# ENERGY – ELASTIC AND INELASTIC COLLISIONS IN TWO DIMENSIONS<sup>\*</sup>

# R.G. (Dick) Baldwin

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License  $3.0^{\dagger}$ 

## Abstract

This module explains elastic and inelastic collisions in two dimensions in a format that is accessible to blind students.

# **1** Table of Contents

- Preface (p. 2)
  - · General (p. 2)
  - $\cdot$  Prerequisites (p. 2)
  - Viewing tip (p. 2)
    - \* Figures (p. 3)
    - \* Listings (p. 3)
  - · Supplemental material (p. 3)
- Discussion (p. 3)
- Example scenarios (p. 7)
  - · One-dimensional scenarios (p. 7)
    - \* The rear end car crash (p. 7)
    - \* A perfectly inelastic car crash (p. 11)
  - Two-dimensional scenarios (p. 14)
    - \* Elastic collision between two pucks on friction-free ice (p. 15)
    - \* Perfectly inelastic collision between objects with odd angles (p. 19)
    - \* Rotating the axes for simplification (p. 23)
- Run the scripts (p. 30)
- Resources (p. 30)
- Miscellaneous (p. 30)

<sup>\*</sup>Version 1.2: Oct 8, 2012 1:53 pm -0500

<sup>&</sup>lt;sup>†</sup>http://creativecommons.org/licenses/by/3.0/

# 2 Preface

## 2.1 General

This module is part of a collection (see http://cnx.org/content/col11294/latest/ $^1$ ) of modules designed to make physics concepts accessible to blind students. The collection is intended to supplement but not to replace the textbook in an introductory course in high school or college physics.

This module explains elastic and inelastic collisions in two dimensions in a format that is accessible to blind students.

## 2.2 Prerequisites

In addition to an Internet connection and a browser, you will need the following tools (as a minimum) to work through the exercises in these modules:

- A graph board for plotting graphs and vector diagrams ( http://www.youtube.com/watch?v=c8plj9UsJbg  $^2$  ).
- A protractor for measuring angles ( http://www.youtube.com/watch?v=v-F06HgiUpw <sup>3</sup> ).
- An audio screen reader that is compatible with your operating system, such as the NonVisual Desktop Access program (NVDA), which is freely available at http://www.nvda-project.org/ $^4$ .
- A refreshable Braille display capable of providing a line by line tactile output of information displayed on the computer monitor (http://www.userite.com/ecampus/lesson1/tools.php<sup>5</sup>).
- A device to create Braille labels. Will be used to label graphs constructed on the graph board.

The minimum prerequisites for understanding the material in these modules include:

- A good understanding of algebra.
- An understanding of the use of a graph board for plotting graphs and vector diagrams (http://www.youtube.com/watch<sup>6</sup>).
- An understanding of the use of a protractor for measuring angles ( http://www.youtube.com/watch?v=v-F06HgiUpw <sup>7</sup> ).
- A basic understanding of the use of sine, cosine, and tangent from trigonometry (http://www.clarku.edu/~djoyce/trig/<sup>8</sup>).
- An introductory understanding of JavaScript programming ( http://www.dickbaldwin.com/tocjscript1.htm  $^{9}$  and http://www.w3schools.com/js/default.asp  $^{10}$  ).
- An understanding of all of the material covered in the earlier modules in this collection.

## 2.3 Viewing tip

I recommend that you open another copy of this document in a separate browser window and use the following links to easily find and view the figures and listings while you are reading about them.

<sup>&</sup>lt;sup>1</sup>http://cnx.org/content/col11294/latest/

<sup>&</sup>lt;sup>2</sup>http://www.youtube.com/watch?v=c8plj9UsJbg

<sup>&</sup>lt;sup>3</sup>http://www.youtube.com/watch?v=v-F06HgiUpw

<sup>&</sup>lt;sup>4</sup>http://www.nvda-project.org/

<sup>&</sup>lt;sup>5</sup>http://www.userite.com/ecampus/lesson1/tools.php

<sup>&</sup>lt;sup>6</sup>http://www.youtube.com/watch?v=c8plj9UsJbg

 $<sup>^{7}</sup>$  http://www.youtube.com/watch?v=v-F06HgiUpw

<sup>&</sup>lt;sup>8</sup>http://www.clarku.edu/~djoyce/trig/

 $<sup>^9</sup> http://www.dickbaldwin.com/tocjscript1.htm$ 

 $<sup>^{10} \</sup>rm http://www.w3 schools.com/js/default.asp$ 

## 2.3.1 Figures

- Figure 1 (p. 5). Equations for collisions of two objects in two-dimensional space.
- Figure 2 (p. 11). Output for the rear end car crash.
- Figure 3 (p. 14). Output for the perfectly inelastic car crash.
- Figure 4 (p. 19). Output for an elastic collision between two pucks on friction-free ice.
- Figure 5 (p. 23). Output for a perfectly inelastic collision between objects with odd angles.
- Figure 6 (p. 29). Output for rotation of the axes for simplification.

## 2.3.2 Listings

- Listing 1 (p. 7). The rear end car crash.
- Listing 2 (p. 11). A perfectly inelastic car crash.
- Listing 3 (p. 15). Elastic collision between two pucks on friction-free ice.
- Listing 4 (p. 19) . Perfectly inelastic collision between objects with odd angles.
- Listing 5 (p. 24). Rotation of the axes for simplification.

#### 2.4 Supplemental material

I recommend that you also study the other lessons in my extensive collection of online programming tutorials. You will find a consolidated index at www.DickBaldwin.com  $^{11}$ .

## **3** Discussion

I have touched on collisions in one dimension in earlier modules. I will deal with collisions in a more rigorous manner in this module, and will also extend the analysis to two dimensions.

#### NOTE: Facts worth remembering – Types of collisions

An **elastic** collision is one in which the total kinetic energy is the same before and after the collision.

An **inelastic** collision is one in which the final kinetic energy is less than the initial kinetic energy.

A **perfectly inelastic** collision is one that results in the two objects sticking together. The decrease of kinetic energy in a perfectly inelastic collision is as large as possible (consistent with the conservation of momentum).

**Momentum** is conserved regardless of whether the collision is elastic or inelastic.

#### A general solution for elastic collisions

I will provide you with three equations that apply in general to elastic collisions in two dimensions. However, as you will see, there are more than three variables involved in such collisions. With only three equations, you can only solve for three unknowns. Therefore, in order to solve the general problem, the values of all the other variables must be known.

