

# Momentum, Angular Momentum, and Multi-Particle Systems

In this chapter we focus on two important physical quantities linear momentum and angular momentum. Under certain conditions, as explained in this chapter, momentum and angular momentum are conserved quantities, meaning that their values are constant during a physical process. The conservation of momentum or angular momentum can then be used to solve problems by comparing the state of a system before a physical process to the state of that same system after this process. In this chapter, we will also study multi-particle systems, continuous mass distributions (i.e. systems that are not point particles), and center of mass. We develop relationships between the total momentum of a system of particles to the momentum of the system’s center of mass. We also develop a relationship between the angular momentum of a system of particles to angular momentum of the system’s center of mass. We will show that these relationships will also hold for continuous mass distributions. Finally, we discuss numerical integration techniques which will be useful when finding a system’s center of mass.

## 4.1 CONSERVATION OF MOMENTUM AND NEWTON’S THIRD LAW

Recall that Newton’s Third Law states that if Particle 2 exerts a force  $\mathbf{F}_{21}$  on Particle 1, then Particle 1 exerts a force  $\mathbf{F}_{12}$  on Particle 2 such that:

$$\mathbf{F}_{12} = -\mathbf{F}_{21} \quad (4.1.1)$$

Newton’s Third Law will lead us to an interesting result. Consider two particles interacting via a force, which we will refer to as an “internal force”. For example, if our two objects are the Earth and Moon, their gravitational interaction would be the internal force. Furthermore, assume that each particle is experiencing an external force due to other bodies outside of the system. In the case of the Earth and the Moon, that could be the Sun’s gravitational force, a body outside the Earth-Moon system. The net force  $\mathbf{F}_1$  on particle 1 is then:

$$\mathbf{F}_1 = \mathbf{F}_{21} + \mathbf{F}_1^{\text{ext}} \quad (4.1.2)$$

108 ■ Classical Mechanics: A Computational Approach

where  $\mathbf{F}_{21}$  is the force on particle 1 exerted by particle 2 (the “internal force”), and  $\mathbf{F}_1^{\text{ext}}$  is the net external force acting on particle 1. Similarly, particle 2 experiences a net force:

$$\mathbf{F}_2 = \mathbf{F}_{12} + \mathbf{F}_2^{\text{ext}} \quad (4.1.3)$$

Let  $\mathbf{p}_1$  and  $\mathbf{p}_2$  represent the linear momenta of the two particles, then Newton’s Second Law states that

$$\dot{\mathbf{p}}_1 = \mathbf{F}_1 = \mathbf{F}_{21} + \mathbf{F}_1^{\text{ext}} \quad (4.1.4)$$

and

$$\dot{\mathbf{p}}_2 = \mathbf{F}_2 = \mathbf{F}_{12} + \mathbf{F}_2^{\text{ext}} \quad (4.1.5)$$

Now, we define the total momentum vector as  $\mathbf{P} = \mathbf{p}_1 + \mathbf{p}_2$ , to be the sum of the momenta of each particle. Then,

$$\dot{\mathbf{P}} = \dot{\mathbf{p}}_1 + \dot{\mathbf{p}}_2 = \mathbf{F}_1^{\text{ext}} + \mathbf{F}_2^{\text{ext}} = \mathbf{F}^{\text{ext}} \quad (4.1.6)$$

Hence, we see that if there are no total net external forces  $\mathbf{F}^{\text{ext}} = \mathbf{F}_1^{\text{ext}} + \mathbf{F}_2^{\text{ext}}$  then  $\dot{\mathbf{P}} = 0$ , or to put it another way:

***The Law of Conservation of Linear Momentum***

If  $\mathbf{F}^{\text{ext}} = 0$ , then  $\mathbf{P} = \text{constant}$ .

The total momentum of a system is conserved if no external forces act on that system. This is an important law in physics, one that we will use to solve problems in cases where the external forces add to zero.

***Example 4.1: Inelastic collision between two bodies***

Consider two objects with masses  $m_1$  and  $m_2$  and moving with velocities  $\mathbf{v}_1$  and  $\mathbf{v}_2$  respectively. The two masses collide and stick together, moving away from the collision as one object with a velocity  $\mathbf{v}$ . Ignoring external forces occurring during the collision find the velocity of the objects immediately after the collision.

***Solution:***

This type of collision is called a *perfectly inelastic collision* because the two bodies are stuck together after the collision. Because we can ignore the external forces during the collision, we can use conservation of momentum to solve this problem. The total momentum before the collision is:

$$\mathbf{P}_{\text{initial}} = m_1 \mathbf{v}_1 + m_2 \mathbf{v}_2$$

and the total momentum after the collision is:

$$\mathbf{P}_{\text{final}} = (m_1 + m_2) \mathbf{v}$$

Note that because the two objects stick together after the collision, both masses have the same velocity, hence the right-hand side of  $\mathbf{P}_{\text{final}}$  above. Conservation of momentum tells us that  $\mathbf{P}_{\text{initial}} = \mathbf{P}_{\text{final}}$  and therefore:

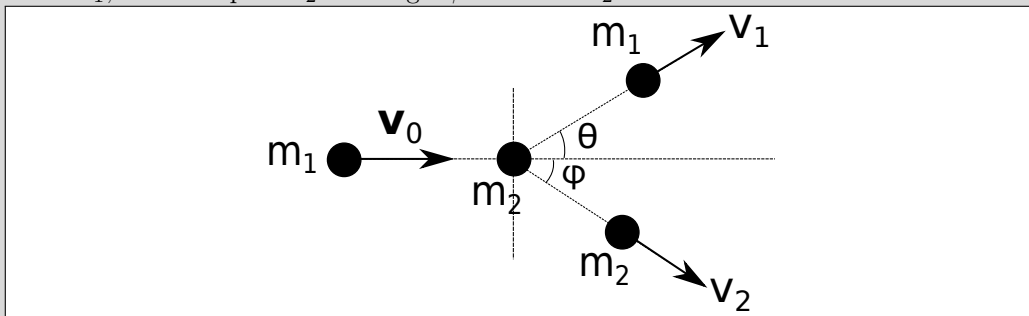
$$\mathbf{v} = \frac{m_1 \mathbf{v}_1 + m_2 \mathbf{v}_2}{m_1 + m_2}$$

Notice that the final velocity is the weighted average of the initial velocities, where the masses of each object serve as the weights for the average. While this problem may seem to be simple, it is an important application of the conservation of momentum; it is used

to solve problems in nuclear physics, astrophysics, and accident reconstruction—to name a few.

