# Beer Pong

# Investigating Scenarios and Strategies

**Maureen Saint Georges Chaumet** 

Scientific Computing May 4, 2007

# Background

Beer Pong is a very popular college drinking game which originated in the early 1950s<sup>1</sup>. It is a basic hand-eye coordination game in which players attempt to make a ball into a set of cups.

The game involves two teams of two players each. Players stand on either side of a table, with a regulation size of 8x2 feet and height of 27.5 inches above the ground:



Twenty 16-ounce cups are then arranged on the table into two equilateral triangles facing the other team. The cups' rims must touch neighboring cups' so as to minimize the empty space in between cups. The cups are then filled to the ridge line, approximately 1 inch from the bottom so as to prevent toppling over. Here is a depiction on the starting arrangement of the cups:

<sup>&</sup>lt;sup>1</sup> "As Young Adults Drink to Win, Marketers Join In" New York Times, October 16, 2005.



As the game progresses and cups are drunk, it becomes more and more difficult to make a cup as they become isolated. A team is thus allowed to "rerack" their opponents' cups twice throughout a single game. Reracking consists of rearranging the cups into a better configuration so as to improve chances of making cups. Here is a depiction of typical reracking configurations:



<sup>&</sup>lt;sup>2</sup> picture courtesy of http://en.wikipedia.org/wiki/Beer\_pong

The game is played with a 40 mm ping-pong ball with a mass of 2.7 g. Players attempt to throw the ball into their opponents' cups. There are several different types of shots:



The arc shot is the most typical throw. The Fastball throw is an aggressive shot designed to topple an opponent's back cups off the table. The bounce shot consists of a single bounce on the table. A player will only throw a bounce shot if he/she believes the other team is not paying attention to their cups, as a bounce shot can be intercepted legitimately by the other players. The incentive of the bounce shot is that it is worth two cups instead of one.

The game ends when a team has made all of their opponents' cups.

# Motivation

Although I have only recently started playing Beer Pong, I developed an interest in the strategies involved in the game. Although players tend to universally shoot the same shots and take the same reracks, I started to question some of these universal strategies. As I played more frequently and got better at throwing the ball where I aimed, I started to realize that I had more and more trouble with some of the reracks that people claimed were the "easiest." The problem was that I would aim in a straight line at a 3-cup rack, for example, and find that the ball kept bouncing off the rim of the

<sup>&</sup>lt;sup>3</sup> picture courtesy of http://en.wikipedia.org/wiki/Beer\_pong

middle triangle in between the cups. However, I had much better luck when the cups were arranged in a 2-cup rack. I thus started to think that while one might intuitively think the 3-cup rack is the best rack for 3 cups, it is possible that arranging the cups in a straight line might actually be an easier rack. I thus decided to investigate, through this project, the possibility of better strategies than the established ones.

# **Project** Aim

The goals of this project are thus to

- 1. model the projectile and bouncing motion of the ping-pong ball
- 2. explore probabilities of advantageous shots and racks

Hypotheses:

- 1. No matter the advantage of the two-cup rule, an arc shot is a better strategy than a bounce shot
- 2. Linear racks are better than triangular racks

#### Reasoning:

- 1. An arc shot probably has a much better chance of making the ball into a cup, since, in my experience, it seems easier to bounce off the side of the cup when taking a bounce shot.
- 2. Triangular racks introduce more "dead" space (space in between cups). This probably increases the chance of the ball bouncing off the rims. Also I believe that if a player's arm deviates from the ideal vertical position as he/she throws the ball, it will have less effect on the endpoint than if his/her arms had deviated in the horizontal position. Thus I believe there is less chance of missing the cups when aiming for a vertical rack than when trying to adjust horizontally for a triangular rack.

# Modeling

# **Projectile Motion**

The trajectory of a projectile depends on several factors: g, the gravitational constant on Earth;  $\theta$ , the angle at which the projectile is thrown; v, the velocity at which the projectile is thrown, and air resistance dependent on  $\eta$ , the viscosity of air.

Let us define the y direction as vertical motion and the x direction as horizontal direction.