The two-dimensional solution can be applied to one-dimensional problems by constraining the directions of motion of the two objects to either be the same or to differ by 180 degrees. If possible, such problems should be structured to cause the directions to be along the x-axis. This will often simplify the solution.

#### A general solution for inelastic collisions

The case for inelastic collisions is more restrictive than the case for elastic collisions. Only two of the equations mentioned above apply to inelastic collisions. As a result, you can only solve for two unknown values for an inelastic collision. The values for all of the other variables must be known.

<sup>&</sup>lt;sup>11</sup>http://www.dickbaldwin.com/toc.htm

## **Collision** equations

The equations for collisions of two objects in two-dimensional space are shown in Figure 1 (p. 5). Note that it is assumed that the two objects constitute an isolated system – that is, a closed system that is not subject to external interactions. This requires that both the magnitude and the direction of the momentum for the system be the same at the beginning and the end of the collision.

Equations for collisions of two objects in two-dimensional space.

```
Using conservation of momentum alone, we have two
equations, allowing us to solve for two unknowns.
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
Using conservation of kinetic energy for the elastic
case gives us one additional equation, allowing us
to solve for three unknowns.
0.5*m1*u1^2 + 0.5*m2*u2^2 = 0.5*m1*v1^2 + 0.5*m2*v2^2
Velocities can be decomposed into their x and
y-components using the following equations:
u1x = u1*cos(a1)
u1y = u1*sin(a1)
u2x = u2*cos(a2)
u2y = u2*sin(a2)
v1x = v1*cos(b1)
v1y = v1*sin(b1)
v2x = v2*\cos(b2)
v2y = v2*sin(b2)
Substitution yields the following for the two momentum
equations:
m1*u1*cos(a1) + m2*u2*cos(a2) = m1*v1*cos(b1) + m2*v2*cos(b2)
m1*u1*sin(a1) + m2*u2*sin(a2) = m1*v1*sin(b1) + m2*v2*sin(b2)
where:
m1 and m2 are the masses of the two objects in kg
u1 and u2 are the magnitudes of the initial
velocities of the two objects. Velocities are
measured in meters/second
v1 and v2 are the magnitudes of the final velocities
of the two objects
u1x, u1y, v1x, and v1y are the x and y-components
of the initial and final velocities in 2D space.
a1 and a2 are angles that describe the initial trp:/cnx.org/content/m98431/1.2/
directions of the two objects through 2D space.
Angles are measured counter-clockwise relative to
the positive x-axis
```

#### What do the equations imply?

The first two equations based on momentum in Figure 1 (p. 5) require that the combined momentum of the two objects be the same, along each axis, before and after the collision. Thus, one of the equations deals with momentum along the horizontal axis and the other equation deals with momentum along the vertical axis.

The terms in the two equations are:

- m1 and m2 represent the masses of object 1 and object 2.
- u1x and u2x represent the components of the velocities of the two objects along the horizontal or x axis before the collision.
- u1y and u2y represent the components of the velocities of the two objects along the vertical or y axis before the collision.
- v1x and v2x represent the components of the velocities of the two objects along the horizontal or x axis after the collision.
- v1y and v2y represent the components of the velocities of the two objects along the vertical or y axis after the collision.

#### The third or energy equation

As indicated in Figure 1 (p. 5), this equation can only be used for the case of an elastic collision. This equation requires that the total kinetic energy of the two objects before the collision be equal to the total kinetic energy of the two objects after the collision.

In addition to the masses described above, this equation introduces the following terms:

- u1 and u2 represent the magnitudes of the velocities of the two objects before the collision.
- v1 and v2 represent the magnitudes of the velocities of the two objects after the collision.

## Definition of the angles

In order to compute the horizontal and vertical components of the velocities before and after the equations, you must know the directions in which the objects are moving before and after the collision. Those directions appear as angles in Figure 1 (p. 5) where

- a1 and a2 are angles measured counter clockwise relative to the horizontal axis that represent the direction of travel of each of the objects respectively before the collision.
- b1 and b2 are angles measured counter clockwise relative to the horizontal axis that represent the direction of travel of each of the objects respectively after the collision.

#### Ten variables

As you can see in Figure 1 (p. 5), these equations involve ten variables. That means that in order for a solution to be possible, the values for eight of the variables must be know for an inelastic collision, and the values for seven of the variables must be known for an elastic collision.

#### Not conceptually difficult

If you believe in the laws of conservation of momentum and conservation of energy on which these three equations are based, the solution to collision problems is not conceptually difficult.

However, depending on which variables are known, which are unknown, and whether the collision is elastic, inelastic, or perfectly inelastic, you can come up with equations that are difficult to solve from an algebra/trigonometry viewpoint.

## Axis rotation

For the case where none of the given directions are along the x- axis or the y-axis, you can sometimes simplify the algebraic/trigonometric problem by rotating the axis so as to place one of those directions along the x-axis or the y-axis. This will often cause one or more terms in the set of equations to go to zero, thus simplifying the solution to the set of equations.

Having done that, you can rotate the axis by the same amount in the opposite direction at the end to cause the final solution to apply to the original axes. I will present an example of this in the next section.

## 4 Example scenarios

#### Several examples

I will use the information in Figure 1 (p. 5) to analyze several scenarios involving collisions in both one dimension and two dimensions in this section.

## The use of JavaScript

All of these examples could be solved using the Google calculator. However, several steps are involved and I find it easier to keep things organized and perform the steps in the correct order by using JavaScript to compute and display the solution.

Note, however, that JavaScript will only do the arithmetic for you. You must still do the algebra/trigonometry yourself. In these examples, I will usually work through the algebra in comment sections and switch to actual code when it is time to compute and display one or more values.

## 4.1 One-dimensional scenarios

The first one-dimensional scenario involves an automobile accident.

## 4.1.1 The rear end car crash

The description as well as the solution to the problem are shown in Listing 1 (p. 7).

```
Listing 1: The rear end car crash.
```

```
<!----- File JavaScript01.html ----->
<html><body>
<script language="JavaScript1.3">
```

#### /\*

```
This script simulates car #2 rear-ending car #1 in a
one-dimensional elastic collision while car #1 was stopped.
The script computes and displays the speed of car #2
immediately before the collision under the assumption that
car #1 was moving at 20 m/s just after the collision.
```

Using conservation of momentum alone, we have two equations, allowing us to solve for two unknowns.

```
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
```

Using conservation of kinetic energy for the elastic case gives us one additional equation, allowing us to solve for three unknowns.