**Example 4.2: Elastic collision between two bodies**

A particle of mass  $m_1 = 0.10$  kg moving with a velocity of  $\mathbf{v}_0 = 10.0$  m/s  $\hat{\mathbf{i}}$  collides with another particle which is at rest and has a mass  $m_2 = 0.20$  kg. After the collision, each particle moves as shown in the figure below, at an angle  $\theta = \pi/6$  rad. The collision is *elastic*, meaning that the kinetic energy of each particle is conserved. Find the speed  $v_1$  of mass  $m_1$ , and the speed  $v_2$  and angle  $\phi$  for mass  $m_2$  after the collision.



**Solution:**

You may remember from your introductory physics course that in an elastic collision both the total momentum  $m\mathbf{v}$  and total kinetic energy  $\frac{1}{2}mv^2$  are conserved. We will discuss kinetic energy more in depth in the next chapter. For now, your knowledge from your introductory physics course will suffice to solve this problem.

Only mass  $m_1$  is moving before the collision, and the problem states that the velocity is in the  $x$ -direction. Therefore,

$$\mathbf{p}_{\text{initial}} = m_1 v_0 \hat{\mathbf{i}}$$

The momentum after the collision involves both particles. By analyzing the velocity vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  into their  $x$  and  $y$  components. We can write:

$$\mathbf{p}_{\text{final}} = (m_1 v_1 \cos \theta + m_2 v_2 \cos \phi) \hat{\mathbf{i}} + (m_1 v_1 \sin \theta - m_2 v_2 \sin \phi) \hat{\mathbf{j}}$$

Using conservation of momentum, we equate  $\mathbf{p}_{\text{initial}} = \mathbf{p}_{\text{final}}$ . In addition, because the collision is elastic, we also equate the initial and final kinetic energies. Together, the two conservation laws lead to three equations:

$$\begin{aligned} m_1 v_0 &= m_1 v_1 \cos \theta + m_2 v_2 \cos \phi \\ 0 &= m_1 v_1 \sin \theta - m_2 v_2 \sin \phi \\ \frac{1}{2} m_1 v_0^2 &= \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \end{aligned}$$

The top two equations represent the total momentum in the  $x$  and  $y$  directions respectively, and the final equation represents the conservation of kinetic energy. In addition, we know that  $v_0 = 10.0$  m/s,  $m_1 = 0.10$  kg, and  $m_2 = 0.20$  kg. As you might imagine, this problem can involve a lot of algebra. However, a CAS will be able to assist us with that algebra. Below is an example of code written in Mathematica, which can be used to solve this problem.

110 ■ Classical Mechanics: A Computational Approach

```

θ = π/6;
v0 = 10;
m1 = 0.1;
m2 = 0.2;

Solve[{m1*v0 == m1*v1*Cos[θ] + m2*v2*Cos[φ], m1*v1*Sin[θ] ==
m2*v2*Sin[φ], 0.5*m1*v0^2 == 0.5*m1*v1^2 + 0.5*m2*v2^2}, {v1, v2, φ}]

OUTPUT: Solve::ifun: Inverse functions are being used by Solve, so some solutions
may not be found; use Reduce for complete solution information.

{{v1 → -3.56822, v2 → -6.6056, φ → 3.00613}, {v1 → -3.56822, v2 → 6.6056, φ →
-0.135459}, {v1 → 9.34172, v2 → -2.52311, φ → -1.95894}, {v1 → 9.34172, v2 →
2.52311, φ → 1.18266}}
    
```

We begin the program by defining our known variables, the final direction  $\theta$  of mass  $m_1$ , the initial velocity of  $m_1(v_0)$ , and the values of the masses  $m_1$  and  $m_2$ . The *Solve* command gives a warning that some solutions may not be found by Mathematica. This is not uncommon when using a CAS to solve transcendental equations. However, four solutions are found. The first three solutions include negative values for  $v_1$  and  $v_2$ . These are non-physical results because  $v_1$  and  $v_2$  are speeds, and speed is the positive magnitude of the velocity vector. The fourth solution found by Mathematica contains all positive values, which is in agreement with the problem’s diagram. Finally, we need to remember that Mathematica, like any calculator, is not respecting significant figures. Therefore, our final answer is:  $v_1 = 9.3$  m/s,  $v_2 = 2.5$  m/s, and  $\phi = 1.2$  rad. It should be noted, that although the warning suggests using the command *Reduce* in this case, the set of solutions produced by *Reduce* includes the ones found by *Solve*.

As we have said before, the availability of CAS programs is not a substitute for knowing how to do algebra. The reader should verify the solution by hand.

Next, we study the case where three or more particles are involved. The question arises: Do we still expect momentum to be conserved when there are many interacting particles? Let’s begin by examining the net force acting on a collection of  $N$  interacting particles. The net force on the  $i^{\text{th}}$  particle is:

$$\mathbf{F}_i = \mathbf{F}_i^{\text{ext}} + \sum_{j \neq i} \mathbf{F}_{ji} \tag{4.1.7}$$

where as previously stated  $\mathbf{F}_i^{\text{ext}}$  is the net external force acting on the  $i^{\text{th}}$  particle, and  $\mathbf{F}_{ji}$  is the force of interaction between particle  $j$  and particle  $i$ . The summation term in the equation above denotes the net internal force acting on the particle. The net force acting on the system would then be obtained by summing the equation above over the  $N$  particles:

$$\mathbf{F} = \sum_{i=1}^N \mathbf{F}_i = \sum_{i=1}^N \mathbf{F}_i^{\text{ext}} + \sum_{i=1}^N \sum_{j \neq i} \mathbf{F}_{ji} \tag{4.1.8}$$

