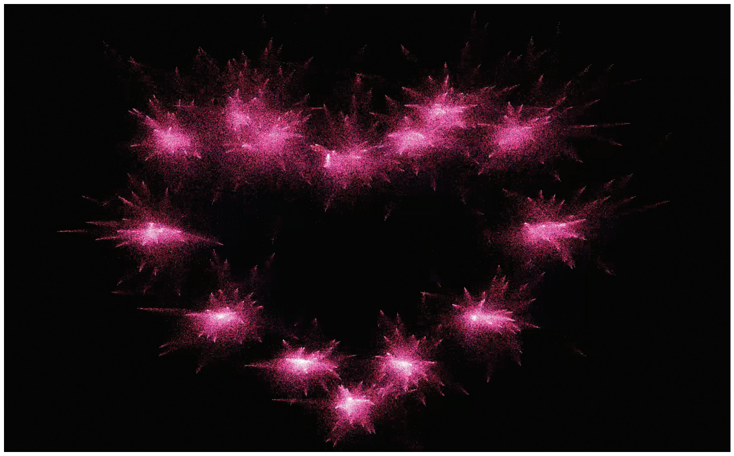


FIGURE 5.1 Fireworks particle system choreographed to create a valentine. (Courtesy by Chen-Jui Fang.)

obtained from numerical integration. These operators will typically take the velocity before and after integration as parameters and return an updated velocity. An example of this type of operator is the velocity update after collision as described in Section 4.4.3. We will call these operators $\mathbf{A}^{\mathrm{op}}()$, to indicate that they are functions. In the simulation loop, the effects of all active additive acceleration operators $\mathbf{a}_k^{+\mathrm{op}}$ on a particle i are summed, and the net acceleration is applied in the integration step to update particle i's velocity. Then, any functional acceleration operators are applied consecutively to the resulting velocity to yield a final modified velocity:

$$\begin{aligned}
\mathbf{a}_i^{\mathrm{net}} &= \sum_k \mathbf{a}_{ki}^{+\mathrm{op}}, \\
\mathbf{v}_{\mathrm{new}} &= \mathbf{v}_i^{[n]} + \mathbf{a}_i^{\mathrm{net}} h, \\
\mathbf{v}_{\mathrm{new}} &= \mathbf{A}_1^{\mathrm{op}}(\mathbf{v}_{\mathrm{new}}, \mathbf{v}_i^{[n]}), \quad \mathbf{v}_{\mathrm{new}} = \mathbf{A}_2^{\mathrm{op}}(\mathbf{v}_{\mathrm{new}}, \mathbf{v}_i^{[n]}), \quad \cdots \\
\mathbf{v}_i^{[n+1]} &= \mathbf{v}_{\mathrm{new}}
\end{aligned}$$

Below, we introduce several examples of acceleration operators. Readers are encouraged to be inventive, taking these examples as a starting place for their own designs.

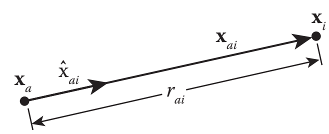
5.1.1 Gravitational Attractors

We have already seen one type of gravitational attractor. That is the attraction of a very large object on a very small object over a very constrained range of separations of the two objects, like the gravitational effect of the earth on a ball. In this case, we have the simplest form of additive acceleration operator, a constant vector,

$$\mathbf{a}^{+\mathrm{op}} = \mathbf{g}.$$

The large object does not move, but the small object moves toward the large object in the direction of the vector $\mathbf{g}$ and with its speed growing at the rate $\|\mathbf{g}\|$.

Gravity gets more interesting when the two objects are at a distance from each other, so that change of distance affects the gravitational attraction. This is the type of force that produces the choreography of the heavens. Planets orbit the sun, comets follow highly elliptical paths with orbits that take them near and then very far from the sun, and interstellar objects follow parabolic paths as they swoop into our solar system and then are thrown back out by the gravitational "slingshot" effect.