 $0.5*m1*u1^2 + 0.5*m2*u2^2 = 0.5*m1*v1^2 + 0.5*m2*v2^2$ 

Variables: m1, m2, u1, u2, v1, v2, a1, a2, b1, b2 \*/

document.write("Start Script </br>");

```
//Solve for u2 and v2 for an elastic collision
var m1 = 1500; //kg
var m2 = 2000; //kg
//Velocities before the collision
var u1 = 0;//meters per second - standing still
var u2;//unknown value to be determined
//Velocities after the collision
var v1 = 20://meters/second
var v2;//unknown value to be determined
//Angles
var a1 = 0;//car was not moving
var a2 = 0;//moving straight ahead
var b1 = 0;//moving straight ahead
var b2 = 0;//moving straight ahead
//Convert angles to radians
A1 = a1*Math.PI/180;
A2 = a2*Math.PI/180;
B1 = b1*Math.PI/180;
B2 = b2*Math.PI/180;
//Compute and print the x and y components of velocity
u1x = u1*Math.cos(A1)
uly = ul*Math.sin(A1)
//u2x = u2*Math.cos(A2)//unknown
//u2y = u2*Math.sin(A2)//unknown
v1x = v1*Math.cos(B1)
v1y = v1*Math.sin(B1)
//v2x = v2*Math.cos(B2)//unknown
//v2y = v2*Math.sin(B2)//unknown
document.write("x and y components of velocity</br>");
document.write("u1x = " + u1x.toFixed(3) + "</br>");
document.write("u1y = " + u1y.toFixed(3) + "</br>");
document.write("v1x = " + v1x.toFixed(3) + "</br>");
document.write("v1y = " + v1y.toFixed(3) + "</br>");
/*
Prepare the equations for use in solving the problem.
Given the following three equations
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
0.5*m1*u1^2 + 0.5*m2*u2^2 = 0.5*m1*v1^2 + 0.5*m2*v2^2
```

Eliminate all of the components for which the above printout

shows zero or for which the given values show zero. 0 + m2\*u2x = m1\*v1x + m2\*v2x0 + m2\*u2y = 0 + m2\*v2y $0 + 0.5 \times m^2 \times u^2 = 0.5 \times m^1 \times v^1^2 + 0.5 \times m^2 \times v^2^2$ Although it isn't totally obvious from the equations, at this point we need to recognize that because all velocities are defined to occur along the x-axis, all of the terms in the middle equation above that deals with the y-component of velocity must be zero. Therefore, we can eliminate that equation entirely. We also need to recognize that because there are no velocity components along the y-axis, the velocity components along the x-axis are actually the magnitudes of those velocity components. Thus, u2x = u2. Now we will make the substitutions and eliminate terms with a value of 0 in the process, yielding m2\*u2 = m1\*v1 + m2\*v2 $0.5*m2*u2^2 = 0.5*m1*v1^2 + 0.5*m2*v2^2$ Substituting known values into the two equations yields 2000\*u2 = 1500\*20 + 2000\*v2 $2000 * u^2 = 1500 * 20 * 20 + 2000 * v^2 = 2000 * v^2 =$ Simplifying the two equations yields u2 = 15 + v2u2\*u2 = 300 + v2\*v2Now we need to eliminate one equation through substitution v2 = u2 - 15u2\*u2 = 300 + (u2 - 15)\*(u2 - 15)u2\*u2 = 300 + u2\*u2 - 30\*u2 + 225u2\*u2 - 300 - u2\*u2 + 30\*u2 - 225 = 0u2\*u2 - u2\*u2 + 30\*u2 - 300 - 225 = 030\*u2 - 525 = 030\*u2 = 525\*/ //Compute and print the first speed value document.write("Speed values</br>"); u2 = 525/30;document.write("u2 = " + u2.toFixed(2) + " m/s</br>");

```
Substituting this value back into an earlier energy equation
yields
v2*v2 = u2*u2 - 300;
*/
//Compute and display the second speed value
v2 = Math.sqrt(u2*u2 - 300);
document.write("v2 = " + v2.toFixed(2) + " m/s < /br >");
//Check the answers for conservation of momentum
document.write("Check for conservation of momentum</br>");
var mou = m1*u1 + m2*u2;
var mov = m1*v1 + m2*v2;
document.write("mou = " + mou.toFixed(0) + " Kg*m/s</br>");
document.write("mov = " + mov.toFixed(0) + " Kg*m/s</br>");
//Check the answer for elastic collision
var keIn = 0.5*m1*u1*u1 + 0.5*m2*u2*u2;
var keOut = 0.5*m1*v1*v1 + 0.5*m2*v2*v2;
document.write("Check for conservation of energy</br>");
document.write("keIn = " + keIn.toFixed(0)
 + " Kg*m^2/s^2</br>");
document.write("keOut = " + keOut.toFixed(0)
 + " Kg*m^2/s^2</br>");
document.write("End Script");
</script>
```

```
</body></html>
```

#### The output

The output produced by this script is shown in Figure 2 (p. 11).

The only comments that I will make in addition to the comments in Listing 1 (p. 7) are that the values displayed for mou, mov, keIn, and keOut near the end of Figure 2 (p. 11) show that both momentum and kinetic energy were conserved. Therefore, the requirements for an elastic collision were met.

The meaning of each of those terms is as follows:

- mou represents total momentum before the crash
- mov represents total momentum after the crash
- keIn represents total kinetic energy before the crash
- keOut represents total kinetic energy after the crash

#### Output for the rear end car crash.

Start Script x and y components of velocity u1x = 0.000u1y = 0.000v1x = 20.000v1y = 0.000------Speed values u2 = 17.50 m/sv2 = 2.50 m/s\_\_\_\_\_ Check for conservation of momentum mou = 35000 Kg\*m/s mov = 35000 Kg\*m/s Check for conservation of energy  $keIn = 306250 Kg*m^{2}/s^{2}$  $keOut = 306250 Kg*m^{2}/s^{2}$ End Script

Figure 2: Output for the rear end car crash.

#### 4.1.2 A perfectly inelastic car crash

The description and the solution to another rear end car crash are provided in Listing 2 (p. 11). Whereas the previous car-crash scenario described an elastic collision, this scenario describes a perfectly inelastic collision. In this crash, the two cars become entangled and move forward as a single object following the collision. Therefore, this is an example of a perfectly inelastic collision.

