The Physics of a Three Point Shot

Team 379: Problem B

Abstract–Accurately predicting the trajectory of a spinning basketball through air and as it interacts with a rim and a backboard, poses several interesting problems. Solving these problems requires an understanding of the effects of lift and drag on the ball as well as the outcomes of collisions between the basketball and the backboard and the rim. We have developed a simulation to predict the trajectory of a basketball as it navigates these obstacles and explored the set of velocities and spins that a basketball can have as it is thrown from the three-point line. Our results indicate that backspin on a basketball allows shots to be made with smaller initial velocities; these results also indicate that it is more favorable to bank the ball off of the backboard into the net.

The Physics of a Three-Point Shot

Team 379

I. Introduction

A great American pastime, basketball raises many interesting questions for the curious physicist. By studying the physics behind a basketball shot, physicists can develop insight into the shooting conditions most likely to result in a successful shot, and therefore may help basketball players focus their training on the parameters most likely to imporve their performance. In this paper, we focus our study on shots taken from the three-point line, and investigate what combinations of spin and initial velocity result in the ball entering the net.

Determining the set of velocities and spins necessary to successfully shoot a basket requires a discussion of projectile motion as well as the kinematics of colliding bodies. In order to investigate this solution space we developed a numerical simulation to model the trajectory of a basketball as it is shot at a basket.

II. The Scenario

Suppose a player in an Olympic basketball game is standing 6.2m from the basket and 45° from the principle axes of the court. We are interested in finding the possible velocities and spins that will allow the player to score a basket.

Prior to developing our model we must define or coordinate system. We have chosen to center our coordinate system at the position of the player, and we have defined the x,y and z axes as depicted in Figure 1. Occasionally it has proven uesful to consider a spherical coordinate system instead of a cartesian coordinate system. In this case we have defined the polar angle (θ) relative to the x-axis and the angle of elevation (ϕ) relative to the floor of the court. Also of interest is the goemetry of the backboard and the rim; these are depicted in Figure 2.

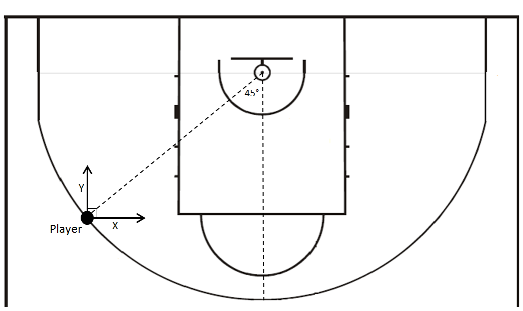

Fig. 1. The position of the player on the court relative to the rim (Figure modified from the International Basketball Federation's Official Basketball Rules 2010 [2])

Similarly, Appendix A contains various parameters obtained from the International Basketball Federation, FIBA, which governs the rules of international basketball including the Olympics [3].

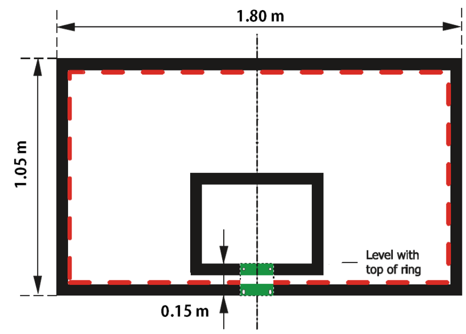

Fig. 2. The dimensions of the backboard and relative dimensions of the rim (Figure modified from FIBA's Official Basketball Rules 2010: Basketball Equipment [3])

III. Model

Our model simulates the trajectory of a spinning basketball from the moment it leaves a player's hands until it falls below the height of the basket, at which point the simulation determines whether or not the shot was successful. The simulation consists of two regions: projectile motion through the air and collisions with the backboard and rim.