In a particle system, one of the objects will be a geometric object in a fixed position, and the other will be a moving particle. Let us call the fixed object the attractor a, give it mass m_a, and label its position $\mathbf{x}_a$. Now, consider this attractor's effect on particle i with position $\mathbf{x}_i$. We will call the vector from the attractor to the particle $\mathbf{x}_{ai} = \mathbf{x}_i - \mathbf{x}_a$, with magnitude $r_{ai} = \|\mathbf{x}_{ai}\|$ and direction $\hat{\mathbf{x}}_{ai} = \mathbf{x}_{ai}/r_{ai}$. According to Newtonian physical law, the force of attraction on particle i due to the attractor will be

$$\mathbf{f}_{ai} = -G\frac{m_a m_i}{r_{ai}^2}\hat{\mathbf{x}}_{ai},$$

where G is known as the universal gravitational constant. To arrive at the acceleration, not the force, on particle i, we divide by the particle's mass m_i. It is also convenient to multiply together Gm_a to yield a single constant G_a. This gives us the final additive acceleration operator

$$\mathbf{a}_{ai}^{+\mathrm{op}} = -G_a\frac{1}{r_{ai}^2}\hat{\mathbf{x}}_{ai}.$$

The effect of this operator is to accelerate particle i toward the position of the attractor a with an acceleration that is independent of the particle's mass and inversely proportional to the square of the distance between the particle and the attractor. The constant G_a can now be thought of as a "strength" constant that is freely tunable by the animator to adjust the gravitational effect.

We can move away from Newtonian physics a bit and give even more control to the animator by replacing the power 2 in the denominator with a constant p, yielding the additive acceleration operator

$$\mathbf{a}_{ai}^{+\mathrm{op}} = -G_a\frac{1}{r_{ai}^p}\hat{\mathbf{x}}_{ai}.$$

Setting $p = 2$ will give us the usual gravitational effect, resulting in particle i taking a parabolic or elliptical orbital path around attractor a, depending on particle i's velocity. Setting $p = -1$ will cause the operator to act like a spring, producing an acceleration on particle i that is proportional to its distance from the attractor.

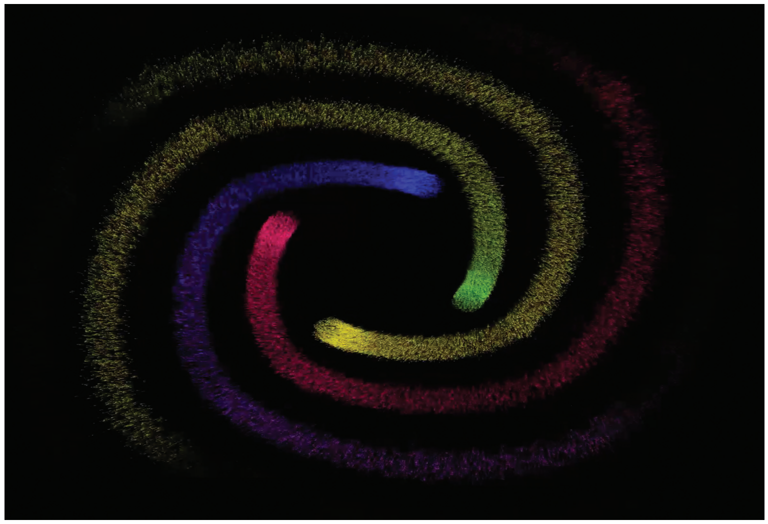
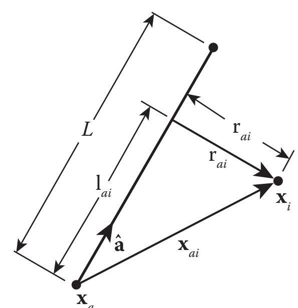


FIGURE 5.2 Spiraling particles choreographed using an attractor. (Courtesy by Kevin Smith and Adam Volin.)

This will cause the particle to be pulled toward the attractor, but it will "bounce" about the attractor like a ball on a rubber band. Other settings of p will produce different effects that are best explored by experimentation. Figure 5.2 shows an example of particle spiraling, created using a gravitational attractor.