Listing 2: A perfectly inelastic car crash.

```
<!----- File JavaScript10.html ----->
<html><body>
<script language="JavaScript1.3">
/*
This script simulates car #2 rear-ending car #1 in a
one-dimensional perfectly inelastic collision while car #1 was
stopped.
```

```
Assume that car #2 was moving at 17.5 m/s immediately before
the collision. The cars became entangled and moved as a single
object in a straight line following the collision. Find
the velocity of the two objects immediately following the
collision. Find the total momentum of the two objects
immediately before and immediately after the collision.
```

Using conservation of momentum alone, we have two equations, allowing us to solve for two unknowns.

```
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
```

```
However, because the cars are moving along the x-axis, all
terms in the second equation must be zero. This limits us to
only the first equation shown above. Note also that because
this is an inelastic collision, we can't use the equation
based on conservation of energy.
```

```
Variables:
m1, m2, u1, u2, v1, v2, a1, a2, b1, b2
*/
```

```
document.write("Start Script </br>");
```

```
var m1 = 1500;//kg
var m2 = 2000;//kg
//Velocities before the collision
var u1 = 0;//meters per second - standing still
var u2 = 17.5;//meters per second
//Velocities after the collision
var v1;//unknown but v2 = v1
var v2;//unknown value to be determined
//Angles
var a1 = 0;//car was not moving
var a2 = 0;//moving straight ahead
var b1;//unknown but not needed
```

```
var b2;//unknown but not needed
//Convert angles to radians
```

```
A1 = a1*Math.PI/180;
A2 = a2*Math.PI/180;
//B1 = b1*Math.PI/180;
```

```
//B2 = b2*Math.PI/180;
```

```
//Compute and print the initial x and y components of velocity.
// Because both cars are moving along the x-axis, all
// y-components are zero and all x components are equal to the
// magnitude.
```

u1x = u1u1y = 0u2x = u2u2y = 0//v1x = v1//unknownv1y = 0//v2x = v2//unknownv2y = 0document.write("Initial velocities</br>"); document.write("u1x = " + u1x.toFixed(2) + "</br>"); document.write("u2x = " + u2x.toFixed(2) + "</br>"); /\* Prepare the equations for use in solving the problem. Given the following three equations m1\*u1x + m2\*u2x = m1\*v1x + m2\*v2xm1\*u1y + m2\*u2y = m1\*v1y + m2\*v2y $0.5*m1*u1^2 + 0.5*m2*u2^2 = 0.5*m1*v1^2 + 0.5*m2*v2^2$ Only the first equation can be used for an inelastic collision moving along the x-axis. This gives us the following equation to work with. m1\*u1x + m2\*u2x = m1\*v1x + m2\*v2xEliminate all of the components which are known to be zero. m2\*u2x = m1\*v1x + m2\*v2xFor a perfectly inelastic collision, v2=v1 yielding m2\*u2x = m1\*v1x + m2\*v1x, or m2\*u2x = (m1 + m2)\*v1xRearranging terms gives the following: \*/ //Compute and print the final speed values v1x = m2\*u2x/(m1 + m2)v2x = v1x;//required for a perfectly inelastic collision document.write("Final speed values</br>"); document.write("v1x = " + v1x.toFixed(2) + " m/s</br>"); document.write("v2x = " + v2x.toFixed(2) + " m/s</br>"); //Check the answer for conservation of momentum

document.write("Check for conservation of momentum</br>");

#### The output

Figure 3 (p. 14) shows the output for the car crash scenario portrayed by Listing 2.

## Output for the perfectly inelastic car crash.

Figure 3: Output for the perfectly inelastic car crash.

## 4.2 Two-dimensional scenarios

Now we will examine some two-dimensional scenarios. In these scenarios, the objects are free to move in a plane described by horizontal and vertical coordinates. Instead of the directions of motion being limited to only forward and backward, any object can move in any direction from 0 to 360 degrees.

#### 4.2.1 Elastic collision between two pucks on friction-free ice

The description and the solution to a scenario involving an elastic collision between two pucks on friction-free ice is shown in Listing 3 (p. 15). As you will see:

- The pucks are free to move in two dimensions.
- Even though this is an elastic collision, there were only two unknowns, so it wasn't necessary to use the energy-conservation equation to find a solution.

#### Three unknowns can be challenging

Elastic collisions involving three unknowns, particularly those where one or more angles are unknown, can be challenging from an algebraic/trigonometric viewpoint. Those solutions typically involve a quadratic equation containing trigonometric functions. This is not one of those especially challenging scenarios.

