|  |
| --- |
| PHYS1521  **Math and Physics for Games**  Realistic Projectile  Simulation Report  Digital Media and IT  School of Applied Sciences and Technology |

|  |  |
| --- | --- |
| **Name(s):** | Alex Kinnear and Pierre Gravelle |
| **Section:** | A01 |
| **Date:** | March 22nd 2017 |
| **Instructor:** | Allan Anderson |



*Fig. 1.* Screenshot of Projectile Motion Flash Simulation.

From “Projectile Motion” by Splung.com at <http://www.splung.com/content/sid/2/page/projectiles>

Table of Contents

[Introduction 1](#_Toc479160359)

[Concept 1](#_Toc479160360)

[Torque 2](#_Toc479160361)

[What Is Torque? 2](#_Toc479160362)

[Axis of rotation: 3](#_Toc479160363)

[Center of Mass 4](#_Toc479160364)

[How do We Calculate Center of Mass? 5](#_Toc479160365)

[How Is It Relevant to Game Programming? 5](#_Toc479160366)

[Drag and Lift 6](#_Toc479160367)

[What Is Drag? 6](#_Toc479160368)

[How do We Calculate Drag? 6](#_Toc479160369)

[What is Lift? 7](#_Toc479160370)

[How do we Calculate Lift? 7](#_Toc479160371)

[How Are These Relevant to Game Programming? 8](#_Toc479160372)

[Conclusion 8](#_Toc479160373)

[References 9](#_Toc479160374)

**Table of Figures**

[Fig. 1. Screenshot of Projectile Motion Flash Simulation. i](#_Toc479160313)

[Fig. 2-B. Application of Torque cont. 2](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160314)

[Fig. 2-A. Application of torque 2](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160315)

[Fig. 2-C. Application of Torque cont. 2](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160316)

[Fig. 3. Diagram of Center of Mass 3](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160317)

[Fig. 4-A. Irregularly Shaped Object 4](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160318)

[Fig. 4-B. Calculating the Void Space 4](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160319)

[Fig. 4-C. Calculating the Center of Mass 5](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160320)

[Fig. 5. Table of Drag Coefficients 6](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160321)

[Fig. 8. Air Pressure Formula 6](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160322)

[Fig. 7. Air Density Formula 6](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160323)

[Fig. 6. Drag Force Formula 6](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160324)

[Fig. 9. Lift, Drag, and Velocity’s Relation 7](#_Toc479160325)

[Fig. 10. Lift Formula 8](file:///C:\Users\Pierre\Documents\GitHub\PhysicsResearchandSimulationProject\A01%20-%20Alex%20and%20Pierre%20-%20Report.docx#_Toc479160326)

# Introduction

Projectiles are moving objects that have plenty of physics associated with them. Our project is to create a virtual and realistic simulation involving launching projectiles at several angles. We chose this topic because it is familiar to us due to our Math and Physics for Games class at NAIT. We also see this as an excellent opportunity to translate our calculations into visual results.

There are plenty of video games that use projectile-like assets. The physics behind these assets are a lot more complicated than they may seem at first. For example, the game “Angry Birds” uses birds as projectiles and they are affected by multiple forces. All these forces must be programmed in, otherwise the projectile may act differently than a user may expect. For example, a projectile may slow down faster than a user expected, hence ruining their attempt at the shot.

This report will highlight our efforts towards creating a realistic projectile simulation and will go into detail on how each physics concept involved affects the projectile.

# Concept

The simulation will involve two different scenarios. One scenario will have the projectile being launched from a cannon, and the other scenario having a ball being flung after multiple circular rotations around a center point. Our simulation will allow the user to change some of the variables in the simulation using a menu, this will demonstrate how well our mathematical calculations will react to the changes instead of being hard-coded values.

In our work, gravity’s acceleration will always be considered as -9.81 m/s2 as this is the constant we have been using in our physics class.

We’ll be using multiple concepts in our simulation. The concepts that we’ll be using that we’ve already learnt about in our Math and Physics for Games class include:

* Momentum Conservation between two colliding objects
* Linear Projectile Motion
* Rotational Projectile Motion

The following concepts are new to our group and will be the key points in our report. They will be explored in detail later:

* Torque
* Drag Force and Lift

## Torque

### What Is Torque?

Torque is a force that causes an object to rotate. The amount it rotates is proportional to the torque applied and the object’s resistance to the rotation.