If gravity can pull a particle toward a point, we could also define an attractor that pulls a particle toward a line with force affected by distance from the line. This could be an infinite line or a line segment of finite length. Let us see how this would be done for a line segment. We specify the line segment by a starting point $\mathbf{x}_a$, a unit vector $\hat{\mathbf{a}}$ in the direction of the line, and length L. Referring to the diagram to the right, consider this attractor's effect on particle i with position $\mathbf{x}_i$. The vector from the starting point of the line segment to particle i is $\mathbf{x}_{ai} = \mathbf{x}_i - \mathbf{x}_a$. The length of the projection of this vector onto the line segment's direction is $l_{ai} = \mathbf{x}_{ai} \cdot \hat{\mathbf{a}}$. At this point, we can test l_{ai} to see if the particle lies within the extent of the line segment. If $0 \leq l_{ai} \leq L$ then particle i lies within the line segment's extent, otherwise it does not. If, instead of a line segment, we want to use an infinite line attractor, all that needs to be done is to ignore this test, and its ramifications below. If the particle lies within the line segment's extent then the vector orthogonal to the line segment out to point $\mathbf{x}_i$ is $\mathbf{r}_{ai} = \mathbf{x}_{ai} - l_{ai}\hat{\mathbf{a}}$, and the distance from the particle to the line is $r_{ai} = \|\mathbf{r}_{ai}\|$. If the particle lies outside of the line segment's extent, we have two choices: we can ignore the line segment's gravitational effect entirely, or we can compute the distance of the particle from the nearest end point of the line segment, yielding

$$\mathbf{r}_{ai} = \begin{cases} \mathbf{x}_i - (\mathbf{x}_a + L\hat{\mathbf{a}}) & \text{if } l_{ai} > L, \\ \mathbf{r}_{ai} & \text{if } 0 \le l_{ai} \le L, \\ \mathbf{x}_{ai} & \text{if } l_{ai} < 0, \end{cases}$$
$$r_{ai} = \|\mathbf{r}_{ai}\|,$$
$$\hat{\mathbf{r}}_{ai} = \mathbf{r}_{ai}/r_{ai}.$$

Now, we can define the additive acceleration operator like we did between two points

$$\mathbf{a}_{ai}^{+\mathrm{op}} = -G_a \frac{1}{r_{ai}^p} \hat{\mathbf{r}}_{ai}.$$

Although the final equation looks quite similar to that of the point attractor, the effect will be quite different. As long as the particle is within the line segment's extent, there will be no acceleration parallel to the line, so any motion of the particle in the parallel direction will be unaffected by the line. If the particle would normally go into an elliptical orbit about a point, it will go into an elliptical spiral around the line segment. If we ignore the effect of the line attractor for particles beyond the line's extent, the particle will spiral around the line and then get thrown out when it moves beyond the end of the attractor.

5.1.2 Random Accelerations

A way to introduce interesting motion to particles is to apply additive accelerations that act randomly. This can be used to simulate any turbulent process, like snowflakes falling subject to slight breezes, or atoms undergoing Brownian motion.

To achieve this type of motion, the random vector generator $\mathbf{S}()$ from Section 4.2 can be scaled by a scale factor S_i and used to produce a small perturbation in particle i's velocity at each timestep. This has the problems that the magnitude of the perturbation will be scaled by the timestep during integration, and the rate at which perturbations occur will be governed by the timestep of the simulation. To produce random perturbations that occur independently from the time step, S_i can be scaled by the timestep S_i/h

$$\mathbf{a}_{si}^{+\mathrm{op}} = \frac{S_i}{h}\mathbf{S}(),$$

and random perturbations can be issued on a schedule independent of timestep. For example, we might want one random acceleration per frame of the animation, so if the animation proceeds at 30 frames per second, we would only create a random acceleration when the time passes an integer multiple of 1/30 of a second.

5.1.3 Drag and Undrag

Additive acceleration operators do not have to be position based. In the bouncing ball problem of Chapter 3 we looked at the effect of air resistance and wind on the ball. The same sort of effect can be used as an acceleration operator for particles. We will call these operators "drag" operators. Letting $\mathbf{v}_i$ be the velocity of particle i, m_i its mass, and $\mathbf{w}$ a constant wind velocity vector, the simplest drag operator would be

$$\mathbf{a}_{wi}^{+\text{op}} = \frac{D}{m_i}(\mathbf{w} - \mathbf{v}_i),$$

where D is a drag "strength" constant that is adjustable by the animator to produce the desired effect.

By changing the sign of D, we create "undrag"–that is, a force tending to increase the speed of the particle. As long as "undrag" is applied, the particle will increase in speed, so this acceleration operator is typically only applied for a short period of time, or until the particle achieves a desired velocity. Used in this way, it is a good way to smoothly increase a particle's speed.