#### Listing 3: Elastic collision between two pucks on friction-free ice.

```
<!----- File JavaScript02.html ----->
<html><body>
<script language="JavaScript1.3">
/*
Obj1 and Obj2 are identical and are on friction-free ice. Obj1
has an initial velocity of 2.0 m/s in the direction
of O degrees. Obj2 is at rest. Obj1 collides elastically with
Obj2 and Obj1 moves off at 1.0 m/s at an angle of 60 degrees
north of east. What is the speed and direction of Obj2 after
the collision?
Using conservation of momentum alone, we have two
equations, allowing us to solve for two unknowns.
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
Using conservation of kinetic energy for the elastic
case gives us one additional equation, allowing us
to solve for three unknowns.
0.5*m1*u1^2 + 0.5*m2*u2^2 = 0.5*m1*v1^2 + 0.5*m2*v2^2
The specifications tell us that this is an elastic collision,
so we are free to use any or all of the three equations to
solve the problem. In this case, we have three equations but
only two unknowns, v2 and b1, so we won't need all three
of the equations.
Note that the specifications don't specify the values of the
masses, but do specify that they are the same. Therefore, we
will be able to cancel them out of all three equations.
Variables:
m1, m2, u1, u2, v1, v2, a1, a2, b1, b2
```

```
*/
document.write("Start Script </br>");
//var m1 = unknown;//kg
//var m2 = m1; //kg
var u1 = 2;//meters per second
var u2 = 0;//meters per second -- at rest
var v1 = 1;//meter per second
var v2;//unknown -- to find
var a1 = 0;//degrees
var a2 = 0;//degrees -- at rest
var b1 = 60;//degrees
var b2;//unknown -- to find
//Convert angles to radians
A1 = a1*Math.PI/180;
A2 = a2*Math.PI/180;
B1 = b1*Math.PI/180;
//B2 = b2*Math.PI/180;//unknown
//Compute the initial x and y components of velocity
u1x = u1*Math.cos(A1)
uly = u1*Math.sin(A1)
u2x = u2*Math.cos(A2)
u2y = u2*Math.sin(A2)
v1x = v1*Math.cos(B1)
v1y = v1*Math.sin(B1)
//v2x = v2*Math.cos(B2) //unknown
//v2y = v2*Math.sin(B2) //unknown
/*
For the special case of m2=m1 the three equations can be
written as follows (after canceling out the m-terms):
u1x + u2x = v1x + v2x
u1y + u2y = v1y + v2y
0.5*u1*u1 + 0.5*u2*u2 = 0.5*v1*v1 + 0.5*v2*v2
Identify all terms that are known to be zero
u1x + 0 = v1x + v2x
0 + 0 = v1y + v2y
0.5*u1*u1 + 0.5*0*0 = 0.5*v1*v1 + 0.5*v2*v2
Removing those terms yields
u1x = v1x + v2x
0 = v1y + v2y
0.5*u1*u1 = 0.5*v1*v1 + 0.5*v2*v2
```

```
Divide through the energy equation by 0.5 for simplification
u1x = v1x + v2x
0 = v1v + v2v
u1*u1 = v1*v1 + v2*v2
Substitute known values
2 = 0.5 + v2x
0 = 0.866 + v2y
u1*u1 + = v1*v1 + v2*v2
This problem can be solved without using the energy equation.
I will use the energy equation later to check the results.
*/
//Compute the components, magnitude, and direction of the
// final velocity of obj #2
v2x = 2 - 0.5;
v2y = -0.866;
v2 = Math.sqrt(v2x*v2x + v2y*v2y);
b2 = getAngle(1.5, -0.866);
//Display the known values along with the results.
document.write("u1x = " + u1x.toFixed(2) + " m/s</br>");
document.write("u1y = " + u1y.toFixed(2) + " m/s</br>");
document.write("u2x = " + u2x.toFixed(2) + " m/s < /br >");
document.write("u2y = " + u2y.toFixed(2) + " m/s</br>");
document.write("v1x = " + v1x.toFixed(2) + " m/s</br>");
document.write("v1y = " + v1y.toFixed(2) + " m/s</br>");
document.write("v2x = " + v2x.toFixed(2) + " m/s</br>");
document.write("v2y = " + v2y.toFixed(3) + " m/s</br>");
document.write("u1 = " + u1.toFixed(2) + " m/s</br>");
document.write("u2 = " + u2.toFixed(2) + " m/s</br>");
document.write("v1 = " + v1.toFixed(2) + " m/s < /br >");
document.write("v2 = " + v2.toFixed(2) + " m/s < /br >");
document.write("b2 = " + b2.toFixed(2) + " degrees</br>");
```

```
//Check the answers assuming that the mass of the objects
// is one Kg each.
var moux = u1x + u2x;
var movx = v1x + v2x;
var mouy = u1y + u2y;
var movy = v1y + v2y;
```

```
document.write("moux = " + moux.toFixed(2) + " Kg*m/s</br>");
document.write("movx = " + movx.toFixed(2) + " Kg*m/s</br>");
document.write("mouy = " + mouy.toFixed(2) + " Kg*m/s</br>");
document.write("movy = " + movy.toFixed(2) + " Kg*m/s</br>");
//Check to confirm an elastic collision
var u1 = Math.sqrt(u1x*u1x + u1y*u1y);
var u2 = Math.sqrt(u2x*u2x + u2y*u2y);
var v1 = Math.sqrt(v1x*v1x + v1y*v1y);
var v2 = Math.sqrt(v2x*v2x + v2y*v2y);
var mou = 0.5*u1*u1 + 0.5*u2*u2;
var mov = 0.5*v1*v1 + 0.5*v2*v2;
document.write("mou = " + mou.toFixed(2) + " Kg*m/s</br>");
document.write("mov = " + mov.toFixed(2) + " Kg*m/s</br>");
//The purpose of this function is to receive the adjacent
// and opposite side values for a right triangle and to
// return the angle in degrees in the correct quadrant.
function getAngle(x,y){
 if((x == 0) \&\& (y == 0)){
    //Angle is indeterminate. Just return zero.
   return 0;
 }else if((x == 0) && (y > 0)){
   //Avoid divide by zero denominator.
   return 90;
 }else if((x == 0) && (y < 0)){
    //Avoid divide by zero denominator.
   return -90;
 }else if((x < 0) && (y >= 0)){
   //Correct to second quadrant
   return Math.atan(y/x)*180/Math.PI + 180;
 }else if((x < 0) & (y <= 0)){
   //Correct to third quadrant
   return Math.atan(y/x)*180/Math.PI + 180;
 }else{
   //First and fourth quadrants. No correction required.
   return Math.atan(y/x)*180/Math.PI;
 }//end else
}//end function getAngle
document.write("End Script");
</script>
</body></html>
```

## The output

The output for this scenario is shown in Figure 4 (p. 19).

Output for an elastic collision between two pucks on friction-free ice.

Start Script u1x = 2.00 m/su1y = 0.00 m/su2x = 0.00 m/su2y = 0.00 m/sv1x = 0.50 m/sv1y = 0.87 m/sv2x = 1.50 m/sv2y = -0.866 m/su1 = 2.00 m/su2 = 0.00 m/sv1 = 1.00 m/sv2 = 1.73 m/sb2 = -30.00 degrees \_\_\_\_\_ moux = 2.00 Kg\*m/smovx = 2.00 Kg\*m/smouy = 0.00 Kg\*m/smovy = 0.00 Kg\*m/smou = 2.00 Kg\*m/smov = 2.00 Kg\*m/sEnd Script

Figure 4: Output for an elastic collision between two pucks on friction-free ice.

Hopefully the comments in the script will be sufficient to explain the solution to this problem.

## 4.2.2 Perfectly inelastic collision between objects with odd angles

Listing 4 (p. 19) provides the description and the solution to a scenario involving a perfectly inelastic collision between two objects in a friction-free environment moving at odd angles. By odd angles, I mean that neither object is moving along either the x-axis or the y-axis, either before or after the collision.

Listing 4: Perfectly inelastic collision between objects with odd angles.