Equation (4.1.8) can be re-written as:

$$\mathbf{F} = \sum_{i=1}^N \mathbf{F}_i^{\text{ext}} + \sum_{i=1}^N \sum_{j > i} (\mathbf{F}_{ij} + \mathbf{F}_{ji}) \tag{4.1.9}$$

If this rewriting is not clear to you, set  $N = 3$  and write out the terms in (4.1.8), you will find that you can rewrite the sum as a collection of two terms, as shown above. Newton’s Third Law tells us that  $\mathbf{F}_{ji} = -\mathbf{F}_{ij}$ , and therefore the second term in the previous equation is zero leading to:

$$\dot{\mathbf{P}} = \mathbf{F} = \sum_{i=1}^N \mathbf{F}_i^{\text{ext}} \quad (4.1.10)$$

***The Law of Conservation of Linear Momentum for a System of  $N$  Particles***

$$\text{If } \sum_{i=1}^N \mathbf{F}_i^{\text{ext}} = 0 \quad \text{then } \mathbf{P} = \text{constant} \quad (4.1.11)$$

In other words, the momentum of the system of particles is conserved if there are no external forces acting on the system, like in the case of two-particle systems.

## 4.2 ROCKETS

A practical example of conservation of momentum is rocket propulsion. The rocket has a challenge of moving forward without pushing against something. For example, an automobile moves forward by turning its wheels so that it pushes against the ground. However, a rocket in space has nothing to push against. Instead, the rocket solves this problem by ejecting mass, similar to the recoil of a gun. Before a gun is fired, the bullet and the gun are at rest. However, once the bullet is fired, it is propelled forward. In order for the total momentum to be conserved, the gun has to move in the direction opposite that of the bullet. The velocities of the gun and the bullet are, of course, very different because the gun has more mass than the bullet does.

We begin by considering a rocket flying horizontally and assume that the rocket is not experiencing any external forces. The rocket has momentum  $p(t) = mv$  at time  $t$ , where  $m$  and  $v$  are the mass and velocity of the rocket. The momentum of the rocket is measured by an observer in a reference frame that is at rest relative to the rocket. A short time later denoted by  $t + dt$ , the rocket has expelled a small amount of mass  $dm'$ , which is moving at a speed  $u$  relative to the rocket, as illustrated in Figure 4.1. The speed  $u$  is sometimes called the *exhaust speed*. The ejection of mass leads to an increase in the rocket’s speed by a small amount  $dv$ , as measured in the rest frame. Therefore, the momentum of the system immediately after the mass was ejected is:

$$p(t + dt) = (m - dm')(v + dv) + dm'(v - u) \quad (4.2.1)$$

where the second term above is the momentum of the ejected mass relative to the rest frame.

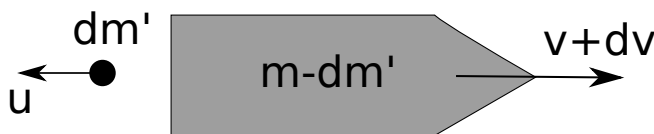


Figure 4.1: A rocket after ejecting a small amount of mass  $dm'$  at a speed relative to the rocket,  $u$ .

**112** ■ Classical Mechanics: A Computational Approach

Applying conservation of momentum:

$$p(t) = p(t + dt) \tag{4.2.2}$$

$$mv = (m - dm')(v + dv) + dm'(v - u) \tag{4.2.3}$$

$$m dv = u dm' \tag{4.2.4}$$

$$m dv = -u dm \tag{4.2.5}$$

where the last line was obtained by noting that the change in mass of the rocket is  $dm = -dm'$ . Furthermore, we ignored the term  $dm' dv$  because both  $dm'$  and  $dv$  are small, therefore their product is negligible. Dividing both sides by  $dt$  we obtain:

$$m \dot{v} = -u \dot{m} \tag{4.2.6}$$

The left hand side of (4.2.6) is Newton’s Second Law ( $F = ma$ ), and therefore we see that the net force acting on the rocket is equal to  $-u \dot{m}$ , which is sometimes called the *thrust*. Note that  $\dot{m} < 0$ , so the thrust is positive (and points to the right in Figure 4.1). We can also write (4.2.6) as:

$$dv = -u \frac{dm}{m} \tag{4.2.7}$$

This equation can be integrated by assuming a constant  $u$ . If at  $t = 0$  the rocket’s speed is  $v_0$  and its mass is  $m_0$ , we obtain:

$$v - v_0 = u \ln \left( \frac{m_0}{m} \right) \tag{4.2.8}$$

which gives the velocity of the rocket as a function of its mass  $m$ . Equation (4.2.8) tells us that in order to make a rocket go as fast as possible, engineers need to create rockets with large exhaust speed  $u$ , and with a large initial to final mass ratio  $m_0/m$ . The final speed of the rocket will be obtained once all of the fuel is burned. Rockets with multiple stages that are ejected can help maximize  $m_0/m$ , so that the rocket can achieve even greater speeds.

Next, we study the case of a rocket moving vertically, experiencing the force of gravity. Our goal is to find an equation which describes the velocity of the rocket. We will use a coordinate system where the positive  $y$ -direction is upward, in the opposite direction of the force of gravity. In this case, gravity is an external force acting on the rocket, and therefore:

$$\frac{dp}{dt} = -mg \tag{4.2.9}$$

$$dp = -mgdt \tag{4.2.10}$$

$$p(t + dt) - p(t) = -mgdt \tag{4.2.11}$$

$$m dv + u dm = -mgdt \tag{4.2.12}$$

$$m \frac{dv}{dt} + u \frac{dm}{dt} = -mg \tag{4.2.13}$$

In order to obtain an equation for  $v$ , we introduce the constant *burn rate*  $\alpha$ :

$$\alpha = -\frac{dm}{dt} \tag{4.2.14}$$

Using our definition of burn rate, the last line of (4.2.13) becomes:

Momentum, Angular Momentum, and Multi-Particle Systems ■ 113

$$\frac{dv}{dt} = -g + \frac{\alpha}{m}u \tag{4.2.15}$$

$$dv = \left(-g + \frac{\alpha u}{m}\right) dt \tag{4.2.16}$$

$$dv = \left(\frac{g}{\alpha} - \frac{u}{m}\right) dm \tag{4.2.17}$$

Integrating the above equation using  $v(0) = v_0$  and  $m(0) = m_0$  produces:

$$v - v_0 = \frac{g}{\alpha}(m - m_0) + u \ln\left(\frac{m_0}{m}\right) \tag{4.2.18}$$

This equation gives the velocity of the rocket as a function of mass  $m$  when the burn rate  $\alpha$  is constant. Sometimes (4.2.18) is rewritten by integrating the definition of the burn rate:  $m - m_0 = -\alpha t$ . In that case, (4.2.18) becomes:

$$v - v_0 = -gt + u \ln\left(\frac{m_0}{m}\right) \tag{4.2.19}$$

The advantage of using (4.2.19) is that one can see two terms which affect the rocket’s motion. The first term,  $-gt$ , is the standard kinematics term for a particle in free fall, and represents the effect of gravity on the velocity. The second term,  $u \ln(m_0/m)$ , is the same term we obtained for the horizontal motion of the rocket and represents the effect of thrust on the rocket’s velocity.

### 4.3 CENTER OF MASS

Problems involving momentum often involve multiple objects, such as in the case of collisions. In situations where the system of interest involves multiple particles or the object under study is not a point particle, the concept of *center of mass* becomes useful. Let us begin by considering a system of  $N$  point particles. The location  $\mathbf{R}$  of the center of mass for a system of particles is defined as:

*Center of Mass for a System of Discrete Particles*

$$\mathbf{R} = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{r}_i \tag{4.3.1}$$

$$X = \frac{1}{M} \sum_{i=1}^N m_i x_i \quad Y = \frac{1}{M} \sum_{i=1}^N m_i y_i \quad Z = \frac{1}{M} \sum_{i=1}^N m_i z_i \tag{4.3.2}$$

Here  $\mathbf{r}_i$  is the position of the  $i^{\text{th}}$  particle which has mass  $m_i$ . The total mass of the system is  $M = \sum m_i$ . Equation (4.3.1) is actually three equations, one for each coordinate of  $\mathbf{R}$ . For example, in Cartesian coordinates the position of the  $i^{\text{th}}$  particle is  $\mathbf{r}_i = x_i \hat{\mathbf{i}} + y_i \hat{\mathbf{j}} + z_i \hat{\mathbf{k}}$  and we can write  $\mathbf{R} = X \hat{\mathbf{i}} + Y \hat{\mathbf{j}} + Z \hat{\mathbf{k}}$ .

The center of mass is the weighted average location of the mass in the system, with the weights being the masses  $m_i$ . It is a weighted average, similar to how grades are often calculated in a course where maybe 50% of your final grade may be based on mid-semester exams, 20% on homework assignments, and 30% on a final examination. The location of the center of mass will be closer to the heavier particles. The next example illustrates this point.

114 ■ Classical Mechanics: A Computational Approach

**Example 4.3: Center of mass of the Earth-Sun system**

Compute the location of the center of mass of the Earth-Sun system.

**Solution:**

We choose a coordinate system such that the center of the Sun is at the origin. We will also approximate the Earth’s orbit as circular, and assume that the plane of the Earth’s orbit is constant. Therefore, we can work in polar coordinates and only need to find the distance of the center of mass of the system from the Sun’s center. The distance between the Earth and the Sun is approximately  $1.5 \times 10^{11}$  m, and the masses of the Earth and Sun are  $5.9 \times 10^{24}$  kg and  $2.0 \times 10^{30}$  kg, respectively. Therefore, in our coordinate system,  $r_{\text{Sun}} = 0$  and  $r_{\text{Earth}} = 1.5 \times 10^{11}$  m. Inserting these numbers into (4.3.1) gives:

$$\frac{(m_{\text{Earth}})(r_{\text{Earth}})}{m_{\text{Sun}} + m_{\text{Earth}}} = \frac{(5.9 \times 10^{24} \text{ kg})(1.5 \times 10^{11} \text{ m})}{5.9 \times 10^{24} \text{ kg} + 2.0 \times 10^{30} \text{ kg}} = 4.6 \times 10^5 \text{ m}.$$

The radius of the Sun is approximately  $7.0 \times 10^8$  m. Therefore, the center of mass of the Earth-Sun system is well inside the Sun, and practically coincides with the center of the Sun.

If we want to extend (4.3.1) to continuous distributions of mass, i.e. objects that are not well-described as point particles, then we need to make some changes. One way to approach a continuous mass distribution is by thinking of it as a collection of infinitesimal masses  $dm$ . Then the sum over discrete particles in (4.3.1) becomes an integral over infinitesimal mass elements  $dm$ . The center of mass equations then become:

**Center of Mass for a Continuous Mass Distribution**

$$\mathbf{R} = \frac{1}{M} \int \mathbf{r} dm \quad M = \int dm \quad (4.3.3)$$

$$X = \frac{1}{M} \int x dm \quad Y = \frac{1}{M} \int y dm \quad Z = \frac{1}{M} \int z dm \quad (4.3.4)$$

In the equations above, the total mass is found from  $M = \int dm$ , and like (4.3.1) equation (4.3.3) actually consists of three equations, one for each coordinate. Typically, one does not integrate (4.3.3) using the mass. Most often the integration is done over the length, area or volume of the object, when the object does not have a uniform density.

The following box summarizes how to calculate the total mass  $M$  by using the density of the object in 1D, 2D, and 3D situations. In the case of one-dimensional objects, we use a linear mass density  $\lambda$  (mass per unit length in units of kg/m). In the case of two-dimensional objects, we use a surface density  $\sigma$  (mass per unit area in units of kg/m<sup>2</sup>), and for three-dimensional objects we use the familiar volume density  $\rho$  (mass per unit volume in units of kg/m<sup>3</sup>).