If gravity is the only force acting on the ping-pong ball, then the velocity component equations are

$$v_x = v_{ox}$$
 and  $v_y = v_{oy} - gt$ ,

the corresponding position equations are

$$\mathbf{x} = \mathbf{x}_{o} + \mathbf{v}_{ox}\mathbf{t}$$
 and  $\mathbf{y} = \mathbf{y}_{o} + \mathbf{v}_{oy}\mathbf{t} - \mathbf{r}_{oy}\mathbf{t}^{2}$ ,

and the time component can be described by

$$t = x / (v_o \cos(\theta)).$$

Using these equations, we can describe the horizontal position of the pingpong ball without the time component:

$$y = y_0 + x \tan \theta - \frac{gx^2}{2(v\cos\theta)^2}$$

From this equation, we can thus derive<sup>4</sup> the equation for the angle necessary to reach a target coordinate (x, y):

Let 
$$y_o = 0$$
.  
 $y = x \tan \theta - \frac{gx^2}{2v^2 \cos^2 \theta}$   
 $y = x \tan \theta - \frac{gx^2 \sec^2 \theta}{2v^2}$  [Trigonometric Identity]  
 $y = x \tan \theta - \frac{gx^2}{2v^2} (1 + \tan^2 \theta)$  [Trigonometric Identity]  
 $0 = \frac{-gx^2}{2v^2} \tan^2 \theta + x \tan \theta - \frac{gx^2}{2v^2} - y$ 

Let  $p = tan(\theta)$ .

<sup>&</sup>lt;sup>4</sup> http://en.wikipedia.org/wiki/Trajectory\_of\_a\_projectile

$$0 = \frac{-gx^2}{2v^2}p^2 + xp - \frac{gx^2}{2v^2} - y$$
[Substitution]  

$$p = \frac{-x \pm \sqrt{x^2 - 4(\frac{-gx^2}{2v^2})(\frac{-gx^2}{2v^2} - y)}}{2(\frac{-gx^2}{2v^2})}$$

$$p = \frac{v^2 \pm \sqrt{v^4 - g(gx^2 + 2yv^2)}}{gx}$$

$$\tan \theta = \frac{v^2 \pm \sqrt{v^4 - g(gx^2 + 2yv^2)}}{gx}$$

Finally, here is the equation for the initial angle necessary to reach a target at coordinates (x, y):

$$\theta = \tan^{-1}\left(\frac{v^2 \pm \sqrt{v^4 - g(gx^2 + 2yv^2)}}{gx}\right)$$

We can substitute in  $x = r \cos(\phi)$  and  $y = r \sin(\phi)$  to get the equation for the angle necessary to reach a target at distance r and angle of elevation  $\phi$ :

$$\theta = \tan^{-1} \left( \frac{v^2 \pm \sqrt{v^4 - g(gr^2 \cos^2 \phi + 2v^2 r \sin \phi)}}{gr \cos \phi} \right)$$

I have set the target point at coordinates (2, -0.5) as I am defining the launch point, i.e. the point at which the player lets go of the ball, as (0, 0). Thus, a cup would roughly be found ½ meter in the negative y direction and roughly 2 meters in the positive x direction.

Here is the Mathematica code defining the functions theta[v\_] and theta2[v\_] which correspond to the negative and positive roots for  $\theta$ , the initial launch angle necessary to reach the target at coordinates (2, -0.5) (in meters), as a function of v, the initial launch velocity:

```
g = 9.8;
(* Target point: *)
xTarget = 2;
yTarget = -0.5;
(* Negative Root: *)
theta[v_] := ArcTan[(v^2 - \sqrt{v^4 - g(g * xTarget^2 + 2 * yTarget * v^2)}) / (g * xTarget)];
(* Positive Root: *)
theta2[v_] := ArcTan[(v^2 + \sqrt{v^4 - g(g * xTarget^2 + 2 * yTarget * v^2)}) / (g * xTarget)];
```

In order to plot these equations for a range of initial velocities, we must first calculate the minimum initial velocity necessary to reach the target at coordinates (2, -0.5) or the functions will break.

Here is the Mathematica code calculating the minimum initial velocity necessary to reach the target:

```
(* Minimum velocity to reach target: *)
vNaught = vv /. (Solve[vv^4 - g^2 * (xTarget) ^2 - 2 * g * yTarget * vv^2 == 0, vv][[4]])
```

For this particular target at coordinates (2, -0.5), the minimum initial velocity necessary is 3.91193 m/s.

Given an initial velocity, greater than the minimum initial velocity vNaught, we can now calculate the possible launch angles to reach the target at coordinates (2, -0.5).