<!----- File JavaScript03.html ----->
<html><body>
<script language="JavaScript1.3">

```
/*
Obj1 with a mass of 1 Kg and an initial velocity of 3000 m/s
in a direction 67 degrees north of east collides in a perfectly
inelastic manner with Obj2, whose mass is also 1 Kg and whose
initial velocity is 2000 m/s in a direction 45 degrees north of east.
Calculate (1) the angle of motion of the combined bodies,
and (2) the magnitude of the momentum of the combined bodies
after the collision.
Using conservation of momentum alone, we have two
equations, allowing us to solve for two unknowns.
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
Variables:
m1, m2, u1, u2, v1, v2, a1, a2, b1, b2
*/
document.write("Start Script </br>");
var m1 = 1; //kg
var m2 = 1; //kg
var u1 = 3000;//meters per second
var u2 = 2000;//meters per second
var v1;//unknown -- to be found
var v2;//unknown -- to be found
var a1 = 67;//degrees
var a2 = 45;//degrees
var b1;//unknown -- to be found
var b2;//unknown -- to be found
//Perfectly inelastic collision so v2=v1 and b2=b1
//Convert angles to radians
A1 = a1*Math.PI/180;
A2 = a2*Math.PI/180;
//B1 = b1*Math.PI/180;//unknown
//B2 = b2*Math.PI/180;//unknown
//Compute the x and y components of velocity
u1x = u1*Math.cos(A1)
u1y = u1*Math.sin(A1)
u2x = u2*Math.cos(A2)
u2y = u2*Math.sin(A2)
//v1x = v1*Math.cos(B1)//unknown
//v1y = v1*Math.sin(B1)//unknown
```

```
//v2x = v2*Math.cos(B2)//unknown
//v2y = v2*Math.sin(B2)//unknown
/*
For the special case of m2=m1=1 and v2=v1 (perfectly inelastic
collision) we can simplify the equations to the following:
u1x + u2x = 2*v1x
u1y + u2y = 2*v1y
*/
//Rearranging terms yields
v1x = (u1x + u2x)/2
v1y = (u1y + u2y)/2
//Knowing the x and y components of the final velocity, we can
// find the angle and magnitude as
b1 = getAngle(v1x,v1y);
v1 = Math.sqrt(v1x*v1x + v1y*v1y);
//Compute the momentum after the collision
var Px = v1x*(m1 + m2);
var Py = v1y*(m1 + m2);
var Pmag = Math.sqrt(Px*Px + Py*Py);
//Display the results
document.write("b1 = " + b1.toFixed(1) + " degrees</br>");
document.write("v1 = " + v1.toFixed(0) + " m/s < /br >");
document.write("v1x = " + v1x.toFixed(0) + " m/s</br>");
document.write("v1y = " + v1y.toFixed(0) + " m/s</br>");
document.write("Px = " + Px.toFixed(0) + " Kg*m/s</br>");
document.write("Py = " + Py.toFixed(0) + " Kg*m/s</br>");
document.write("Pmag = " + Pmag.toFixed(0) + " Kg*m/s</br>");
//Check the answer for perfect inelastic collision
v2x = v1x;
v2y = v1y;
v2 = v1;
var moux = u1x + u2x;
var movx = v1x + v2x;
var mouy = u1y + u2y;
var movy = v1y + v2y;
var mou = Math.sqrt(moux * moux + mouy * mouy);
var mov = Math.sqrt(movx * movx + movy * movy);
document.write("moux = " + moux.toFixed(0) + " Kg*m/s</br>");
document.write("movx = " + movx.toFixed(0) + " Kg*m/s</br>");
document.write("mouy = " + mouy.toFixed(0) + " Kg*m/s</br>");
document.write("movy = " + movy.toFixed(0) + " Kg*m/s</br>");
document.write("mou = " + mou.toFixed(0) + " Kg*m/s</br>");
```

```
document.write("mov = " + mov.toFixed(0) + " Kg*m/s</br>");
//The purpose of this function is to receive the adjacent
// and opposite side values for a right triangle and to
// return the angle in degrees in the correct quadrant.
function getAngle(x,y){
 if((x == 0) \&\& (y == 0)){
   //Angle is indeterminate. Just return zero.
   return 0;
 }else if((x == 0) && (y > 0)){
   //Avoid divide by zero denominator.
   return 90;
 }else if((x == 0) && (y < 0)){
   //Avoid divide by zero denominator.
   return -90;
 }else if((x < 0) && (y >= 0)){
   //Correct to second quadrant
   return Math.atan(y/x)*180/Math.PI + 180;
 }else if((x < 0) & (y <= 0)){
   //Correct to third quadrant
   return Math.atan(y/x)*180/Math.PI + 180;
 }else{
   //First and fourth quadrants. No correction required.
   return Math.atan(y/x)*180/Math.PI;
 }//end else
}//end function getAngle
```

document.write("End Script");
</script>

```
</body></html>
```

#### The output

The output for the script shown in Listing 4 (p. 19) is provided in Figure 5 (p. 23).

Output for a perfectly inelastic collision between objects with odd angles.

```
Start Script
b1 = 58.2 degrees
v1 = 2456 \text{ m/s}
v1x = 1293 \text{ m/s}
v1y = 2088 \text{ m/s}
Px = 2586 \text{ Kg*m/s}
Py = 4176 \text{ Kg*m/s}
Pmag = 4912 Kg*m/s
_____
moux = 2586 \text{ Kg*m/s}
movx = 2586 \text{ Kg*m/s}
mouy = 4176 \text{ Kg*m/s}
movy = 4176 \text{ Kg*m/s}
mou = 4912 \text{ Kg*m/s}
mov = 4912 \text{ Kg*m/s}
_____
End Script
```

Figure 5: Output for a perfectly inelastic collision between objects with odd angles.

As before, I will allow the comments in Listing 4 (p. 19) serve as the explanation for the solution of this scenario.

#### 4.2.3 Rotating the axes for simplification

The scenario shown in Listing 5 (p. 24) illustrates how, in some cases, you can simplify the algebra/trigonometry by rotating the axes. This situation occurs when none of the directions involving the two objects lies along either the x or y axis. In those cases, you may be able to simplify the three equations that you develop by rotating the axes such that one of the directions lies along either the x or the y axis. That will often cause some of the terms in the equations to go to zero and drop out of the equations.

## A crash at an intersection

If you examine the scenario shown in Listing 5 (p. 24) carefully, you will see that it is the classic collision at the intersection of two streets that are perpendicular to one another. However, the streets don't run in north-south, east-west directions.