**Evaluating the Mass  $dm$  for 1D, 2D and 3D Objects**

$$dm = \lambda dx \quad M = \int \lambda dx \quad (4.3.5)$$

$$dm = \sigma dA \quad M = \int \sigma dA \quad (4.3.6)$$

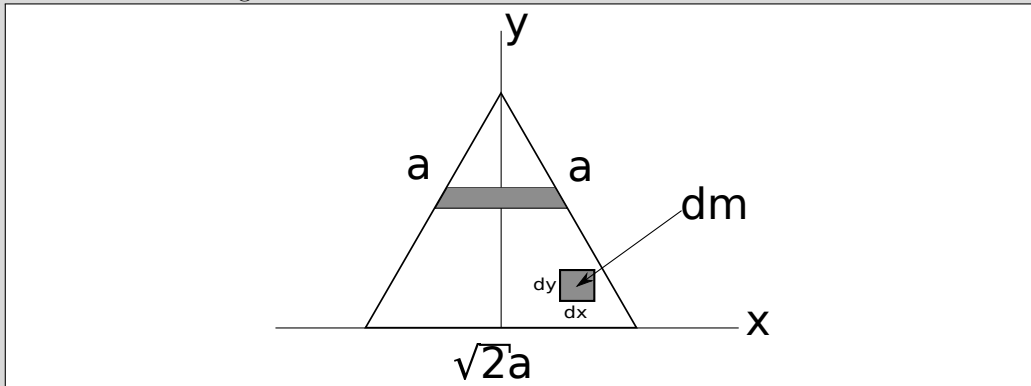


$$dm = \rho dV \quad M = \int \rho dV \quad (4.3.7)$$

The next example will demonstrate how to apply (4.3.3) to a continuous mass distribution.

**Example 4.4: Center of mass of an isosceles triangle**

A lamina of uniform mass per unit area  $\sigma$ , is shaped into an isosceles triangle shown below. The triangle has two sides of length  $a$  and the base has length  $\sqrt{2}a$ . Find the center of mass of this triangular lamina.



**Solution:**

First note that due to the symmetry of the triangle, the center of mass lies along the  $y$ -axis. Computing the center of mass of a mass distribution involves identifying two components. The first component to identify is the *mass element*  $dm$ , shown in the diagram of the triangle. Cartesian coordinates are a natural choice of coordinates for this problem. Therefore,  $dm = \sigma dx dy$  will be used for the mass element.

The second component needed to find the center of mass are *the limits of the integral* in (4.3.3). The limits of the integral range over the size of the object. We included a horizontal gray strip near the top of the triangle to illustrate that the width of the triangle varies with the height, with the strip becoming narrower as it moves up the triangle. The right side of the strip is bound by the line  $y = a/\sqrt{2} - x$  and the left side is bound by the line  $y = a/\sqrt{2} + x$ , because the triangle has a height  $a/\sqrt{2}$ . Therefore, we can consider the left end of the strip to be at  $x = y - a/\sqrt{2}$ , and the right end to be at  $x = a/\sqrt{2} - y$ .

The  $y$ -coordinate of the center of mass can then be found by computing  $Y = 1/M \int y dm$ :

116 ■ Classical Mechanics: A Computational Approach

$$\begin{aligned}
 Y &= \frac{1}{M} \int_{y=0}^{y=a/\sqrt{2}} \int_{x=y-a/\sqrt{2}}^{a/\sqrt{2}-y} y \sigma dx dy \\
 &= \frac{\sigma}{M} \int_0^{a/\sqrt{2}} yx \Big|_{y-a/\sqrt{2}}^{a/\sqrt{2}-y} dy \\
 &= \frac{\sigma}{M} \int_0^{a/\sqrt{2}} 2y \left( \frac{a}{\sqrt{2}} - y \right) dy \\
 &= \frac{\sigma}{M} \left( \frac{ay^2}{\sqrt{2}} - \frac{2y^3}{3} \right) \Big|_0^{a/\sqrt{2}} \\
 &= \frac{a^3}{6\sqrt{2}} \frac{\sigma}{M}
 \end{aligned}$$

To finish the problem, we need to find the total mass  $M$ .

$$M = \int dm = \int_{y=0}^{y=a/\sqrt{2}} \int_{x=y-a/\sqrt{2}}^{a/\sqrt{2}-y} \sigma dx dy = \sigma \int_0^{a/\sqrt{2}} 2 \left( \frac{a}{\sqrt{2}} - y \right) dy = \frac{\sigma a^2}{2}$$

By using the value for  $M$  found above we find that:

$$Y = \frac{a}{3\sqrt{2}}$$

We can use a CAS like Mathematica to evaluate the integrals, even if the limits are not constant. Below is an example of how to use Mathematica to solve the problem. Note that we use the command /. which means replace. The line  $Y /. m \rightarrow M$  asks Mathematica to take the values stored as  $M$  ( $a^2\sigma/2$ ) and insert that where the letter  $m$  appears in the expression stored in  $Y$  ( $a^3\sigma/6m\sqrt{2}$ ).

```
Y = Integrate [  $\frac{\sigma*y}{m}$ , {y, 0,  $\frac{a}{\sqrt{2}}$ }, {x, y -  $\frac{a}{\sqrt{2}}$ ,  $\frac{a}{\sqrt{2}}$  - y} ]
```

```
OUTPUT:  $\frac{a^3\sigma}{6\sqrt{2}m}$ 
```

```
M = Integrate [  $\sigma$ , {y, 0,  $\frac{a}{\sqrt{2}}$ }, {x, y -  $\frac{a}{\sqrt{2}}$ ,  $\frac{a}{\sqrt{2}}$  - y} ]
```