τ = Torque

I = the moment of Inertia

α = angular acceleration

We can find the amount of torque that’s being applied to an object by using the formulas below.

or ∅)

τ = Torque

F˔ = the perpendicular component of the force applied

F is the force applied

r = distance to the axis of rotation

∅ = the angle the force is applied

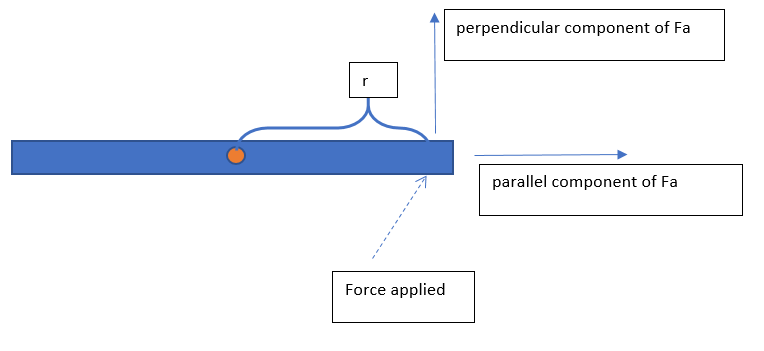
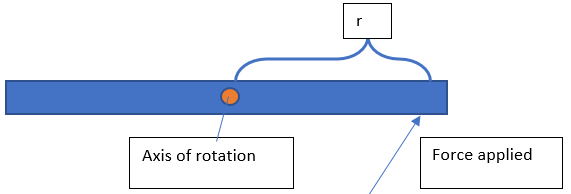
This looks like:

Fig. 2-B. Application of Torque cont.

By Alex K. (2017)

Fig. 2-A. Application of torque

By Alex K. (2017)

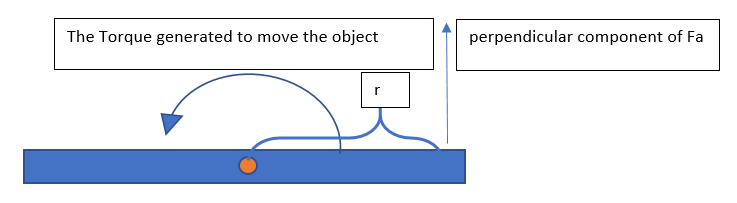
Calculus-based mathematics are needed to calculate the moment of Inertia. However, the moment of inertia depends on the shape of the object being rotated. You’ll see the shape of an object also affects other things such as Drag Force and Lift Force (seen later in this report).

Fig. 2-C. Application of Torque cont.

By Alex K. (2017)

There are resources to find the formula for the moment of inertia for a specific shape. However here are the basic formulas:

**m** = mass  
**r** = distance to axis of rotation.

This formula will find the Inertia of a point object (mass is all at one point), add all point masses to get the sum.

For an object that has its mass evenly distributed, its moment of inertia can be:

Where L is the length and the axis of rotation is at the end of the object. (A cylindrical shape)

Where the axis is at the center. (A cylindrical shape)

Where this is a cylinder with the axis through the center

For a sphere rotating with the axis through the center

If you’re unwilling to learn calculus but still want to get its moment of Inertia, you can find it by splitting the object into ‘common geometric shapes’. Find the moment of Inertia from the center of one of these shapes and have the rest as point masses.

### Axis of rotation:

To find the axis of rotation determine if the object is fixed to something or if it’s free.

If it’s fixed to something, that point would be the axis of rotation. For example, if a tire is fixed to the axle of a vehicle, then the tire’s axis of rotation would be where the tire is connected to the axel.