## Rotating the axes simplified the problem

In this particular case, rotating the axes so that one street runs east and west while the other street runs north and south simplifies the equations considerably.

#### Don't forget the rotate the axes back at the end

Of course, if you rotate the axes to simplify the solution, you must remember to rotate it back again, and correct the solution accordingly, once you have a solution. The procedure for doing that is illustrated in the scenario shown in Listing 5 (p. 24).

## Listing 5: Rotation of the axes for simplification.

```
<!----- File JavaScript04.html ----->
<html><body>
<script language="JavaScript1.3">
/*
Two objects collide in a perfectly inelastic collision. Obj1
has a mass of 10000 kg and is traveling 30 degrees north of
east at 15 m/s. Obj2 has a mass of 1500 kg and is traveling
30 degrees west of north, or 120 degrees at 25 m/s. Find
the direction that the two object move and the speed of that
movement following the collision.
Note that this script illustrates rotating the axis to simplify
the problem.
Using conservation of momentum alone, we have two
equations, allowing us to solve for two unknowns.
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
Variables:
m1, m2, u1, u2, v1, v2, a1, a2, b1, b2
*/
document.write("Start Script </br>");
var m1 = 10000; //kg
var m2 = 1500; //kg
var u1 = 15;//meters per second
var u2 = 25;//meters per second
var v1;//unknown -- to be found
var v2;//unknown -- to be found
var a1 = 30;//degrees
var a2 = 120;//degrees
var b1;//unknown -- to be found
var b2;//unknown -- to be found
//For a perfectly inelastic collision, v2=v1 and b2=b1
//Convert angles to radians
A1 = a1*Math.PI/180;
A2 = a2*Math.PI/180;
//B1 = b1*Math.PI/180;//unknown
//B2 = b2*Math.PI/180;//unknown
//Compute the x and y components of velocity
```

u1x = u1\*Math.cos(A1)

```
uly = ul*Math.sin(A1)
u2x = u2*Math.cos(A2)
u2y = u2*Math.sin(A2)
//v1x = v1*Math.cos(B1)//unknown
//v1y = v1*Math.sin(B1)//unknown
//v2x = v2*Math.cos(B2)//unknown
//v2y = v2*Math.sin(B2)//unknown
//Display the x and y components of initial velocity
document.write("x and y components of initial velocity</br>");
document.write("u1x = " + u1x.toFixed(2) + " m/s</br>");
document.write("u1y = " + u1y.toFixed(2) + " m/s</br>");
document.write("u2x = " + u2x.toFixed(2) + " m/s</br>");
document.write("u2y = " + u2y.toFixed(2) + " m/s</br>");
/*
Given the following two equations
m1*u1x + m2*u2x = m1*v1x + m2*v2x
m1*u1y + m2*u2y = m1*v1y + m2*v2y
For the special case of v2=v1 (perfectly inelastic
collision), the equations can be simplified to the following:
m1*u1x + m2*u2x = (m1 + m2)*v1x
m1*u1y + m2*u2y = (m1 + m2)*v1y
Replace the terms with known x and y velocity component values.
m1*13 + m2*(-12.5) = (m1 + m2)*v1x
m1*7.5 + m2*21.6 = (m1 + m2)*v1y
A judicious examination of the problem reveals that if we were
to rotate the axis by 30 degrees clockwise, we could cause
some of the terms in these two equations to go to zero. We will
subtract 30 degrees from the given angles at this point and
add that 30 degrees back into the results at the end.
*/
a1 = a1 - 30;
a2 = a2 - 30;
//Convert the new angles to radians
A1 = a1*Math.PI/180;
A2 = a2*Math.PI/180;
//Recompute the x and y components of velocity
u1x = u1*Math.cos(A1)
u1y = u1*Math.sin(A1)
```

