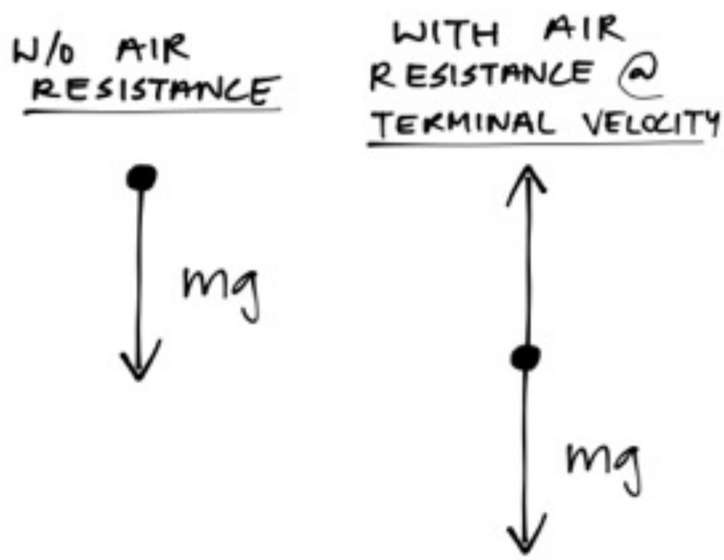
Kushal Singh
Mr. Greene
AP Computer Science (1)
10 February 2015

Background information: Suppose you have a tennis ball as an example of a projectile. When the tennis ball is hit with an initial velocity v0 (m/s), a downward acceleration of 9.8 m/s^2 acts on it. The greater the cross-sectional area (A), the greater the coefficient of drag (Cd). Drag impacts the trajectory of the projectile by preventing it from following a perfectly parabolic path. The density of air (p) stays constant.



Problem Statement:

Range is calculated via the formula: (v^2/g) * sin (2θ) where the projectile is launched from a flat surface and returns to that same surface height after a period of time. The projectile can travel the furthest when the angle is 45° because the formula for range is (v^2/g) * sin (2θ) and sin 90° equals 1, which is the maximum value of sin. Therefore, the “optimum angle” should be 45°.

This experiment will also test the impact of air resistance on the x-range of a projectile, which should theoretically lessen the range since air drag resists/is in the opposite direction of motion.

Finally, this experiment will also test the impact of air resistance on the y-range of a projectile.

Assumptions:

This project includes two parts: projectile motion with air resistance and projectile motion without air resistance. In both cases, the same angles are tested ($\pi/12$, $\pi/6$, $\pi/4$, $\pi/3$, π). Furthermore, in both cases, the same velocity magnitude was used with a value of 40 m/s. To accomplish the task at hand, basic physics formulas of projectile motion were used in addition to certain key constants. These values are listed below:

Coefficient of Drag	Mass	Density	Cross-Sectional Area	Time
0.4	0.057 kg	1.2 kg/m^3	0.0064 m^2	0.04 seconds

Equations

Projectile Motion without Air Drag

1. vx = v * cos(theta)
2. vy = v * sin(theta)
3. x = x + (vx * Δt)
4. y = y - (vy * Δt)

Projectile Motion with Air Drag

1. v = √((vx)^2 + (vy)^2)
2. θ = arctan(vy/vx)
3. a = (0.5 * Drag * Area * velocity * velocity * density)/mass;
4. ax = a * cos(theta)
5. ay = a * sin(theta)
6. vx = vx - (ax * Δt)
7. vy = vy + (g * Δt) - (ay * Δt)
8. x = x + (vx * Δt)
9. y = y + (vy * Δt)

Results:

X-Range Air Drag vs. X-Range No Air Drag

Xinitial = 0

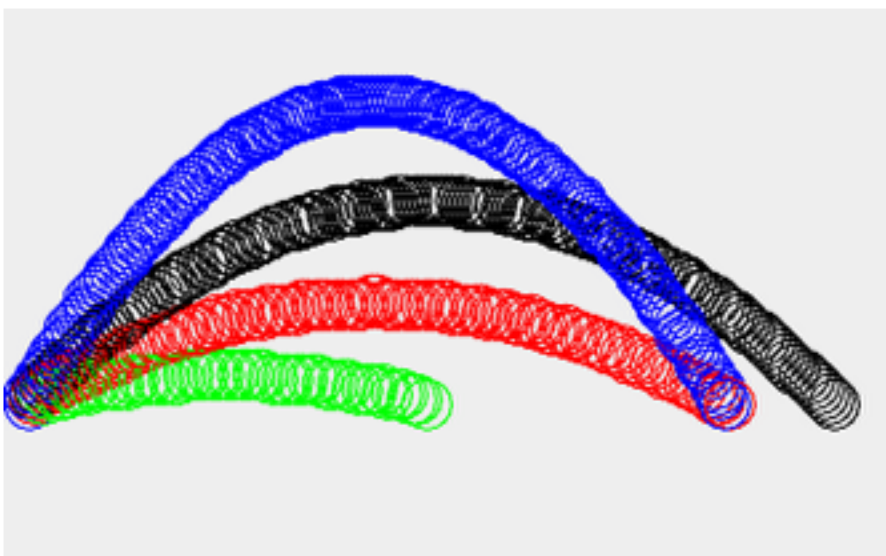
Yinitial = 150

Yfinal = 150

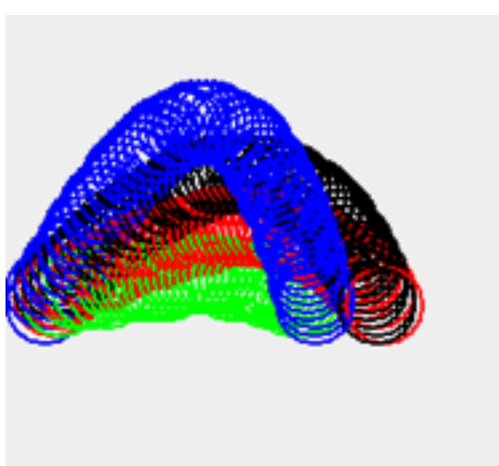
Vinitial = 40

Angle (θ)	X-Range with Air Drag	X-Range without Air Drag
π/12	68	156
π/6	86	278
π/4	86	322
π/3	70	278
π	0	0

X-Range without Air Drag



X-Range with Air Drag



Y-Range Air Drag vs. Y-Range No Air Drag
Xinitial = 0
Yinitial = 150
Vinitial = 40

Angle (θ)	ΔY with Air Drag	ΔY without Air Drag
π/12	8	12
π/6	20	42
π/4	34	82
π/3	48	122
π	0	0

No Air Drag Case

To calculate the maximum y-height that the projectile reaches, I placed a conditional in the beginning of the run() method. The code is as follows: if (projII.vy > 0){…}
The reason I did this is because at the maximum height, the y-component of the velocity vector is 0 and vy starts off as positive.

Air Drag Case

To calculate the maximum y-height that the projectile reaches, I placed a conditional in the beginning of the run() method. The code is as follows: if (projII.vy < 0){…}
The reason I did this is because at the maximum height, the y-component of the velocity vector is 0 and vy starts off as negative.

Conclusion:

From the results, it is quite evident that the optimum launch angle is indeed 45°. When the tennis ball was launched at 45°, it traveled an x-distance of 322 m, without air drag, which was the furthest distance traveled of all the other angles.

Likewise, air resistance did have an impact on the projectile as it decreased the x-range of the projectile. Regardless of the angle measure, the x-distance that the projectile traveled when it encountered air resistance was significantly less than the x-distance that it traveled when it didn't encounter air resistance.

Similar results were witnessed in the impact of air resistance on the y-range of the projectile. Once again, regardless of the angle measure, the y-distance that the projectile traveled when it encountered air resistance was less than the y-distance that it traveled when it didn't encounter air resistance.