A. Path through air

1) Theory and Assumptions: A spinning ball in the air is subject to three forces: gravity, drag and the Magnus force. Therefore, the motion of the ball is governed by the following equation:

$$\frac{d^2\vec{r}}{dt^2} = \vec{g} + \frac{1}{m_{ball}}\left(\frac{16}{3}\pi^2 r_{ball}^3 \rho\vec{\omega}\times\vec{v} - .5C_d\rho A|\vec{v}|^2\hat{v}\right) \quad (1)$$

where $\vec{r}$, $\vec{v}$, and $\vec{\omega}$ are the position, velocity, and angular velocity of the basketball, $\vec{g}$ is the acceleration due to gravity, m_{ball}, r_{ball} and A are the mass, radius, and cross-sectional area of the basketball, ρ is the density of air and C_d is the drag coefficient of the basketball in the air which we found in literature to be 0.54 [1] [6]. The first term in Equation 1 accounts for gravity, the second term accounts for the Magnus force due to the ball's spin [4] and the third term accounts for the drag force [7]. For simplicity, the basketball was assumed to be spinning with a constant angular velocity throughout its trajectory. It was also assumed that the spin was purely backspin, meaning that the axis of rotation lies in the xy-plane and is perpendicular to the ball's trajectory.

2) Implementation: Equation 1 was solved numerically using a 4^{th} order Runge-Kutta differential equation solver in MATLAB. The trajectory through the air was calculated until the ball either collided with the backboard or rim (see below) or fell below the height of the hoop.

B. Collisions

1) Theory and Assumptions: Collisions between a basketball and the backboard or the rim are affected by the material properties of the basketball, the backboard and the rim as well as the motion of all of the components involved. The rim and the backboard can be assumed to be at rest before the collision, any small vibrations that these exhibit before and during the collision will not significantly affect the ball's trajectory. Considering frictionless collisions simplifies the situation by eliminating the effect of the ball's spin on the outcome. This simplifying assumption may have a measurable effect on the outcome of the simulation. During the collision between the ball and the backboard or the rim, both bodies will deform; resistive forces during this deformation will dissipate kinetic energy making the collision inelastic. In a perfectly elastic collision the impulse would be equal to twice the momentum of the ball in the direction perpendicular to the plane of collision. This change in momentum is given by:

$$\Delta\vec{p} = 2\frac{\vec{p}\cdot\vec{n}}{||n||^2}\vec{n}$$

where $\vec{n}$ is a vector normal to the plane of collision. Given the simplifying assumptions and an inelastic collision, however, the impulse is given by:

$$\Delta\vec{p} = (1+\epsilon)\frac{\vec{p}\cdot\vec{n}}{||n||^2}\vec{n}$$

where ϵ is the coefficient of restitution of the basketball ($0 \leq \epsilon \leq 1$). This coeffient of resitution is determined by the material properties of the ball as well as the backboard. Olympic regulations mandate that the coefficient of restitution between the backboard and the basketball (ϵ_b) must be greater than .5; the same regulations mandate that the coefficient of restitution between the basketball and the rim (ϵ_r) must be between .65 and .5 [3]. Because the collision is assumed to be frictionless, the momentum parallel to the collision plane is conserved.

2) Implementation: In order to incorporate collisions into the simulation we subdivided the simulation into several smaller simulations. During each of these sub-simulations MATLAB's ordinary differential equation solver was used to predict the path of the basketball until it either reached a collision point or met a terminating condition (i.e. the ball missed the backboard or fell below the rim). Once a collision was detected, the collision plane was determined and the impulse was calculated, the ball's resulting velocity was then used as an initial condition for the next sub-simulation. This process was repeated iteratively until a terminating condition was met, at which time the ball's position could be evaluated to determine if the player had scored. The simulation code, including the math used to identify collisions and determine the collision plane, can be found in Appendix B.

IV. Results

To investigate the possible initial condidtions of the basketball shot, we varied both the initial velocity and the spin of the basketball. We investigated the initial velocity in terms of three components: the speed v_0, the angle of elevation ϕ (measured up from the ground), and the polar angle θ (measured counterclockwise from the x-axis). With $\theta = \frac{\pi}{4}$ and no spin, Figure 3 shows in grey which values of v_0 and ϕ result in a successful shot. These results are similar to those obtained by Silverberg, Tran and Adcock [5] in their simulations of basketball free-throws.