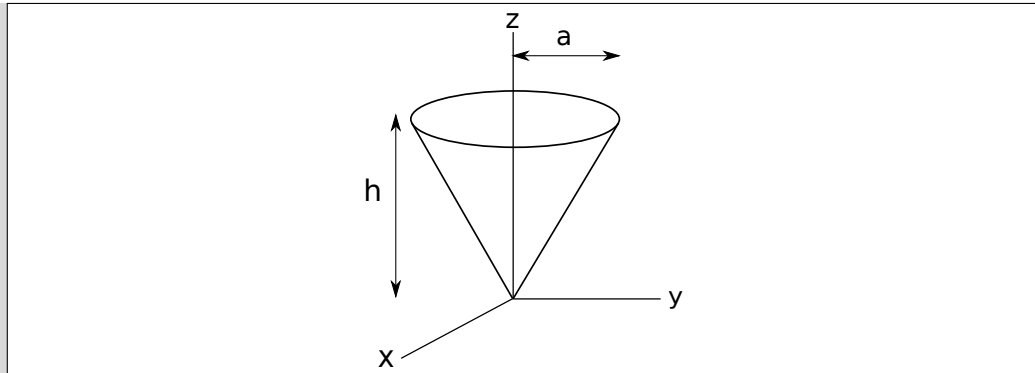
```
OUTPUT:  $\frac{a^2\sigma}{2}$ 
```

```
Y /. m → M
```

```
OUTPUT:  $\frac{a}{3\sqrt{2}}$ 
```

**Example 4.5: The center of mass of a solid cone.**

A cone of uniform density  $\rho$  is shown in the figure below. The cone has a radius of  $a$  at the top and a height  $h$ . Compute the center of mass of the cone.



**Solution:**

To solve this problem, we first need to identify a coordinate system. A cone can be naturally described in cylindrical coordinates. Therefore we use  $dm = \rho dV = \rho r dr d\theta dz$ . By symmetry, the center of mass is along the  $z$ -axis. The  $\theta$  coordinate ranges from 0 to  $2\pi$  and the  $z$  coordinate ranges from 0 to  $h$ . However, the radius of the cone depends on the height of the mass element above the  $xy$ -plane. The equation of the edge of the cone in the  $yz$ -plane in the figure is  $r = az/h$ , so that the radius  $r$  ranges from 0 to  $az/h$ . Putting all of this together,

$$\begin{aligned} Z &= \frac{1}{M} \int_{z=0}^h \int_{\theta=0}^{2\pi} \int_{r=0}^{az/h} z \rho r dr d\theta dz \\ &= \frac{2\pi\rho}{M} \int_0^h z \left. \frac{r^2}{2} \right|_0^{az/h} dz \\ &= \frac{2\pi\rho a^2}{2h^2 M} \int_0^h z^3 dz = \frac{\pi\rho a^2 h^2}{4M} \end{aligned}$$

Solving for the total mass  $M$ :

$$\begin{aligned} M &= \int_0^h \int_0^{2\pi} \int_0^{az/h} \rho r dr d\theta dz \\ &= 2\pi\rho \int_0^h \left. \frac{r^2}{2} \right|_0^{az/h} dz \\ &= \frac{\pi\rho a^2}{h^2} \int_0^h z^2 dz = \frac{\pi\rho a^2 h}{3} \end{aligned}$$

Finally, inserting  $M$  into the result for  $Z$ , we get:

$$Z = \frac{3}{4}h$$

Again, we can perform the triple integration in Mathematica as follows.

118 ■ Classical Mechanics: A Computational Approach

```
Z = Integrate[ $\frac{\rho * z * r}{m}$ , { $\theta$ , 0, 2 $\pi$ }, {z, 0, h}, {r, 0, az/h}]
```

```
OUTPUT:  $\frac{a^2 h^2 \pi \rho}{4m}$ 
```

```
M = Integrate[ $\rho * r$ , { $\theta$ , 0, 2 $\pi$ }, {z, 0, h}, {r, 0, az/h}]
```

```
OUTPUT:  $\frac{1}{3} a^2 h \pi \rho$ 
```

```
Z/.m -> M
```

```
OUTPUT:  $\frac{3h}{4}$ 
```

Notice that in order to perform the triple integral in cylindrical coordinates, we needed to explicitly include the equation for  $r = az/h$  in the argument of the *Integrate* command.

#### 4.4 NUMERICAL INTEGRATION AND THE CENTER OF MASS

We will sometimes need to use numerical integration techniques when finding the center of mass of objects that are irregularly shaped, or have complicated density functions. In this section we will present some basic numerical integration techniques and apply them to problems involving center of mass.

##### 4.4.1 Trapezoidal Rule

The most basic numerical integration algorithm is called the trapezoidal rule. As the name suggests, the trapezoidal rule approximates the area under the curve by using a single trapezoid as shown in Figure 4.2.

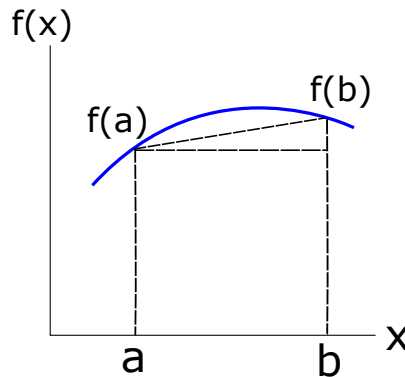


Figure 4.2: The area under the curve  $f(x)$  between  $x = a$  and  $x = b$  can be approximated using the trapezoidal rule, which estimates the area using the dashed trapezoid shown in the figure.

The trapezoidal area under the curve between the points  $x = a$  and  $x = b$  is found by taking the sum of the triangle and rectangle shown in Figure 4.2:

$$\int_a^b f(x)dx = (b-a)f(a) + 1/2(b-a)(f(b) - f(a)) + O((b-a)^3 f'') \quad (4.4.1)$$

$$= \frac{b-a}{2} [f(b) - f(a)] + O((b-a)^3 f'') \quad (4.4.2)$$

where the error term  $O((b-a)^3 f'')$  is not included in calculations. The  $f''$  in the term is the second derivative of  $f(x)$  evaluated at a point between  $x = a$  and  $x = b$  that maximizes the second derivative. This is done to give an upper bound on the error. As expected, the trapezoidal rule is not a preferred algorithm when it comes to calculating integrals, because the area of a more complicated function may not be well-approximated by a single trapezoid. However, the trapezoidal rule is the basis of other algorithms, hence it is important to know.