If it’s free, its axis of rotation would be the center of mass of the object. So if a sheet of plywood is freefalling in a void space, and an object is thrown at it, when they collide the object would apply a torque to the sheet of plywood, causing it to rotate around its **center of mass.**

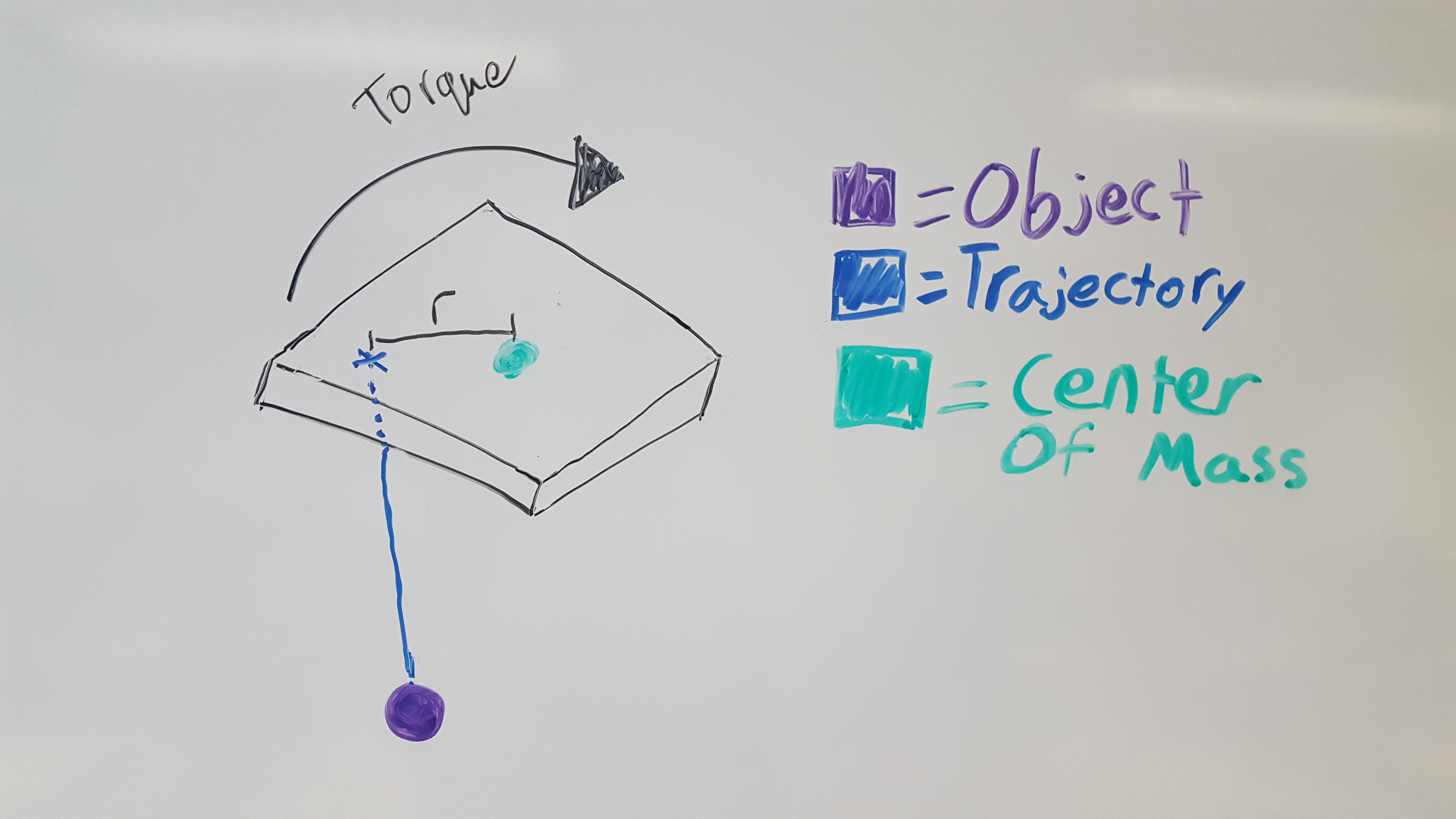


Fig. 3. Diagram of Center of Mass

By Pierre G. (2017)

### Center of Mass

The Center of Mass is the “center of weight” in any object, regardless of shape or size.

For primitive shapes, such as spheres, circles, squares and cubes which are symmetrical, the Center of Mass will be the very center of the object because the mass is distributed evenly around it.