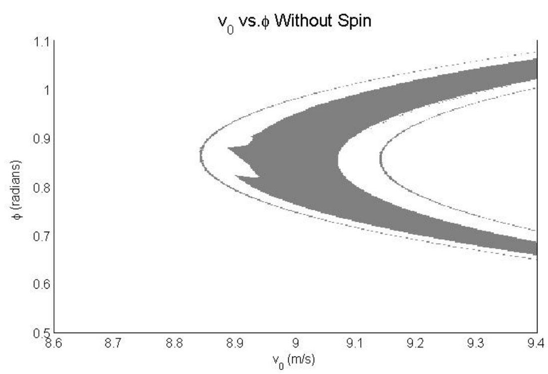

Fig. 3. Speeds and angles of elevation resulting in successful shots when $\theta = \frac{\pi}{4}$ and the ball has no spin.

In Figure 3, the largest continuous area of grey represents all the the shots during which the basketball either enters the net directly without any collisions or bounces off the inside of the rim before entering the net. The thinner parabolas outside of the thicker area represent the shots in which the ball hits either the front or back edge of the rim, bounces upwards, and then falls back down into the net. The irregularities near the vertex of the thick parabola are due to shots in which the ball just barely hits the top of the hoop and then bounces away instead of bouncing in. It is worth noting that for shots with $\theta = \frac{\pi}{4}$ there are no successful cases in which the ball hits the backboard. This is because a collision with the backboard would occur directly above the hoop, causing the ball to bounce off to the side, therefore missing the shot.

Assuming that a player were aiming his shot directly towards the net, Figure 3 shows that the player should attempt to shoot with an initial speed of approximately 9 m/s at an angle of elevation of .85 radians. This shot would provide the greatest chance of success because there is a large region around that point within which the shot would still be successful. In Figure 3, it can also be seen that the top half of the parabola is thicker than the bottom half. This means that the range of small angles for which the basketball enters the net is smaller than the range of large angles. This result indicates that throws shot with a larger angle of elevation are more likely to be successful than those shot with a smaller angle of elevation, because there is a greater tolerance for error.

A similar situation is seen when the basketball is given backspin. Figure 4 shows the possible successful shots when $\theta = \frac{\pi}{4}$ and the ball has a backspin of two revolutions per second. This graph shows the same trends as Figure 3, but with the parabolas shifted towards lower velocities and angles of elevation. This effect is because the backspin of the ball creates a lift force that opposses gravity, thereby allowing shots that would otherwise be too low to be successful. Therefore, by introducing backspin to a shot, basketball players can shoot with lower initial speeds.

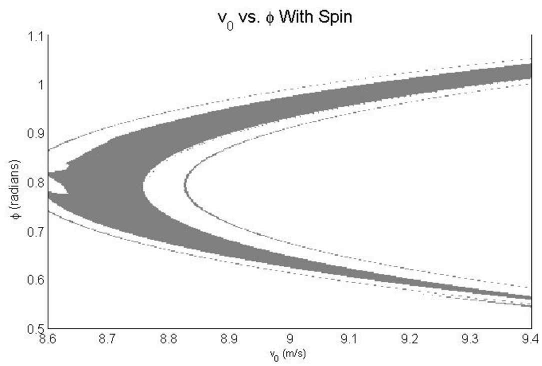

Fig. 4. Speeds and angles of elevation resulting in successful shots when $\theta = \frac{\pi}{4}$ and the ball has a backspin of two revolutions per second.

Often basketball players use the backboard to place the ball in the basket. We investigated the possible angles, the angle of elevation of the shot as well as the polar angle of the shot. Figure 5 provides a depiction of the angles that will allow a ball to enter the basket.