5.1.4 Velocity Limiters

Often we want to assure that a particle maintains at least a minimum speed, or that it is limited to some maximum speed. Sims presents a method to maintain a particle's speed above some minimum speed V_τ. This operator is applied after numerical integration, so it is not an additive acceleration, but operates on the new and previous velocities to produce an updated new velocity:

$$\mathbf{A}^{\text{op}}(\mathbf{v}_i^{[n+1]}, \mathbf{v}_i^{[n]}) = \mathbf{max}[\mathbf{v}_i^{[n+1]}, \mathbf{min}(\mathbf{v}_i^{[n]}, V_\tau \hat{\mathbf{v}}_i^{[n]})].$$

In this operator, the **max** and **min** functions select the vector with the largest or smallest magnitude from their two-vector arguments. As long as a particle's new speed $\|\mathbf{v}_i^{[n+1]}\|$ is above V_τ the new velocity will be selected unchanged, since the second parameter's magnitude can never exceed V_τ. Likewise, if the new speed is above the previous speed $\|\mathbf{v}_i^{[n]}\|$, the new velocity will be selected. Otherwise, the speed will be set to V_τ with direction the same as the previous velocity if the previous speed is greater than V_τ, or will be set to the previous velocity if it is less than V_τ. A similar approach can be used to assure that a particle never exceeds a given maximum speed.

5.2 VELOCITY OPERATORS

In contrast to the acceleration operators, velocity operators work outside of physical law, providing velocity updates for a single timestep, without changing the momentum of a particle. The effect is to reposition the particle, as if there were a velocity, but the velocity itself is not modified. In this way, both acceleration operators and velocity operators can act on the same particle without any interference. This is implemented by modifying the velocity that is used in the numerical integration to compute the new position. Our modified Euler integrator computes the new position by the update

$$\mathbf{x}_i^{[n+1]} = \mathbf{x}_i^{[n]} + \frac{\mathbf{v}_i^{[n+1]} + \mathbf{v}_i^{[n]}}{2} h.$$

We think of a velocity operator $\mathbf{V}^{\mathrm{op}}$ as a function applied to a velocity to return a new velocity. Applying a velocity operator during the integration process, the position update is modified to be

$$\mathbf{x}_i^{[n+1]} = \mathbf{x}_i^{[n]} + \mathbf{V}^{\mathrm{op}}\left(\frac{\mathbf{v}_i^{[n+1]} + \mathbf{v}_i^{[n]}}{2}\right) h.$$

If multiple velocity operators are applied in one timestep we understand $\mathbf{V}^{\mathrm{op}}$ to be the repetitive application of all of the velocity operators taken in order.

5.2.1 Affine Velocity Operators

All of the affine transformations can be used as velocity operators, with the most useful being translation by offset $\triangle\mathbf{v}$:

$$\mathbf{V}^{\mathrm{op}}(\mathbf{v}) = \mathbf{v} + \triangle\mathbf{v},$$

rotation by angle θ about axis $\hat{\mathbf{u}}$*:

$$\mathbf{V}^{\mathrm{op}}(\mathbf{v}) = R(\mathbf{v} : \theta, \hat{\mathbf{u}}),$$

and scale by scale factor s:

$$\mathbf{V}^{\mathrm{op}}(\mathbf{v}) = s\mathbf{v}.$$

All of these operators can be given interesting variation by applying a small amount of randomness to the operator's parameter at each step. Figure 5.3 shows a sequence with a helicopter formed from particles, and then destroyed by a projectile. The formation motion is guided by velocity operators attracting the particles to an underlying, invisible geometric model.

In using velocity operators it is important to remember that their effect is scaled by the step size h to update position at a each timestep. For example, the translation operator will cause the position of the particle to be moved by $\triangle\mathbf{v}h$ at each timestep. If a total translation of $\triangle\mathbf{v}$ is desired, then the velocity operator will have to stay in effect for $1/h$ timesteps (i.e., if $h = 1/30$ then the operator will have to be applied for 30 timesteps). This will result in a smooth translation over the required time. If a sudden jump is desired, then instead of the translation $\triangle\mathbf{v}$, the translation $\triangle\mathbf{v}/h$ may be used for a single timestep.

*This can be implemented using Rodrigues' formula, see Appendix D.6.