### 4.4.2 Simpson's Rule

Simpson's rule has two advantages over the trapezoid rule. First, it breaks up the range of the integral into more pieces. Second, it replaces the straight-line top of the trapezoid with a quadratic polynomial. In this subsection, we will forego the derivation of Simpson's rule, and simply present its formula. Simpson's rule approximates the integral

$$\int_{x_0}^{x_2} f(x')dx' \quad (4.4.3)$$

by sampling the function  $f(x)$  at three points  $x_0$ ,  $x_1$ , and  $x_2$ , equally spaced by a distance  $h$ . The formula for Simpson's rule is:

$$\int_{x_0}^{x_2} f(x')dx' = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + O(h^5 f^{(4)}(x')) \quad (4.4.4)$$

where  $h = x_2 - x_1 = x_1 - x_0$  and  $x_1 = (x_0 + x_2)/2$ . The last term in (4.4.4) is the remainder (or error) term and is not included in calculations. The fourth-order derivative is evaluated at some value in the interval  $[x_0, x_2]$ . While Simpson's rule divides the area under  $f(x)$  into more segments than the trapezoidal rule, even more segments are often required. In those cases, the extended Simpson's rule is needed.

The extended Simpson's rule breaks the interval up into  $N$  equal size divisions using the formula:

$$\int_{x_1}^{x_N} f(x')dx' = \frac{h}{3} (f_1 + 4f_2 + 2f_3 + 4f_4 + \dots + 2f_{N-2} + 4f_{N-1} + f_N) + O\left(\frac{1}{N^4}\right) \quad (4.4.5)$$

where  $f_i = f(x_i)$ ,  $x_i = x_1 + ih$ , and  $h = (x_N - x_0)/N$ . The error term at the end of the extended Simpson's rule is again not included in calculations. The material included in this section demonstrates only two techniques for numerical integration. For more information and more techniques, the interested reader should see [Press et al.(2007)Press, Teukolsky, Vetterling, and Flannery].

In order to implement the extended Simpson's rule, follow these steps:

1. Define the function to be integrated.
2. Define the lower and upper limits of the integral ( $a$  and  $b$ ) and the stepsize  $h$ .

**120 ■ Classical Mechanics: A Computational Approach**

3. Create an array whose elements are the terms in the sum of the right-hand side of (4.4.5).
  - (a) Define the first element of the array to be  $f(a)$ , and the last element to be  $f(b)$ .
  - (b) Define the other elements as follows: the elements with an even index  $i$  are evaluated using the expression  $4f(a + ih)$ , and elements with an odd index  $i$  are evaluated using the expression  $2f(a + ih)$ .
4. Sum the array and multiply the result by  $h/3$ .

Algorithm 3 uses Python to demonstrate these three different algorithms: the trapezoidal rule, the simple Simpson’s rule, and the extended Simpson’s rules, for this simple integral:

$$\int_0^3 (x^2 + 2x + 3) dx = 27 \tag{4.4.6}$$

In Algorithm 3 we changed some notation for simplicity, and instead of using an array for the extended Simpson’s rule, we used a for-loop to demonstrate an alternative method. The variables  $a$  and  $b$  are the lower and upper limits of integration, respectively. Recall that lines which begin with `#` are comments and are not executed by the Python interpreter. Also recall that lines beginning with the word `OUTPUT` show the output of the code, and are not part of the code. Because we are going to need to evaluate  $f(x) = x^2 + 2x + 3$  many times, we begin the program by defining the function being integrated. Notice that the trapezoidal and basic Simpson’s rules are simple enough to perform in one or two lines of code each. Furthermore, we included the `print` command to show how formatting can be done in Python.

The calculation of the extended Simpson’s rule is more involved than the other two methods. The coefficients used in the extended Simpson’s rule depend on the index of each term in the right-hand side of (4.4.5). The coefficient of  $f_i$  is 4 when  $i$  is even, and is 2 when  $i$  is odd, except for  $i = 1$  and  $i = N$ . In order to handle these dependencies, we used the `if` and `elif` conditional statements. The `if` statement tests whether or not a condition is true and if so, the indented line is executed.

The first `if` statement tests to see if  $i = 1$  or  $i = N$ . If so, then the term  $hf(x)/3$  is added to the sum. Notice that the variable `extended` contains the sum of the terms on the right-hand side of (4.4.5), and that the command `+=` adds to and overwrites the variable `extended`. If  $i \neq 1$  or  $N$ , then the next conditional statement is evaluated.

The next conditional statement is `elif`, or “else-if”, which tests to see if  $i$  is even by calculating `i modulo 2`. In Python, the “`%`” symbol is used for the modulo calculation. The `elif` statement provides an additional conditional statement to test after the first `if` statement, and is useful when there are multiple possible outcomes for a conditional (in this case  $i$  can be even, odd, 1, or  $N$ ). Finally, if none of the above conditional statements return true, then  $i$  must be odd. The `else` conditional statement tells Python what to execute if no other prior conditionals are true; in this case the term  $\frac{4}{3}hf(x)$  is added to the sum.

Notice that the trapezoidal rule performs poorly in our example. However, the basic Simpson’s rule (using two divisions) works better than the extended Simpson’s rule which uses 10,000 divisions. This is a good example of higher order calculations not always performing better than lower order ones. You should experiment with the number of divisions used in Algorithm 3 to find how the accuracy of the calculation changes with the number of divisions,  $N$ .