u2x = u2\*Math.cos(A2)

```
u2y = u2*Math.sin(A2)
//Display the new x and y components of initial velocity
document.write(" New x and y components of velocity</br>");
document.write("u1x = " + u1x.toFixed(2) + " m/s</br>");
document.write("u1y = " + u1y.toFixed(2) + " m/s</br>");
document.write("u2x = " + u2x.toFixed(2) + " m/s < /br >");
document.write("u2y = " + u2y.toFixed(2) + " m/s</br>");
/*
Returning now to the special case of v2=v1 (perfectly inelastic
collision):
m1*u1x + m2*u2x = (m1 + m2)*v1x
m1*u1y + m2*u2y = (m1 + m2)*v1y
Replace the x and y components of velocity with the modified
values, two of which are now 0. This results in a simplification
of the equations.
m1*15 + m2*0 = (m1 + m2)*v1x
m1*0 + m2*25 = (m1 + m2)*v1y
Plugging in the values for mass and eliminating terms with
a zero value yields
10000*15 = (10000 + 1500)*v1x
1500*25 = (10000 + 1500)*v1y
*/
//Rearranging terms yields
v1x = (10000 * 15) / (10000 + 1500)
v_{1y} = (1500 * 25) / (10000 + 1500)
//Compute the magnitude and the angle of the velocity of the
// two objects following the collision.
v1 = Math.sqrt(v1x*v1x + v1y*v1y);
b1 = getAngle(v1x,v1y);
//Because this is a perfectly inelastic collision, v2=v1
// and b2=b1
v2 = v1;
b2 = b1;
//Display results for the modified angles
document.write("Results for modified angles</br>");
document.write("v1x = " + v1x.toFixed(1) + " m/s</br>");
document.write("v1y = " + v1y.toFixed(1) + " m/s</br>");
document.write("v1 = " + v1.toFixed(1) + " m/s</br>");
document.write("v2 = " + v2.toFixed(1) + " m/s < /br >");
document.write("b1 = " + b1.toFixed(1) + " degrees</br>");
document.write("b2 = " + b2.toFixed(1) + " degrees</br>");
```

```
/*
Now we need to rotate the axis by 30 degrees counter-clockwise
to correct for the original rotation by 30 degrees clockwise.
The magnitude of the final velocity is the same regardless of
the orientation of the axes. Therefore, we will add 30 degrees
to the values of b1 and b2, and use the new angles along with
the magnitude of the final velocity to recompute the x and y
components of the final velocity.
*/
b1 = b1 + 30;
b2 = b1;
v1x = v1*Math.cos(b1*Math.PI/180);
v1y = v1*Math.sin(b1*Math.PI/180);
//Display results for the corrected angle
document.write("Results for corrected angle</br>");
document.write("v1x = " + v1x.toFixed(1) + " m/s < /br >");
document.write("v1y = " + v1y.toFixed(1) + " m/s</br>");
document.write("v1 = " + v1.toFixed(1) + " m/s < /br >");
document.write("v2 = " + v2.toFixed(1) + " m/s</br>");
document.write("b1 = " + b1.toFixed(1) + " degrees</br>");
document.write("b2 = " + b2.toFixed(1) + " degrees</br>");
//Check the answer for a perfect inelastic collision. Must
// recognize that v2=v1 and correct the angles for a1 and a2.
v2x = v1x;
v2y = v1y;
u1x = u1*Math.cos((a1+30)*Math.PI/180);
uly = u1*Math.sin((a1+30)*Math.PI/180);
u2x = u2*Math.cos((a2+30)*Math.PI/180);
u2y = u2*Math.sin((a2+30)*Math.PI/180);
var moux = m1*u1x + m2*u2x;
var movx = m1*v1x + m2*v2x;
var mouy = m1*u1y + m2*u2y;
var movy = m1*v1y + m2*v2y;
var mou = Math.sqrt(moux * moux + mouy * mouy);
var mov = Math.sqrt(movx * movx + movy * movy);
document.write("Check the answers.</br>");
document.write("moux = " + moux.toFixed(0) + " Kg*m/s</br>");
document.write("movx = " + movx.toFixed(0) + " Kg*m/s</br>");
document.write("mouy = " + mouy.toFixed(0) + " Kg*m/s</br>");
document.write("movy = " + movy.toFixed(0) + " Kg*m/s</br>");
document.write("mou = " + mou.toFixed(0) + " Kg*m/s</br>");
```

```
document.write("mov = " + mov.toFixed(0) + " Kg*m/s</br>");
//The purpose of this function is to receive the adjacent
// and opposite side values for a right triangle and to
// return the angle in degrees in the correct quadrant.
function getAngle(x,y){
 if((x == 0) \&\& (y == 0)){
   //Angle is indeterminate. Just return zero.
   return 0;
 }else if((x == 0) && (y > 0)){
   //Avoid divide by zero denominator.
   return 90;
 }else if((x == 0) && (y < 0)){
    //Avoid divide by zero denominator.
   return -90;
 }else if((x < 0) && (y >= 0)){
   //Correct to second quadrant
   return Math.atan(y/x)*180/Math.PI + 180;
 }else if((x < 0) & (y <= 0)){
   //Correct to third quadrant
   return Math.atan(y/x)*180/Math.PI + 180;
 }else{
   //First and fourth quadrants. No correction required.
   return Math.atan(y/x)*180/Math.PI;
 }//end else
}//end function getAngle
//The purpose of the getRoots function is to compute and
// return the roots of a quadratic equation expressed in
// the format
11
   a*x^2 + b*x + c = 0
//The roots are returned in the elements of a two-element
// array. If the roots are imaginary, the function
// returns NaN for the value of each root.
function getRoots(a,b,c){
 var roots = new Array(2);
 roots[0] = (-b+Math.sqrt(b*b-4*a*c))/(2*a);
 roots[1] = (-b-Math.sqrt(b*b-4*a*c))/(2*a);
 return roots;
}//end getRoots
document.write("End Script");
</script>
</body></html>
```

## The output

The output for this scenario is shown in Figure 6 (p. 29).

Output for rotation of the axes for simplification.

Start Script x and y components of initial velocity u1x = 12.99 m/su1y = 7.50 m/su2x = -12.50 m/su2y = 21.65 m/s\_\_\_\_\_ New x and y components of velocity u1x = 15.00 m/su1y = 0.00 m/su2x = 0.00 m/su2y = 25.00 m/s------Results for modified angles v1x = 13.0 m/sv1y = 3.3 m/sv1 = 13.4 m/sv2 = 13.4 m/sb1 = 14.0 degrees b2 = 14.0 degrees Results for corrected angle v1x = 9.7 m/sv1y = 9.3 m/sv1 = 13.4 m/sv2 = 13.4 m/sb1 = 44.0 degrees b2 = 44.0 degrees Check the answers. moux = 111154 Kg\*m/smovx = 111154 Kg\*m/s mouy = 107476 Kg\*m/smovy = 107476 Kg\*m/s mou = 154616 Kg\*m/s mov = 154616 Kg\*m/s ------End Script

Figure 6: Output for rotation of the axes for simplification.

Once more, I will allow the comments in Listing 4 (p. 19) to serve as the explanation for the solution of this scenario.

# 5 Run the scripts

I encourage you to run the scripts that I have presented in this lesson to confirm that you get the same results. Copy the code for each script into a text file with an extension of html. Then open that file in your browser. Experiment with the code, making changes, and observing the results of your changes. Make certain that you can explain why your changes behave as they do.

# 6 Resources

I will publish a module containing consolidated links to resources on my Connexions web page and will update and add to the list as additional modules in this collection are published.

# 7 Miscellaneous

This section contains a variety of miscellaneous information.

## NOTE: Housekeeping material

- Module name: Energy Elastic and Inelastic Collisions in Two Dimensions for Blind Students
- File: Phy1215.htm
- Keywords:
  - $\cdot$  physics
  - $\cdot$  accessible
  - · accessibility
  - $\cdot$  blind
  - · graph board
  - $\cdot$  protractor
  - $\cdot$  screen reader
  - · refreshable Braille display
  - · JavaScript
  - $\cdot$  trigonometry
  - $\cdot$  one-dimensional
  - $\cdot$ two-dimensional
  - $\cdot$  collision
  - $\cdot$  elastic collision
  - $\cdot$  inelastic collision
  - $\cdot$  perfectly inelastic collision
  - $\cdot$  kinetic energy
  - $\cdot$  conservation of momentum
  - $\cdot$  conservation of energy

NOTE: **Disclaimers:** Financial : Although the Connexions site makes it possible for you to download a PDF file for this module at no charge, and also makes it possible for you to purchase a pre-printed version of the PDF file, you should be aware that some of the HTML elements in this module may not translate well into PDF.

I also want you to know that I receive no financial compensation from the Connexions website even if you purchase the PDF version of the module. **Affiliation** : I am a professor of Computer Information Technology at Austin Community College in Austin, TX.

-end-