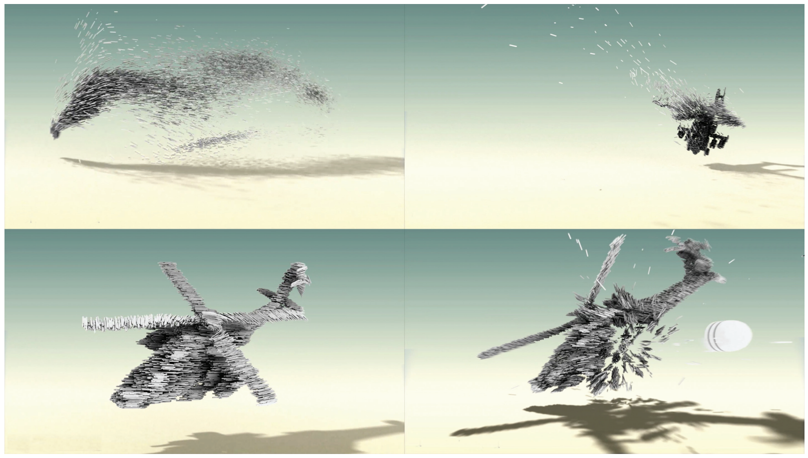


FIGURE 5.3 Particles choreographed to form a helicopter, which is later destroyed. (Courtesy by Hongyuan Johnny Jia.)

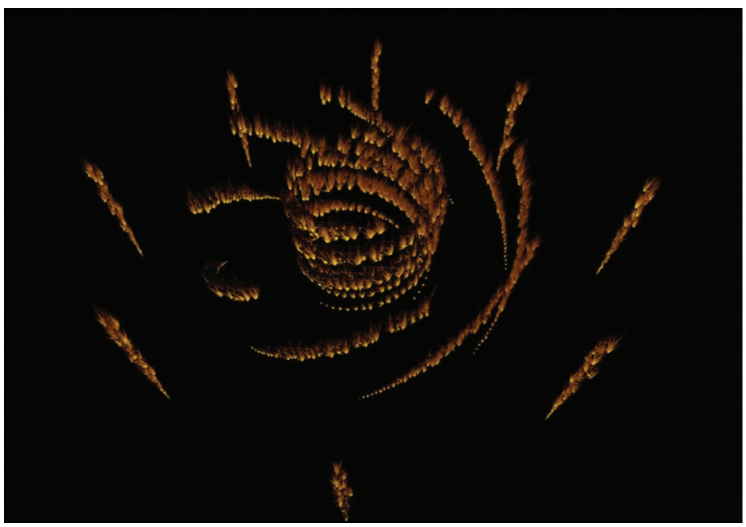


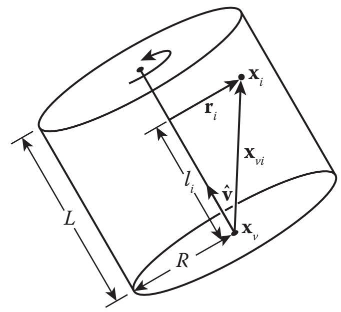
FIGURE 5.4 Spiraling particles choreographed using a vortex. (Courtesy by Ashwin Bangalore.)

5.2.2 Vortices

One of the most interesting velocity operators is the vortex operator. A physical vortex, like a tornado or the water draining in a sink, is a fluid phenomenon characterized by the fluid rotating around an axis, with angular velocity that is very high, at the physical limit, near the rotational axis but decreases with distance from the axis. The decrease with radial distance r from the axis is usually characterized by the scale factor $1/r^2$. Figure 5.4 shows a vortex in action, with particles rendered in a flame-like style.

We can create a vortex effect in our particle system using a velocity operator. If we attempted to create a vortex effect using an acceleration operator, the momentum caused by the high angular velocity near the axis would cause the particles to be flung out of the vortex, rather than being captured by it. By using a velocity operator, we can avoid this problem.

A vortex operator is specified by a cylindrical region in space with base center location $\mathbf{x}_v$, axis oriented along the direction $\hat{\mathbf{v}}$, length L, and radius R, as indicated in the figure to the right. We can think of this as the vortex volume. A particle i, with position $\mathbf{x}_i$ located inside this volume is affected by the vortex, but if it is outside it is unaffected. The vector from the vortex base center to particle i is



$$\mathbf{x}_{vi} = \mathbf{x}_i - \mathbf{x}_v.$$

The length of the projection of this vector onto the vortex' direction is

$$l_i = \hat{\mathbf{v}} \cdot \mathbf{x}_{vi}.$$

If $0 \le l_i \le L$ then particle i lies within the vortex' axial extent, otherwise it does not. The orthogonal vector from the axis to particle i is

$$\mathbf{r}_i = \mathbf{x}_{vi} - l_i \hat{\mathbf{v}},$$

so the distance from the particle to the axis is

$$r_i = \|\mathbf{r}_i\|.$$

If $r_i > R$ then the particle lies outside of the vortex' radial extent, otherwise it is inside the vortex. Now, let f_R be the rotational frequency of the vortex at radius R, measured in cycles per second (Hz). The rotational frequency at the particle's distance r_i is then