Here is the Mathematica code calculating the two launch angles for an initial velocity:

```
(* Initial velocity: *)
v = vNaught + 0.5;
(* Launch angles (in degrees) associated with initial velocity: *)
theta[v] * 180 / Pi
theta2[v] * 180 / Pi
```

The resulting angles, in degrees, are 16.6071° and 59.3567°. The first angle corresponds to the fastball shot while the second angle corresponds to the arc shot. A ping-pong ball thrown at initial velocity v with either of these launch angles would reach the cup situated at coordinates (2, -0.5). The values for the initial velocity and launch angles for both shots seem to make sense in the physical world.

Here is the Mathematica code plotting these angles for the range of initial velocities [vNaught, 15 m/s]:

```
(* Plots of launch angles vs. initial velocity: *)
Plot[{theta[v] * 180 / Pi, theta2[v] * 180 / Pi}, {v, vNaught, 15}];
```

Here is the plot of launch angle vs. initial velocity to reach the target at coordinates (2, -0.5):



As you can see, in the range of velocities feasible for a beer pong player to throw a ping-pong ball (  $\approx$  4 m/s – 6 m/s ), the launch angles seem to correspond to real life data (as ascertained from observation).

In order to make this more life-like, the trajectory of the ball must take into account air resistance on top of gravity.

Here is the Mathematica code which defines the elements involved in air resistance:

```
(* Air resistance: *)
r = 0.02;
n = 1.78 * 10<sup>^</sup>(-5);
k = 6 * Pi * n * r;
m = 0.0027;
```

r (m) is the radius of the ping-pong ball η (kg/(m\*s)) is the viscosity of air k (kg/s) is the drag constant m (kg) is the mass of the ping-pong ball

Here is the new definition of velocity, taking into account air resistance, as a function of the initial velocity and the horizontal distance traveled:

 $Vel[v_, x_] = v * e^{(-k/m * (x/v))};$ 

The next step is now to plot the trajectories of the ping pong ball for both the fastball shot and arc shot, using theta with this new velocity. Here is the Mathematica code defining the trajectories of the ball:

```
(* Trajectories for launch angles: *)
Traj[x_] :=
   Tan[theta[Vel[v, 0]]] * x - (g * x^2 / (2 * (Vel[v, 0] * Cos[theta[Vel[v, 0]]])^2));
Traj2[x_] := Tan[theta2[Vel[v, 0]]] * x -
        (g * x^2 / (2 * (Vel[v, 0] * Cos[theta2[Vel[v, 0]]])^2));
```

#### Here is the Mathematica code plotting these trajectories:

```
 (* Plots of trajectories for launch angles: *) \\ Plot[Traj[x], {x, 0, xTarget}, AxesOrigin \rightarrow {0, 0}, AspectRatio \rightarrow Automatic, \\ PlotRange \rightarrow \{ \{0, 2\}, \{-0.5, 0.1\} \}, PlotStyle \rightarrow \{Thickness[0.015] \} ] \\ Plot[Traj2[x], {x, 0, xTarget}, AxesOrigin \rightarrow \{0, 0\}, AspectRatio \rightarrow Automatic, \\ PlotRange \rightarrow \{ \{0, 2\}, \{-0.5, 1.1\} \}, PlotStyle \rightarrow \{Thickness[0.015] \} ] \\ \end{cases}
```

Here are the plotted trajectories of the ball for both launch angles to reach the target at coordinates (2, -0.5):

Negative root launch angle: 0 y -0.2 fastball shot -0.4 -0.6 2 X 0.25 0.5 0.75 1.25 1.5 1.75 1 Positive root launch angle: 1 0.75 0.5 arc shot 0.25 **y** 0 -0.25 -0.5 -0.75 Х 0.25 0.5 0.75 1.25 1.5 1.75 2 1