Next we will demonstrate how to use numerical integration algorithms to obtain the center of mass of continuous mass distributions. In the following example, we will use

```

def f(x):
    return x**2 + 2*x + 3

a = 0
b = 3

#trapezoid rule
trapezoid = (f(b)-f(a))*(b-a)/2.0
print("The trapezoid result is: {:.5f}".format(trapezoid))
OUIPUT: The trapezoid result is: 22.50000

#simpon 's rule
c = (a+b)/2
h = b - c
simpson = h*(f(a)/3.0 + 4*f(c)/3.0+f(b)/3.0)
print("The Simpson result is: {:.5f}".format(simpson))
OUIPUT: The Simpson result is: 27.00000

#extended simpson 's rule
N = 10000 #number of steps

extended = 0
h = (b-a)/(N-1) #need N steps starting at an index of 0
for i in range(0,N):
    x = a + i*h
    if i == 0 or i == N-1:
        extended += h*f(x)/3.0
    elif i % 2 == 0:
        extended += 2.0*h*f(x)/3.0
    else:
        extended += 4.0*h*f(x)/3.0

print("The extended Simpson result is: {:.5f}".format(extended))
OUIPUT: The extended Simpson result is: 26.99820

```

**Algorithm 3:** Examples of the trapezoidal and Simpson’s rule in Python.

122 ■ Classical Mechanics: A Computational Approach

Mathematica’s built-in numerical integration routine *NIntegrate*, and Python’s numerical integration routine *nquad* from the SciPy library.

**Example 4.6: The center of mass of a non-uniform isosceles triangle**

Consider the triangle in Example 4.3, but now with a non-uniform density  $\sigma = 2y$ , where the constant has units that produce a density measured in  $\text{kg}/\text{m}^2$ . Compute the center of mass for the triangle with  $a = 1$  m.

**Solution:**

In this case, the solution is very similar to that of Example 4.4, but now the triangle has a non-uniform density. The limits of integration are the same, with  $a = 1$ . However,  $dm$  is changed due to the non-uniform density. In this case,  $dm = \sigma dx dy = 2y dx dy$ . Because the density changes only in the  $y$ -direction, we still expect that the  $x$ -coordinate of the center of mass is zero. The integrals we need to solve are:

$$Y = \frac{1}{M} \int_{y=0}^{y=a/\sqrt{2}} \int_{x=y-a/\sqrt{2}}^{a/\sqrt{2}-y} 2y^2 dx dy$$

$$M = \int_{y=0}^{y=a/\sqrt{2}} \int_{x=y-a/\sqrt{2}}^{a/\sqrt{2}-y} 2y dx dy$$

First, we use Mathematica’s *NIntegrate* command to solve these integrals. *NIntegrate* uses an adaptive algorithm, which recursively subdivides the interval of integration as needed, instead of choosing the number of divisions ahead of time (as done in the extended Simpson’s rule). This can be advantageous when the function to be integrated changes rapidly over some intervals while remaining nearly constant over others. Notice that in the Mathematica program, we included the result for  $X$  to show that it is zero.

```
a = 1.0;
mass = NIntegrate [2 * y, {y, 0, a / sqrt(2)}, {x, y - a / sqrt(2), a / sqrt(2) - y}];
Y = Integrate [2 * y^2, {y, 0, a / sqrt(2)}, {x, y - a / sqrt(2), a / sqrt(2) - y}] / mass

OUTPUT: 0.353553

X = Integrate [2 * y * x, {y, 0, a / sqrt(2)}, {x, y - a / sqrt(2), a / sqrt(2) - y}] / mass

OUTPUT: 0.
```

Next we perform the integration using the numerical integration routine, *nquad*, from SciPy’s *integrate* library. The command *nquad* uses the command *quad* over multiple variables of integration. The command *quad* also comes from SciPy’s *integrate* library, and uses a FORTRAN 77 library called QUADPACK [Piessens et al. (1983) Piessens, Doncker-Kapenga, Überhuber, and Kahaner] to evaluate an integral. The code to perform the integration is shown in Algorithm 2.

Notice that we can include the bounds as a function; this allows us to use the non-constant limits of integration necessary to describe the sides of the triangle. In addition to finding  $Y$ , we also calculated  $X$  to show that  $X = 0$ , as expected.



```

from scipy import integrate
import numpy as np

a = 1.0

def sigma(x,y):
    return 2.0*y

def fx(x,y):
    return x*sigma(x,y)

def fy(x,y):
    return y*sigma(x,y)

def y_bounds():
    return [0,1/np.sqrt(2.0)]

def x_bounds(y):
    return [y-1/np.sqrt(2.0),1/np.sqrt(2.0)-y]

Y = integrate.nquad(fy,[x_bounds,y_bounds])[0]
X = integrate.nquad(fx,[x_bounds,y_bounds])[0]
mass = integrate.nquad(sigma,[x_bounds,y_bounds])[0]

print('The x-coordinate is: '+str(X/mass))
print('The y-coordinate is: '+str(Y/mass))

```

OUTPUT

```

The x-coordinate is: 0.0
The y-coordinate is: 0.35355339059327384

```

**Algorithm 4:** Example of center of mass calculation in Python.

## 4.5 MOMENTUM OF A SYSTEM OF MULTIPLE PARTICLES

Next, consider a system of  $N$  point particles with a total mass  $M$ . We begin by calculating the net force acting on the whole system. We know from Section 4.1 that the internal forces between particles cancel out, and that the net force acting on the system is the sum of external forces. The net force  $\mathbf{F}$  acting on the particles is:

$$\mathbf{F} = \sum_{i=1}^N m_i \ddot{\mathbf{r}}_i = \frac{d^2}{dt^2} \sum_{i=1}^N m_i \mathbf{r}_i = \frac{d^2}{dt^2} M \mathbf{R} = M \ddot{\mathbf{R}} \quad (4.5.1)$$

where we used the definition of the center of mass from (4.3.1). Therefore, the system of particles moves like a single particle of mass  $M$ , acted upon by the external forces. We can also compute the net momentum of the system of particles:

$$\mathbf{P} = \sum_{i=1}^N \mathbf{p}_i = \sum_{i=1}^N m_i \dot{\mathbf{r}}_i = \frac{d}{dt} \sum_{i=1}^N m_i \mathbf{r}_i = \frac{d}{dt} M \mathbf{R} = M \dot{\mathbf{R}} \quad (4.5.2)$$