The two nearly symmetric regions on the left of Figure 5 are centered around an angle of $\frac{\pi}{4}$, indicating that these are shots that do not collide with the backboard. As the polar angle of the shot increases there is a region for which no shots will enter the basket (for the particular speed chosen). These angles correspond to shots that may or may not collide with the rim or the backboard, but nonetheless do not enter the basket. At a polar angle of approximately .825 radians, a large region of successful shots is found. These are "bankshots" that the player has bounced off of the backboard and into the basket.

It is interesting to note that this region is much larger than either of regions that did not involve collisions with the backboard. This observation accounts for the reason that most successful shots employ the use of the backboard.

To further investigate the bankshots, we searched for successful shots of various elevation angles and speeds at a fixed polar angle of .85 radians. The results of this simulation can be seen in Figure 6. As one might expect, these results are similar to those at a polar angle of $\frac{\pi}{4}$, seen in Figure 3, with a shifted origin. There is a remarkable difference between the vertices of the parabolas, however. The bankshots do not exhibit the chaotic behavior that the other shots do. Shots that are aimed directly at the basket have the possibility of skipping off of the rim, while shots aimed at the backboard to not have this danger.

These results indicate that it is both safer and more likely that a basketball will enter the basket if it is shot at the

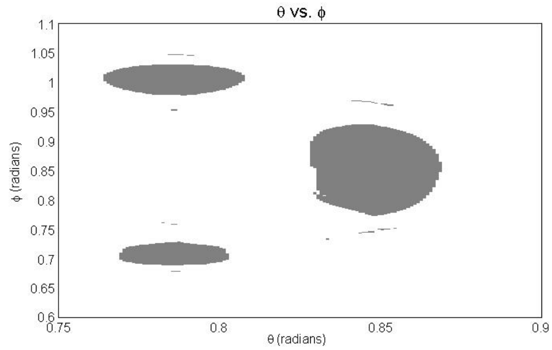

Fig. 5. The range of angles that would allow a shot of 9.25m/s to enter the basket. This figure shows a region of possible bank shots (right), as well as two regions that will not collide with the backboard

backboard.

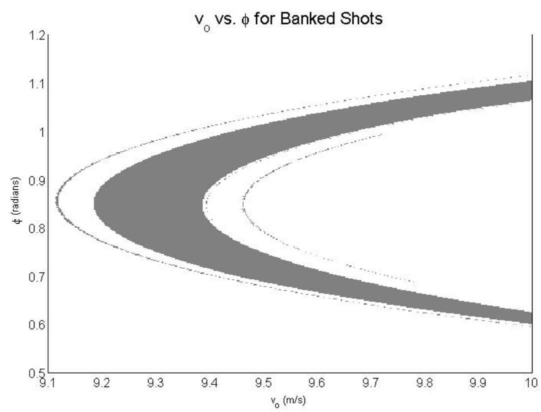

Fig. 6. The spread of speeds and elevation angles that can result in a successful shot at a polar angle of .85 radians. Similar to Figure **??**, but does not exhibit chaotic behavior at the vertex of the parabola.

V. Discussion

The differential equations that describe the motion of a spinning basketball through a fluid are non-linear, and as a result no closed-form solutions can be found. Numerical methods that approximate solutions can be powerful tools in analyzing solutions. One of the principle strengths of our approach is the ability to approximate these "impossible" solutions. We have found that putting backspin on a ball allows players to shoot with lower initial speeds. In addition, when standing at an angle from the hoop, shots off the backboard are both easier and more likely to be successful.

Several simplifying assumptions were made in order to maintain the speed and efficiency of simulation. Some of these assumptions were more valid than others. Vibrations and deformations in the backboard and rim, for example, were ignored because they would consume lengthy processing times with very little effect on the result. The friction during collisions, however, probably greatly affects the outcome of a shot that bounces off of the backboard or the rim; ignoring these effects sacrifices the accuracy of the simulation, while improving processing efficiency. Other assumptions were made for ease of visualization. It is difficult to visualize solution spaces that exist in more than two dimensions; unfortunately the solution space for this problem is six dimensional - three dimensions for linear velocity and an additional three for angular velocity (spin). As a result we eliminated two dimensions by considering purely backspin on the ball; this is the way most basketball players shoot.