#### Here is the entire Mathematica function code for the fastball and arc shot:

```
NoBounce[xTarget_, yTarget_] :=
 Module [{g, restCoef, r, n, k, m, Vel, vNaughtF, vF, theta, theta2, TrajD1, TrajD2,},
  g = 9.8;
  (* Air resistance: *)
  r = 0.02;
  n = 1.78 \times 10^{(-5)};
  k = 6 * P1 * n * r;
  m = 0.0027;
  Vel[v, x] = v * e^{(-k/m * (x/v))};
  (* Minimum velocity to reach target: *)
  vNaughtF = vv /. (Solve[vv^4 - g^2 * (xTarget)^2 - 2 * g * yTarget * vv^2 == 0, vv][[4]]);
  (* Initial velocity: *)
  vF = vNaughtF + 1;
  (* First Bounce: Launch angles *)
  theta[v ] :=
   \operatorname{ArcTan}\left[\left(vF^{2} - \sqrt{vF^{4} - g\left(g * xTarget^{2} + 2 * yTarget * vF^{2}\right)}\right) / (g * xTarget)\right];
  theta2[v] := ArcTan \left[ \left( vF^2 + \sqrt{vF^4} - g \left( g * xTarget^2 + 2 * yTarget * vF^2 \right) \right) \right]
      (g * xTarget)];
  (* First Bounce: Trajectories for launch angles *)
  TrajD1[x ] :=
   Tan[theta[Vel[vF, 0]]] * x - (g * x^2 / (2 * (Vel[vF, 0] * Cos[theta[Vel[vF, 0]])^2));
  TrajD2[x ] := Tan[theta2[Vel[vF, 0]]] * x -
     (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta2[Vel[vF, 0]]])<sup>2</sup>));
  (* First Bounce: Plots of trajectories for launch angles *)
  Plot[TrajD1[x], {x, 0, xTarget}, AxesOrigin \rightarrow {0, -0.9}, AspectRatio \rightarrow Automatic,
   PlotRange \rightarrow \{\{0, 2\}, \{-0.8, 0.1\}\}, PlotStyle \rightarrow \{Thickness[0.015]\}];
  Plot[TrajD2[x], {x, 0, xTarget}, AxesOrigin \rightarrow {0, -0.9}, AspectRatio \rightarrow Automatic,
    PlotRange \rightarrow {{0, 2}, {-0.8, 1.1}}, PlotStyle \rightarrow {Thickness[0.015]}];
NoBounce[2, -0.5]
```

I have thus successfully modeled the projectile motion of the ping-ball for both the fastball and arc shot. Given a target point, i.e. cup, I can determine an appropriate initial velocity (close to the minimum initial velocity) and thus calculate the appropriate launch angle to make the cup.

# **Bouncing Motion**

I must now model the bounce shot. This is a very challenging problem as bouncing is a fairly hard concept to model. I first tried to model a bouncing ball by using event locators in Mathematica. However, having failed, I changed my strategy to defining the bouncing motion of a pingpong ball as a Piecewise function.

I implemented this bouncing in a Module called Bounce in Mathematica. In this module, I took the following steps:

I first redefined the minimum initial velocity to be the minimum initial velocity to reach some intermediary target at coordinate (x0, y0) and chose an appropriate initial velocity (greater than this new minimum initial velocity).

Here is the Mathematica code for this new definition:

```
(* Minimum velocity to reach target: *)
vNaughtF = vv /. (Solve[vv<sup>4</sup> - g<sup>2</sup> * (x0)<sup>2</sup> - 2 * g * y0 * vv<sup>2</sup> == 0, vv][[4]]);
(* Initial velocity: *)
vF = vNaughtF + 1;
```

I then defined the launch angle of the first bounce as the launch angle necessary to reach the target at coordinates (x0, y0).

Here is the Mathematica code defining the launch angle functions for the first bounce:

```
(* First Bounce: Launch angles *)
theta[v_] := ArcTan[(vF^2 - \sqrt{vF^4} - g(g \star x0^2 + 2 \star y0 \star vF^2))/(g \star x0)];
theta2[v_] := ArcTan[(vF^2 + \sqrt{vF^4} - g(g \star x0^2 + 2 \star y0 \star vF^2))/(g \star x0)];
```

I then defined the trajectories for the launch angles for the first bounce. Here is the corresponding Mathematica code:

```
(* First Bounce: Trajectories for launch angles *)
TrajD1[x_] :=
Tan[theta[Vel[vF, 0]]] * x - (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta[Vel[vF, 0]]])<sup>2</sup>));
TrajD2[x_] := Tan[theta2[Vel[vF, 0]]] * x -
   (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta2[Vel[vF, 0]]])<sup>2</sup>));
```

I then calculated the velocity after this first bounce. In order to do this, I used the air resistance function for velocity and scaled it by the restitution coefficient for a ping-pong ball: 0.93<sup>5</sup>. This restitution coefficient describes how much energy is lost to impact and friction during the bounce of a ping-

<sup>&</sup>lt;sup>5</sup> Nagurka, Mark and Shuguang Huang. "A Mass-Spring-Damper Model of a Bouncing Ball" *Proceeding of the 2004 American Control Conference*. Boston: 2004.

pong ball. I thus scaled the final velocity of the first bounce by this coefficient so as to take into account the effect of the table on the ball. Here is the Mathematica code for the final velocity after the first bounce/initial velocity of the second bounce:

```
restCoef = 0.93;
(* Velocity after first bounce *)
vS = restCoef * Vel[vF, x0];
```

I then need to know the final angles of the first bounces so as to infer the launch angles of the second bounces. However, I had no way of knowing these angles. Therefore, I simply took two points near the very end of the first bounces and deduced the final angles by using their slope.