However, for more complex objects it would be necessary to determine the Center of Mass before being able to determine how it would react to Torque or another force. *(Hatton, n.d.)* A good example of torque with “no fixed axis” is balance. Gravity works on the object’s Center of Mass as though it were the pivot point, and then rotates the object accordingly.

The easiest way to get the Center of Mass is to create a reference point (X, Y or X, Y and Z coordinate graph) and put the object onto the reference.

If the object is made of the same materials, you don’t need to worry about what materials or mass make up the object as all of it would wind up having the same effect on the object’s Center of Mass.

Once you have your reference and object, ‘split’ the object into primitive shapes then find and collect each of their Center of Mass.

In Fig. 4-A we can see that this object’s shape can be separated into a square, a large rectangle, and a triangle.

In Fig. 4-B, object number 3 is the empty (or void) space of the object in Fig. 4-A.

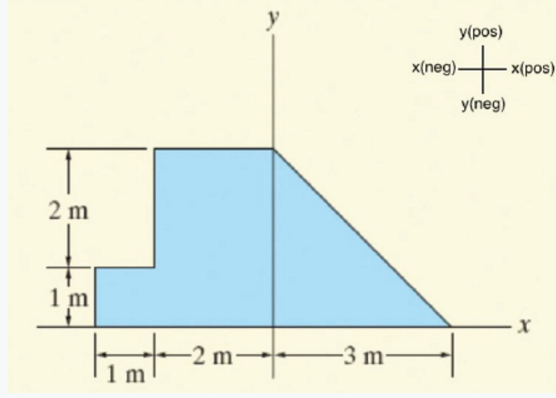


Fig. 4-A. Irregularly Shaped Object

From “How to Determine the Center of Mass of Any Load” by Laura Hatton

Fig. 4-B. Calculating the Void Space

From “How to Determine the Center of Mass of Any Load” by Laura Hatton

If the object has holes or sections of ‘nothingness’ you can treat them as being part of the primitive shape on the condition that you create a shape for them and find its Center of Mass for those holes and note them as being “Void space” (for subtracting their effect later).

So, in short, we collect for each object’s Center of Mass and how much weight it has. (The amount of space it takes, therefore the Area for 2D objects and the Volume for 3D objects).

### How do We Calculate Center of Mass?

D is the coordinate (X, Y and Z) and W is its ‘weight’

or

So if the original object was split like so (1 and 2 being where mass is and 3 being “void” space), this is what it would look like.

**NOTE:** We subtracted the object ‘C’ as it was “empty” space that was included by the other objects.

Fig. 4-C. Calculating the Center of Mass

From “How to Determine the Center of Mass of Any Load” by Laura Hatton

How Is It Relevant to Game Programming?

Torque and Center of Mass are both related to game programming because they are physics engine properties.

Where torque would be used for any rotating object (whether that starts, accelerates or does not affect the rotation speed) in order to determine its new rotation speed.

Center of Mass would be used for balancing any object and as a rotation point.

## Drag and Lift

### What Is Drag?

**Drag** (also known as **Air Resistance**) is a force that pushes in the opposite direction of an object’s velocity. Air resistance is dependent on an object’s velocity *(Rit.edu, n.d.).* So, the faster an object is moving, the more air resistance it will have. For example, a cube being dropped from a 500m high building will start with very little air resistance. However, as the object falls it starts to accelerate and the air resistance becomes a lot stronger.

Since air resistance is a force moving against our velocity, this means that the object’s speed will eventually hit a maximum value. The velocity of the object will no longer increase, and the air resistance will stay constant as well.

Drag is also dependent on the shape of the object being used. A sphere for example, will have less air resistance than a cube would. The reason for this is the **Drag Coefficient**.