Acknowledgment

The authors would like to thank our advisor Yevgeniya Zastavker.

References

[1] Hiroki Okubo, Mont Hubbard, *Identification of basketball parameters for a simulation model*, Procedia Engineering, Volume 2, Issue 2, June 2010, Pages 3281-3286.http://www.sciencedirect.com/science/article/pii/S1877705810003991

[2] Official Basketball Rules 2010, *International Basketball Federation*, 2010. Puerto Rico. http://www.fiba.com/downloads/Rules/2010/OfficialBasketballRules2010.pdf

[3] Official Basketball Rules 2010: Basketball Equipment, *International Basketball Federation*, 2010. Puerto Rico. http://www.fiba.com/downloads/Rules/2010/BasketballEquipment2010_V2.pdf

[4] Ideal Lift of a Spinning Ball, *National Aeronautics and Space Administration*, 2008. http://www.grc.nasa.gov/WWW/k-12/airplane/beach.html

[5] Silverberg, L., Tran, C. &, Adcock, K. *Numerical Analysis of Basketball Shot*, Journal of Dynamic Systems, Measurement, and Control, Volume 125, December 2003. http://www.mae.ncsu.edu/Numerical\%20Analysis\%20paper.pdf

[6] Ujihashi, Sadayuki. The Impact of Technology on Sport II, CRC Press, 2007.

[7] Wolfson R. & Pasachoff J. M. (1999) Physics for Scientists and Engineers, Second Edn., New York, Pearson.

A Model Parameters

Basketball Properties	
Circumference	0.75 m
Mass	0.600 kg
Rim Properties	
Inside Diameter	0.45 m
Metal Diameter	0.02 m
Height from Floor	3.050 m
Distance from Bottom of the Backboard	0.150 m
Minimum Horizontal Distance to Backboard	0.151 m
Coefficient of Restitution	0.65
Backboard Properties	
Width	1.8 m
Height	1.05 m
Coefficient of Restitution	0.60

Table 1: The various values obtained from the International Basketball Federation's Official Basketball Rules 2010: Basketball Equipment [3]