Here is the Mathematica code calculating the final angles of the first bounces:

```
(* Second Bounce: Launch angles *)
NegAngle = ArcTan[(TrajD1[x0] - TrajD1[(x0 - 0.003)]) / 0.003];
PosAngle = ArcTan[(TrajD2[x0] - TrajD2[(x0 - 0.003)]) / 0.003];
```

I chose to separate my points by 0.003 as I felt that it was a small enough interval so as to reflect only the end of the curve, but big enough to avoid taking the slope of points too close.

I then used the negatives of these launch angles with the final velocity of the first bounces to calculate the trajectory of the second bounces. Here is the Mathematica code describing the trajectory functions for the second bounces:

```
(* Second Bounce: Trajectories for launch angles *)
TrajD12[x_] :=
Tan[-NegAngle] * (x - x0) - (g * (x - x0) ^2 / (2 * (Vel[vS, x0] * Cos[-NegAngle]) ^2)) + y0;
TrajD22[x_] := Tan[-PosAngle] * (x - x0) -
    (g * (x - x0) ^2 / (2 * (Vel[vS, x0] * Cos[-PosAngle]) ^2)) + y0;
```

I finally calculated the final velocity after the second bounce. Here is the Mathematica code corresponding to this calculation:

```
(* Velocity after second bounce *)
vFinal = restCoef * Vel[vS, xTarget];
```

I now needed to construct the Piecewise functions with these pieces of a bounce. I thus defined the function from  $0 \le x \le x0$  as the first bounce and from  $x0 < x \le x$ Target as the second bounce. Here is the Mathematica code for these Piecewise constructions:

```
(* Piecewise for trajectories for launch angles *)
TrajD = Piecewise[{{TrajD1[x], x \ge 0 && x \le x0}, {TrajD12[x], x > x0 && x \le xTarget}];
TrajD3 = Piecewise[{{TrajD2[x], x \ge 0 && x \le x0}, {TrajD22[x], x > x0 && x \le xTarget}];
```

I then plotted the Piecewise functions:

```
(* Plot piecewise *)
Plot[TrajD, {x, 0.01, xTarget}, AxesOrigin → {0, y0},
AspectRatio → Automatic, PlotStyle → {Thickness[0.015]}];
Plot[TrajD3, {x, 0.01, xTarget}, AxesOrigin → {0, y0}, AspectRatio → Automatic,
PlotStyle → {Thickness[0.015]}];
```

Finally, the module returned the y position of the ball when it reaches xTarget (for the positive root launch angle):

```
TrajD22[xTarget]
```

Using this module with variables for x0, y0, xTarget and yTarget, I was able to set:

y0 to be the table height (  $\approx$  -0.8 m)

xTarget to be the distance to a cup (  $\approx$  2 m)

yTarget to be the height of the rim of a cup (  $\approx$  -0.5 m)

I then used FindRoot to find the value for x0 for which Bounce returned the correct yTarget. In this case, the x value at which the player should bounce the ball is  $\{x0 \rightarrow 1.40191\}$  m.

Using this value, I plotted the path of the ball if the player used a fastball bounce or arc bounce and aimed at x0=1.40191:

Negative root launch angle:



Positive root launch angle:



#### Here is the entire Mathematica function code for the bounce shots:

```
Bounce[x0_, y0_, xTarget_, yTarget_] :=
 Module [{g, restCoef, r, n, k, m, Vel, vNaughtF, vF, theta, theta2, TrajD1,
   TrajD2, vS, NegAngle, PosAngle, TrajD12, TrajD22, vFinal, TrajD, TrajD3},
  q = 9.8;
  restCoef = 0.93;
  (* Air resistance: *)
  r = 0.02;
  n = 1.78 \times 10^{(-5)};
  k = 6 * Pi * n * r;
  m = 0.0027;
  Vel[v_, x_] = v * e^{(-k/m*(x/v))};
  (* Minimum velocity to reach target: *)
  vNaughtF = vv /. (Solve[vv^4 - g^2 * (x0)^2 - 2 * g * y0 * vv^2 = 0, vv][[4]]);
  (* Initial velocity: *)
  vF = vNaughtF + 1;
  (* First Bounce: Launch angles *)
  theta[v_] := ArcTan[(vF<sup>2</sup> - \sqrt{vF^4} - g(g * x0^2 + 2 * y0 * vF^2))/(g * x0)];
  theta2[v_] := ArcTan \left[ \left( vF^2 + \sqrt{vF^4 - g(g * x0^2 + 2 * y0 * vF^2)} \right) / (g * x0) \right];
  (* First Bounce: Trajectories for launch angles *)
  TrajD1[x ] :=
   Tan[theta[Vel[vF, 0]]] * x - (g * x^2 / (2 * (Vel[vF, 0] * Cos[theta[Vel[vF, 0]])^2));
  Tra_{D2}[x] := Tan[theta_2[Vel[vF, 0]]] * x -
    (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta2[Vel[vF, 0]]])<sup>2</sup>));
```

```
Bounce[x0 , y0 , xTarget , yTarget ] :=
 Module [{g, restCoef, r, n, k, m, Vel, vNaughtF, vF, theta, theta2, TrajD1,
   TrajD2, vS, NegAngle, PosAngle, TrajD12, TrajD22, vFinal, TrajD, TrajD3},
  g = 9.8;
  restCoef = 0.93;
  (* Air resistance: *)
  r = 0.02;
  n = 1.78 \pm 10^{(-5)};
  k = 6 * P1 * n * r;
  m = 0.0027;
  Vel[v_, x_] = v * e^{(-k/m*(x/v))};
  (* Minimum velocity to reach target: *)
  vNaughtF = vv /. (Solve[vv<sup>4</sup> - g<sup>2</sup> * (x0)<sup>2</sup> - 2 * g * y0 * vv<sup>2</sup> == 0, vv][[4]]);
  (* Initial velocity: *)
  vF = vNaughtF + 1;
  (* First Bounce: Launch angles *)
  theta[v_] := ArcTan[(vF^2 - \sqrt{vF^4} - g(g \star x0^2 + 2 \star y0 \star vF^2))/(g \star x0)];
  theta2[v] := ArcTan \left[ \left( vF^2 + \sqrt{vF^4 - g(g * x0^2 + 2 * y0 * vF^2)} \right) / (g * x0) \right];
  (* First Bounce: Trajectories for launch angles *)
  TrajD1[x ] :=
   Tan[theta[Vel[vF, 0]]] * x - (g * x^2 / (2 * (Vel[vF, 0] * Cos[theta[Vel[vF, 0]]))^2));
  TrajD2[x_] := Tan[theta2[Vel[vF, 0]]] * x -
     (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta2[Vel[vF, 0]]])<sup>2</sup>));
   (* Second Bounce: Plots of trajectories for launch angles *)
   (* Plot[TrajD12[x], {x,x0,xTarget},
     AxesOrigin \rightarrow \{0, 0\}, PlotStyle \rightarrow \{Thickness[0.015]\};
    Plot[TrajD22[x], \{x, x0, xTarget\}, AxesOrigin\rightarrow \{0, 0\},
     PlotStyle→{Thickness[0.015]}]; *)
   (* Velocity after second bounce *)
   vFinal = restCoef * Vel[vS, xTarget];
   (* Piecewise for trajectories for launch angles *)
   TrajD = Piecewise[{{TrajD1[x], x \ge 0 \&\& x \le x0}, {TrajD12[x], x > x0 \&\& x \le xTarget}}];
   TrajD3 = Piecewise[\{\{TrajD2[x], x \ge 0 \& \& x \le x0\}, \{TrajD22[x], x > x0 \& \& x \le xTarget\}\}];
   (* Plot piecewise *)
   Plot[TrajD, {x, 0.01, xTarget}, AxesOrigin \rightarrow {0, y0},
    AspectRatio \rightarrow Automatic, PlotStyle \rightarrow {Thickness[0.015]}];
   Plot[TrajD3, {x, 0.01, xTarget}, AxesOrigin \rightarrow {0, y0}, AspectRatio \rightarrow Automatic,
    PlotStyle \rightarrow {Thickness[0.015]}; TrajD22[xTarget]
```

```
x00 = 0.5;
xCoor = 2;
yCoor = -0.5;
TableHeight = -0.8;
solution = FindRoot[Bounce[x0, TableHeight, xCoor, yCoor] == yCoor, {x0, x00}]
Bounce[x0 /. solution, TableHeight, xCoor, yCoor]
```