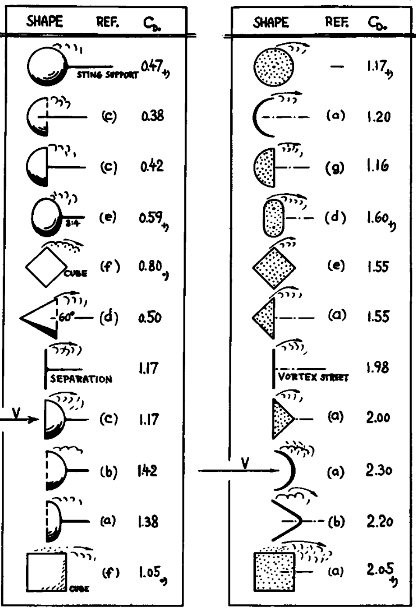
The Drag Coefficient is a number that represents how much drag an object will have; it is dependent on the shape of the object being used. *(engineeringtoolbox.com, n.d.).* See Fig. 2 for a table full of drag coefficients and the shapes they are associated to.

Fig. 5. Table of Drag Coefficients

From “Fluid-Dynamic Drag” by Sighard Hoerner (1965)

Modern vehicles take drag coefficient into serious consideration, that’s why they have a very smooth and aerodynamic shape. Patrick E. George from HowStuffWorks.com gives the example of the Toyota Prius. He writes:

*“Among other efficient characteristics, its Cd of .26 helps it achieve very high mileage. In fact, reducing the Cd of a car by just 0.01 can result in a 0.2 miles per gallon (.09 kilometers per liter) increase in fuel economy.”*   
- Patrick E. George (March 2009)

### How do We Calculate Drag?

To find the drag force, we must use the Drag Force formula (See Fig. 6.). The formula takes factors into account that we haven’t seen in class, such as **air density**, and the **frontal area** of the object.   
The density of the air that the object is travelling through is measured in kg/m3 (kilograms per cubic meter). It is calculated using the Air Density formula (See Fig. 7) However, as you may have seen from the figure, to find the density of the air we need to know the air pressure. Lucky for us, there’s another formula (See Fig. 8) to calculate this.

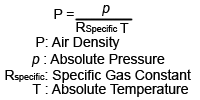
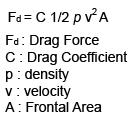
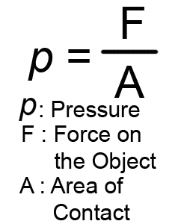


Fig. 8. Air Pressure Formula

By Pierre G. (2017)

Fig. 7. Air Density Formula

By Pierre G. (2017)

Fig. 6. Drag Force Formula

By Pierre G. (2017)

### What is Lift?

Lift is a force that carries an object upwards while travelling high speeds. Some may think it’s the opposite of drag, however that isn’t the case. Drag goes in the opposite direction of velocity, while Lift helps carry the velocity of the object upwards. In the case of a perfect sphere, it would have to rotate to have lift. The reason for this is that a perfect sphere would evenly distribute the air around it, making the pressure of the air above the sphere and the pressure of the air bellow the sphere equal. Equal pressure means there’s no lift force.

Lift’s direction is dependent on the velocity’s direction. You can see this in the diagram I created bellow (Fig. 9.). As velocity moves downwards, the lift of the object follows along with it. Drag also follows along because as stated before, drag is a force that moves in the opposite direction of velocity.