Initial Height of Basketball	6 m
Drag Coefficient	0.54

Table 2: Parameters we assumed

B Source Code

```
function score = Basketball_noPlot(varargin)
    global omega
    %% Manage Inputs
    if nargin==0
        speed = 8.7;
        phi = .7;
        theta = pi/4;
        omega = [1;-1;0];
    else
        speed = varargin{1};
        phi = varargin{2};
        theta = varargin{3};
        omega = varargin{4};
    end
```

```matlab
%% Define Constants
r_ball = .75/(2*pi);
r_hoop = .225;
h_hoop_top = 3.050;
h_bb = 1.050;
w_bb = 1.800;
pos_hoop = 6.2/sqrt(2)*[1;1;0] + h_hoop_top*[0;0;1];
pos_bb = pos_hoop + [0;r_hoop+.151;h_bb/2-.15];

%% Initial Conditions
Vx = speed*cos(phi)*cos(theta);
Vy = speed*cos(phi)*sin(theta);
Vz = speed*sin(phi);

nextPosition = [0,0,2];%position
nextVelocity = [Vx,Vy,Vz];%velocity
nextTime = 0;

%% Setup Options
function [value,isterminal,direction] = events(¬,Y)
    %escape past rim
    value(1) = Y(3) - h_hoop_top;
    isterminal(1) = 1;
    direction(1) = -1;

    %escape past backboard
    value(2) = Y(2)-pos_bb(2);
    isterminal(2) = ¬inRect(r_ball,w_bb,h_bb,Y([1,3]),pos_bb([1,3]));
    direction(2) = 1;

    %escape in x-dir
    value(3) = Y(1) - 6.2/sqrt(2) - w_bb/2;
    isterminal(3) = 1;
    direction(3) = 0;

    %collide with the backboard
    value(4) = distanceToRect(w_bb,h_bb,Y(1:3),pos_bb)-r_ball;
    isterminal(4) = 1;
    direction(4) = -1;

    %collide with the ring
    value(5) = distanceToRing(r_hoop,Y(1:3),pos_hoop)-r_ball;
    isterminal(5) = 1;
    direction(5) = -1;

    %the ball has fallen below the floor
    %catch any missed cases
    value(6) = Y(3);
    isterminal(6) = 1;
    direction(6) = 0;
end

options = odeset('Events',@events,'RelTol',1e-5);
```

```
    %% Iterate through all bounces
    while(true)
        Y0(1:3) = nextPosition;
        Y0(4:6) = nextVelocity;
        [T,Y,¬,¬,IE] = ode45(@Equations_Of_Motion,[nextTime,nextTime+10],Y0,options);
        if IE(end) == 1 || IE(end) == 2 || IE(end) == 3 || IE(end) == 6
            break
        elseif IE(end) == 4
            %collision with the backboard
            n = [0,-1,0];
            energyAbsorbtion = .5;
        elseif IE(end) == 5
            %collision with the hoop
            n=minDiscplacementFromHoop(r_hoop,Y(end,1:3),pos_hoop);
            n=n/norm(n);
            energyAbsorbtion = .65;
        end
        [p1, p2] = parallelPerp(Y(end,4:6),n);
        nextVelocity = p2-p1*sqrt(1-energyAbsorbtion);
        nextPosition = Y(end,1:3);
        nextTime = T(end);
    end

    %% Determine Score
    score = norm(Y(end,(1:3))-pos_hoop')<r_hoop-r_ball;
end

%% Simulation
function dY = Equations_Of_Motion(¬,Y)
    Vel = Y(4:6);
    Acc = Acceleration(Vel);

    dY = [Vel;Acc];
end

function Acc = Acceleration(Vel)
    global omega
    C_d = .54;
    m_ball = .6;
    r_ball = .75/(2*pi);
    A = pi*r_ball^2;
    rho = 1;

    F_drag = -.5*C_d*rho*A*norm(Vel)*Vel;
    F_magnus = 16/3*pi^2*r_ball^3*rho*cross(omega,Vel);

    a = (F_drag+F_magnus)/m_ball;
    g = [0;0;-9.81];
    Acc = a+g;
end

%% Helper Functions

function [parallel,perp] = parallelPerp(v1,v2)
```

```
    parallel = sum(v1.*v2)/norm(v2)/norm(v2)*v2;
    perp = v1-parallel;
end

function test = inRect(r,w,h,p1,p2)
    p = abs(p1-p2);
    if p(1)≥w/2+r || p(2)≥h/2+r
        test = false;
    elseif (norm(p-[w/2;h/2])≥r) && ((p(1)≥w/2) || (p(2)≥h/2))
        test = false;
    else
        test = true;
    end
end

function d = distanceToRect(w,h,p1,p2)
    %assumes that the Rect is in the x-z plane (as the backboard will be)
    p = abs(p1-p2);
    if p(1)<w/2 %if the center is inside the width
        if p(3)<h/2
            %distance from the point to the x-z plane
            d=p(2);
        else
            %distance from the point to the horizontal edge
            p = p-[0;0;h/2];
            d = norm(p(2:3));
        end
    else
        if p(3)<h/2
            %distance from the point to the vertical edge
            p = p-[w/2;0;0];
            d = norm(p(1:2));
        else
            %distance from the point to the corner
            p = p-[w/2;0;h/2];
            d = norm(p);
        end
    end
end

function d = distanceToRing(r,p1,p2)
    p = p1-p2;
    d = norm([p(3),norm(p(1:2))-r]);
end

function X = minDiscplacementFromHoop(r,p1,p2)
%the hoop is assumed to be in the horizontal plane
    p = p1-p2';
    [¬,proj] = parallelPerp(p,[0,0,1]);
    X = proj-proj/norm(proj)*r + [0,0,p1(3)-p2(3)];
end
```