$$f_i = \left(\frac{R}{r_i}\right)^{\tau} f_R.$$

The exponent τ can be thought of as a "tightness" parameter that governs how quickly the vortex decays with distance from its axis. For a physical vortex $\tau = 2$. To avoid numerical problems very near the axis, we can apply a frequency limit $f_{\max}$ giving $f_i = \min(f_{\max}, (\frac{R}{r_i})^{\tau} f_R)$. Now, we need to rotate the particle's velocity vector through an angle that will achieve the rotational frequency f_i. If we set the particle's angular velocity to

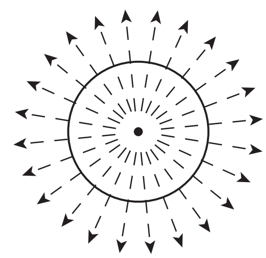
$$\omega = 2\pi f_i,$$

then the actual rotation achieved in one timestep will be ωh, or exactly $2\pi f_i$ radians in 1 second of simulation time.

5.3 COLLISION AVOIDANCE

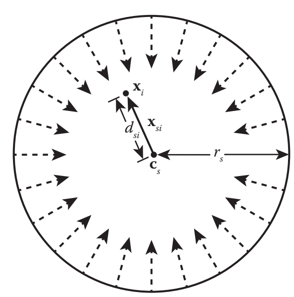
In Chapter 3 we covered the handling of collisions between a ball and a geometric object, but often the effect that we desire is to avoid collisions, directing particles around environmental objects so that they never collide. Imagine a stream of air blowing across a pole. Due to the pressure of the air, the flow will be directed around the pole, with very few air molecules actually colliding with the pole. Collision avoidance in a particle system is a way to have a stream of particles mimic this sort of behavior. We will see this topic again in Chapter 6, when we look at the simulation of the behavior of a flock of birds or school of fish.

5.3.1 Potential Fields



The simplest way to implement collision avoidance is to build a potential field around each environmental obstacle that provides an acceleration away from the obstacle. The point or line gravity acceleration operators described in Section 5.1.1 will work very well for this purpose. By removing the minus sign from these operators, they will act to repel objects rather than attracting them, making them anti-gravity operators. If the object to be avoided is regular in shape, like a sphere or a cube, placing a single anti-gravity point at its center may be sufficient. If the object is elongated, like a pole or skyscraper, then an anti-gravity line can be used. For complex shapes, a collection of points and lines might be needed.

A useful idea here is to provide an extent for each of the operators, so that it exerts no acceleration beyond a maximum distance. In this way, we avoid the problem of having to compute and add the effect of multiple operators at every timestep. Only operators near to a particle will affect it.



Sometimes it is desired to constrain a particle system to remain inside a geometric object. In this case, we can make the entire surface of the containing object be the field generator, so that particles are always pushed away from the surface of the object. A simple example would be a sphere s with center $\mathbf{c}_s$, and radius r_s acting as a container. The acceleration operator would be computed as follows:

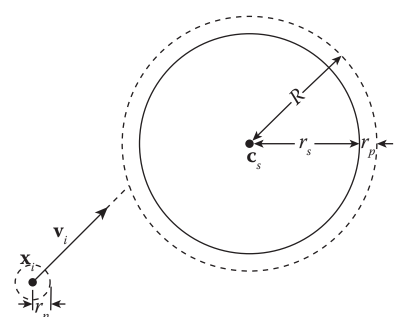
$$\begin{aligned}\mathbf{x}_{si} &= \mathbf{x}_i - \mathbf{c}_s,\\ d_{si} &= \|\mathbf{x}_{si}\|,\\ \mathbf{a}_{si}^{+\text{op}} &= -G_s \frac{1}{(r_s - d_{si})^{p_s}} \hat{\mathbf{x}}_{si}.\end{aligned}$$

Here, G_s is a tunable strength constant, and p_s governs how quickly the force of the anti-gravity falls off with distance from the sphere surface.

Although potential fields are easy to implement and can be effective in preventing collisions, they often produce effects that are not physically realistic. A particle headed directly toward an anti-gravity point will be slowed down and pushed backward but will not turn away. Another problem is that a particle near an anti-gravity point will be pushed away by it whether or not there is any danger of a collision. The net effect will be that particles are prevented by the net force field from colliding with protected objects but in a way that can look physically nonrealistic. However, it can look remarkably like "Shields up Captain!"