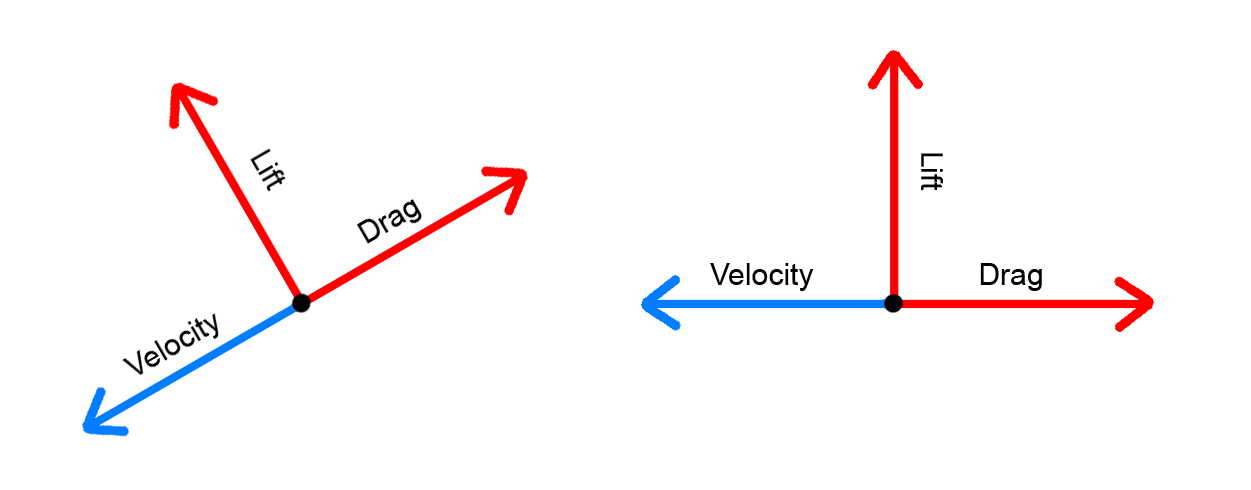


Fig. 9. Lift, Drag, and Velocity’s Relation

By Pierre G. (2017)

### How do we Calculate Lift?

In order to calculate Lift we’ll need to use multiple formulas from the Torque section seen of this report.

First, we need to find the Alpha that’s being applied to this object. We can find this by rearranging the formula for Sum of Torque seen in the torque section. Don’t forget that Alpha is a symbol representing Angular Acceleration and that it’s calculated in rads/s2.

Once we have calculated the Alpha of this object, we are now able to calculate the Omega of the rotation, which is the Angular Velocity (also known as rotation speed) in rads/s. All we need is to know the amount of time that the object is rotating for.

We need this Angular Velocity to find the revolutions that the object does around its pivot point per second. This is also known as Revolutions per Second (RPS). In order to convert our rads/s into rev/s we need to understand that a full revolution (rotation) in radians is equivalent to 2π. So we simply need to divide our Omega by 2π in order to find our RPS.

We’re almost done. Now all we need to do is use the Lift formula seen below in order to calculate lift!

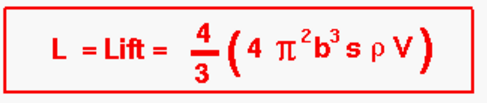


Fig. 10. Lift Formula

From “Ideal Lift of a Spinning Ball” By Nancy Hall

In this formula **b** is equal to the radius of the object, **s** is equal to the RPS that we just found, **p** is equal to the air density (which we can find using the formula shown in Fig. 4.), and finally **V** is the velocity that the object is travelling at.

### How Are These Relevant to Game Programming?

In video game programming, the developers may want the most realistic physics possible. Calculating the drag force and lift force increases the realism of the video game. Plenty of video games are dedicated to using realistic physics, such as the Battlefield series. Bullet trajectories are calculated realistically. A larger projectile (like a missile from a rocket launcher) falls quicker and doesn’t go as far as a small bullet does.

Developers may also base the concept of their game on physics as well. In games that use physics to solve puzzles, the player should know what to expect. Realistic physics due to drag force and lift can make a drastic difference between a “Game Over” screen and a “You Win!” screen.

# Conclusion

In conclusion, projectiles have many more physics concepts attached to them than most people care to think about. Torque affects the rotation of the object, which in turn may extend the projectile’s trajectory using Drag and Lift forces. Therefore, using all these concepts and calculations in our game programming will drastically increase the realism of it.

We learnt how Torque influences the rotation of an object while it’s in mid-air and how external forces (such as objects) can increase or reduce the Torque of an object in mid-air. The concept of Torque could be included into the PHYS1521 course during the rotational motion concept, as that is the point when we learn about angular velocity and tangential velocity. Rotational motion teaches us what happens to an object while it’s rotating around a point, this is very similar to Torque.

The concepts of Drag and Lift taught us about additional forces that are being applied on a projectile during motion. Drag force moves against the object’s velocity, and Lift force would help keep the object in the air for longer periods of time. This concept could be included into the PHYS1521 course during linear motion by introducing external forces during a projectile’s travel. For example, wind has a velocity and applies force on the object while it’s in the air, therefore our calculations would have to reflect that.

While these extra physics concepts may seem like an addition to projectile physics, they were there all along and therefore aren’t additional. We plan on using these in our simulation to increase the realism of our simulation.

# References

Put all your references here.