I have thus successfully modeled the bouncing motion of the pingpong ball for both the negative and positive root launch angles (i.e. the "fastball" and "arc" versions of the bounce shot). Given a target point, i.e. cup, I can determine an appropriate initial velocity (close to the minimum initial velocity) and thus calculate the appropriate launch angle and x position to make the cup on a bounce.

# **Strategy**

## **Rack exploration**

I wanted to explore how movement in the horizontal and vertical position affected the resulting target reached.

I thus looked at how the target changed as I varied the launch angle. I decided to change the initial launch angle by 10% for the fastball shot. Here is the Mathematica code reflecting this alteration:

```
TrajD1[x_] :=
Tan[theta[Vel[vF, 0]]] * x - (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta[Vel[vF, 0]])<sup>2</sup>));
TrajDlerror[x_] := Tan[theta[Vel[vF, 0]] * 1.1] * x -
   (g * x<sup>2</sup> / (2 * (Vel[vF, 0] * Cos[theta[Vel[vF, 0]] * 1.1])<sup>2</sup>));
```

Since  $\frac{\text{theta}[Ve1[vF, 0]]}{\text{measures}}$  measures 0.166048 radians, increasing it by 10% would result in a variation of 0.0166048 radians.

```
Print[TrajD1[xTarget]]; \Rightarrow -0.5
Print[TrajD1error[xTarget]]; \Rightarrow -0.470659
```

Thus, a change of 0.0166048 radians causes a change in the vertical coordinate of the target of approximately 0.03 m.

Let us now examine how a comparable change in the horizontal direction would affect the target reached. Imagine that the horizontal coordinate of the target changed by 0.03 m. This creates a triangle with y = 0.03 m, x = 2 m (target x). The angle of this triangle would be  $\tan^{-1}(y/x) = \tan^{-1}(0.03/2) = 0.0150000$ . Thus a 0.03 m change in the horizontal direction would require an angle 94% as large as the angle required for a 0.03 m change in the vertical direction. Thus, for fastball shots, this suggests that linear racks are more advantageous than triangular racks (in addition to the advantage of less "dead" space created by triangles). This result confirms my hypothesis that linear racks are the most advantageous racks.

## Shot exploration

In order to explore different strategies in shooting, I compared a 10% change in initial launch angle for a fastball shot and a "fastball" bounce. While the fastball shot caused a 6% (0.03/0.5) change in the vertical position, the "fastball" bounce only caused a 0.76% (0.0035/0.466) change in the vertical position. I did not expect this as I had thought that bouncing would compound the error so as to make the change more noticeable. However, it seems as though bouncing has an effect important enough to counteract an increase in initial launch angle.

An even more interesting result, however, is that, while I did not explicitly state this hypothesis, I truly believed that arc shots were more advantageous than fastball shots. From experience, it seems as though players who use fastball shots often miss the cups by bouncing off the rims. However, while increasing the angle of an arc shot by only 10%, I got the following result:



# Conclusion

Conclusively, it seems as though my instincts about shots were completely off. While the bounce shot can easily be intercepted, it seems as though it is the best shot because, in addition to absorbing player movement very effectively, the bounce shot is worth two cups.

Fastball shots are very close contenders as they vary very little with vertical and horizontal player movement. Thus it seems as though, in terms of player movement, the arc shot is the worst shot. However, I still believe, though I was not able to explore this aspect of shots, that the arc shot offers a wider "field" of cups as it allows for a more direct approach to the cups.

Thus, while the other shots arrive at the cups at an angle, the near perpendicular approach of arc shots make it much less likely to hit the rims or sides of the cups.

All in all, though, it seems that the only hypothesis that turned out to be provably true is the fact that linear racks are more advantageous than triangular racks. This is especially true since I have discovered the effect vertical movement has on arc shots, because, while the vertical position does change dramatically, as you can see from the previous graph, the horizontal position did not change as drastically. What is interesting about this result is that triangular racks are the most sought-after racks in Beer Pong. It would thus be interesting to see if changing rack strategy would affect the outcome of games. I am very eager to try out some of these new theories.