Appendix:

```
import java.awt.*;
import java.applet.*;
import javax.swing.*;
import java.util.*;
import java.awt.event.*;

public class projmotion extends JApplet implements Runnable, MouseListener {
   Thread t;
   int timeStep = 40;
   Projectile proj1;
   Projectile proj2;
   Projectile proj3;
   Projectile proj4;
   Projectile proj5;

   Projectile_NoAirDrag projI;
   Projectile_NoAirDrag projII;
   Projectile_NoAirDrag projIII;
   Projectile_NoAirDrag projIV;
   Projectile_NoAirDrag projV;

   public void init() {
      resize (800,620);
      proj1 = new Projectile (40, Math.PI/12);
      proj2 = new Projectile (40, Math.PI/6);
      proj3 = new Projectile (40, Math.PI/4);
      proj4 = new Projectile (40, Math.PI/3);
      proj5 = new Projectile (40, Math.PI);

      projI = new Projectile_NoAirDrag (40, Math.PI/12);
      projII = new Projectile_NoAirDrag (40, Math.PI/6);
      projIII = new Projectile_NoAirDrag (40, Math.PI/4);
      projIV = new Projectile_NoAirDrag (40, Math.PI/3);
      projV = new Projectile_NoAirDrag (40, Math.PI);


      addMouseListener(this);
      t = new Thread(this);
      t.start();

   }
```

```
public void paint (Graphics g) {
   g.setColor(Color.green);
   g.drawOval(2*(int)proj1.x, 2*(int)proj1.y, 20, 20);

   g.setColor(Color.red);
   g.drawOval(2*(int)proj2.x, 2*(int)proj2.y, 20, 20);

   g.setColor(Color.black);
   g.drawOval(2*(int)proj3.x, 2*(int)proj3.y, 20, 20);

   g.setColor(Color.blue);
   g.drawOval(2*(int)proj4.x, 2*(int)proj4.y, 20, 20);
}
public void mouseClicked(MouseEvent e) {
   int x = e.getX();
   int y = e.getY();
}
public void mousePressed(MouseEvent e) {}
public void mouseReleased(MouseEvent e) {}
public void mouseEntered(MouseEvent e) {}
public void mouseExited(MouseEvent e) {}
public void run() {
   try {
      while(true) {
         if (proj1.y <= 75){
            proj1.move();
            repaint();
            t.sleep(timeStep);
         }
         if (proj2.y <= 75){
            proj2.move();
            repaint();
            t.sleep(timeStep);
         }
         if (proj3.y <= 75){
            proj3.move();
            repaint();
            t.sleep(timeStep);
         }
         if (proj4.y <= 75){
            proj4.move();
            repaint();
            t.sleep(timeStep);
```

```
                }
            }
        } catch (InterruptedException e) {}
    }
}

class Projectile {
    double x;
    double y;
    double g;
    double vx;
    double vy;
    int timeStep;
    double a;
    double theta;
    double ax;
    double ay;
    double v;
    static final double cD = 0.4;
    static final double m = 0.057;
    static final double density = 1.2;
    static final double A = 0.0064;

    public Projectile (double v, double theta){
        this.theta = theta;
        this.v = v;
        vx = v * Math.cos(theta);
        vy = -v * Math.sin(theta);
        x = 0;
        y = 75;
        g = 9.8;
        timeStep = 40;
    }
    public void move(){
        v = Math.sqrt(vx*vx+vy*vy);
        theta = Math.atan(vy/vx);
        a = (0.5 * cD * A * v * v * density)/m;
        ax = a * Math.cos(theta);
        ay = a * Math.sin(theta);

        vx = vx - (ax * (timeStep/1000.));
        vy = vy + (g * (timeStep/1000.)) - (ay * (timeStep/1000.));
```

```
        x = x + (vx * (timeStep/1000.));
        y = y + (vy * (timeStep/1000.));

    }
}

class Projectile_NoAirDrag {
   double x;
   double y;
   double g;
   double vx;
   double vy;
   int timeStep;
   public Projectile_NoAirDrag (double v, double theta){
        vx = v * Math.cos(theta);
        vy = v * Math.sin(theta);
        x = 0;
        y = 75;
        g = 9.8;
        timeStep = 40;
   }
   public void move(){
       vx = vx;
       vy = vy - (g * (timeStep/1000.));
       x = x + (vx * (timeStep/1000.));
       y = y - (vy * (timeStep/1000.));
   }
}
```