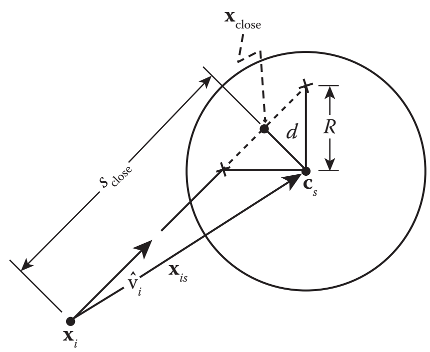
5.3.2 Steering

A more complex but more interesting approach to collision avoidance is particle *steering*. Here we think of the particle as an intelligent actor, imagining that it has eyes looking ahead, and the ability to steer itself by planning a route around impending obstacles. The approach is to locate a viewing coordinate frame at the position of the particle, with its z axis aligned with the velocity vector. We associate with this an angular field of view, which can be tuned by the animator to achieve the desired behavior. Finally, we set a maximum viewing distance, which should be determined by the particle's speed and the estimated time required to take evasive action. The faster the particle is traveling, the farther this distance should be. Now, we imagine that each object within the particle's view cone can be seen by the particle, and its image used by the particle to plan any required evasive action. If the particle's current velocity will bring it into collision with an object, the particle estimates the time to collision and checks the angular distance to the nearest silhouette edge of the obstacle. It then computes an acceleration that will cause it to turn enough to avoid the obstacle within the allotted time. An example will serve to illustrate the process.



The simplest nontrivial example is that of a single fixed sphere obstacle. In practice, this example is actually very useful, as it is often expedient to represent geometry, for collision avoidance purposes, by a simple convex bounding object. For the following analysis, let the sphere center be located at the point $\mathbf{c}_s$ in world coordinates, and its radius be r_s. Let the position of particle i be $\mathbf{x}_i$, and its velocity be $\mathbf{v}_i$. Since we want the particle to pass by the sphere without contacting it, we also define the distance r_p to be a "safe" distance from the collision sphere, and imagine that the particle is protected within a sphere of this radius. The resulting path is identical to steering the particle so it grazes a sphere of radius $R = r_s + r_p$.

We only have to worry about the sphere if the particle is on a collision path with it, and if the collision is scheduled to occur soon. To begin, we compute the particle's closest predicted approach to the sphere's center, assuming that the particle continues to travel at its current velocity. Normalizing the velocity, we create the direction vector $\hat{\mathbf{v}}_i = \mathbf{v}_i/\|\mathbf{v}_i\|$, and we let the vector defined by the sphere center and the particle's position be



$$\mathbf{x}_{is} = \mathbf{c}_s - \mathbf{x}_i.$$

We can treat the particle position together with its velocity direction vector as a ray extending out from the particle along its current direction of travel. The distance along this ray, at the ray's closest approach to the sphere center, is given by

$$s_{\text{close}} = \mathbf{x}_{is} \cdot \hat{\mathbf{v}}_i.$$

If $s_{\text{close}} < 0$ then the sphere is behind the particle and we can ignore it. Also, if the sphere is far in front of the particle we can ignore it until we get closer, so we compute a distance of concern

$$d_c = \|\mathbf{v}_i\| t_c,$$

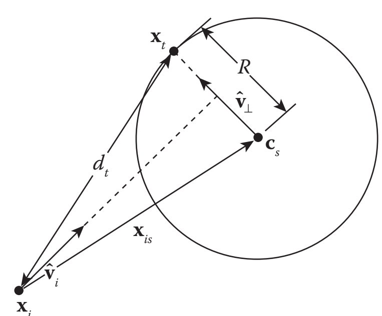
where t_c is a threshold time to collision that can be set by the animator. If $s_{\text{close}} > d_c$, we can ignore the sphere. If we are still concerned with the sphere, we compute the point of closest approach

$$\mathbf{x}_{\text{close}} = \mathbf{x}_i + s_{\text{close}} \hat{\mathbf{v}}_i.$$

The distance of this point to the sphere center is

$$d = \|\mathbf{x}_{\text{close}} - \mathbf{c}_s\|.$$

If $d > R$ then the closest the ray comes to the sphere center is greater than the expanded sphere radius, so there is no predicted collision and we can ignore the sphere. Otherwise, we are scheduled for an imminent collision with the sphere, and we need to take corrective action.



The corrective action will be to apply an acceleration that will adjust the particle's path so that it curves around the sphere. In order to provide the required steering, we first find a spatial target to direct the particle toward. The vector from the sphere center to the closest point of approach is orthogonal to the velocity vector and lies in the plane formed by the velocity vector and $\mathbf{x}_{is}$. This is denoted

$$\mathbf{v}_\perp = \mathbf{x}_{\text{close}} - \mathbf{c}_s.$$

So, the direction from the sphere's center orthogonal to the velocity is $\hat{\mathbf{v}}_\perp = \mathbf{v}_\perp / \|\mathbf{v}_\perp\|$. We determine the turning target

$$\mathbf{x}_t = \mathbf{c}_s + R\hat{\mathbf{v}}_\perp$$

by traveling out along this vector from the sphere's center to the boundary of the extended sphere. The distance from the particle to this point is

$$d_t = \|\mathbf{x}_t - \mathbf{x}_i\|,$$

the speed at which this point is being approached is

$$v_t = \mathbf{v}_i \cdot (\mathbf{x}_t - \mathbf{x}_i)/d_t,$$

and the time to reach this point will be

$$t_t = d_t/v_t.$$

The increased average speed in the direction orthogonal to the current velocity that will be needed to reach this point in the allotted time is

$$\Delta v_s = \|\hat{\mathbf{v}}_i \times (\mathbf{x}_t - \mathbf{x}_i)\|/t_t.$$

To maintain an average speed increase in that direction of v_s, during the elapsed time, the required magnitude of the acceleration is

$$a_s = 2\Delta v_s/t_t.$$

Finally, the required acceleration is

$$\mathbf{a}^{+\text{op}} = a_s \hat{\mathbf{v}}_\perp.$$

One note of caution about this approach to steering around a sphere is that it is still possible for a collision to occur. If other sources of acceleration act to cancel the steering acceleration, its effect can be negated. The solution to this is that the steering operator should override any acceleration opposite to it. If we let $\mathbf{a}_{\text{total}}$ be the sum of all of the other acceleration operators, we can compute its component

$$e = \hat{\mathbf{v}}_\perp \cdot \mathbf{a}_{\text{total}}$$

in the direction of the steering acceleration. If e is negative, then it is opposing the steering acceleration, so we need to strengthen the steering to cancel its effect. If e is positive, then there is already some turning in the direction of steering and we can reduce the steering acceleration by this amount. In either case, this leaves us with the corrected acceleration

$$\mathbf{a}^{+\mathrm{op}} = \max(a_s - e, 0)\hat{\mathbf{v}}_{\perp}.$$

The other way in which steering can fail to prevent collision can be seen by examining the figure above. The vector between $\mathbf{x}_t$ and $\mathbf{x}_i$ actually passes through the expanded sphere, so if the correction $\mathbf{a}^{+\mathrm{op}}$ is applied just once, a collision may occur (depending on the safety factor r_p). In a simulation, this will normally not be a problem, since a corrective acceleration will be recomputed at each timestep, iteratively refining the solution. However, if the speed of the particle is so large that it can reach the sphere in just one or two timesteps, a collision may occur. A more perfect solution is to compute a tangent to the sphere that passes through the particle's position, lies in the plane defined by the particle's velocity and the vector to the sphere's center, and is nearest to the closest point of approach $\mathbf{x}_{\mathrm{close}}$. We leave the details as an exercise for the student.

5.4 SUMMARY

Here is a brief review of some of the main concepts to take away from this chapter:

- While a lot can be done with a particle system simply by making particle generator choices, gaining control over interesting motion requires the ability to choreograph particles to achieve a desired look or effect.
- One way to organize thinking about particle choreography is to think in terms of operators, operating on the position and velocity of the particles.
- Acceleration operators affect a particle's velocity by applying accelerations. These are equivalent to applying forces. These can be used to produce effects such as gravitational attraction, random perturbations, and drag.
- Velocity operators affect a particle's position by applying velocity increments. Unlike forces, these do not result in a change in particle momentum, but simply affect a particle's position. These can be used to produce effects such as affine transformations of the particles and vortices.
- Potential fields can be used to prevent particle collisions with an object by applying forces pushing away from the object.
- A more sophisticated approach to collision avoidance involves computing steering forces that tend to steer particles around an object.