Mathematical Modeling of Bowling Ball Trajectory: Optimizing Entry Angle Through Variable Analysis

Jacob Berger

December 2024

Abstract

Bowling is a sport that combines physical skill with an understanding of physics and geometry. A critical factor in achieving a strike is the ball's entry angle into the pins. Research indicates that an entry angle of approximately 6 degrees significantly increases the likelihood of a strike [2]. This paper presents a mathematical model to predict the trajectory of a bowling ball, focusing on how key variables—such as ball speed, rev rate, axis of rotation, and oil pattern—collectively influence its path. By making strategic approximations and incorporating these factors into a system of differential equations, we aim to create a tool that helps bowlers determine optimal strategies in real-time. The model is designed to be adaptable, allowing for any oil pattern input based on universally known parameters accessible to bowlers. This approach distinguishes our work from previous complex models by prioritizing usability without sacrificing essential accuracy. Future work includes implementing the model computationally and validating it against empirical data.

1 Introduction

Bowling is not only a test of physical prowess but also an application of physics and geometry. Achieving a strike often depends on the ball's entry angle into the pins, with research suggesting that an entry angle of approximately 6 degrees significantly increases strike probability [2]. This project aims to develop a mathematical model that predicts the trajectory of a bowling ball, focusing on key variables such as ball speed, rev rate, axis of rotation, and oil pattern. By understanding these factors, we seek to provide practical strategies for bowlers to optimize their performance without the need for extensive data collection.

The model was implemented computationally, leveraging differential equations to simulate the bowling ball's trajectory. It integrates essential variables such as the lane's oil pattern and the ball's physical parameters, translating complex motion dynamics into a visual representation of predicted trajectories. Figure 1 illustrates the model and its ability to simulate multiple trajectories under varying conditions.

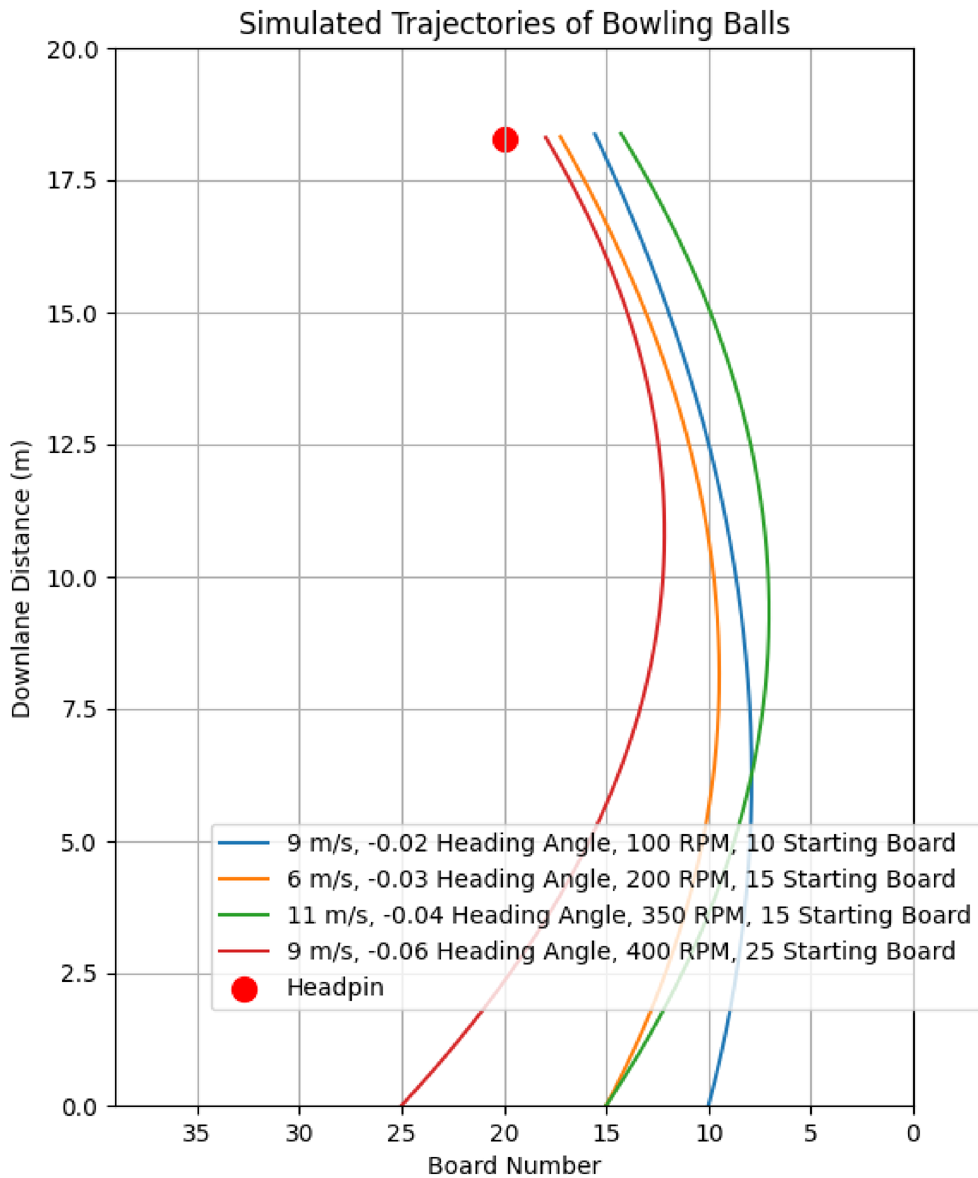

Figure 1: Simulation output showing bowling ball trajectories under different initial conditions. The model incorporates friction dynamics, oil pattern variations, and physical ball properties to predict downlane positioning and entry angles.

This simulation-based approach not only provides insights into the ball's motion but also bridges the gap between theoretical physics and practical application in bowling. The following sections describe the methodology used to construct this model, including the equations governing its motion, the approximations made, and the role of the oil pattern in determining the trajectory.

1.1 Background

Previous studies have explored the physics behind bowling ball motion. Frohlich [1] investigated the factors that cause a bowling ball to hook, highlighting the role of friction and the ball's rotational dynamics. The United States Bowling Congress (USBC) conducted comprehensive research identifying critical factors like the coefficient of friction (COF), radius of gyration (RG), and differential RG that influence ball motion on the lane [3]. These studies often involve complex models and require sophisticated equipment for data collection.

While these contributions have advanced the understanding of bowling physics, they may not be readily applicable for individual bowlers seeking to improve their game. The existing models can be too intricate for practical use during gameplay, where quick adjustments are necessary as lane conditions change.

1.2 Distinction from Previous Work

This project distinguishes itself by simplifying the complex models from prior research into a practical framework that bowlers can use in real-time. Instead of delving into all possible variables, we focus on the most influential factors that a bowler can easily observe or adjust:

- **Ball Speed**: The initial speed of the ball upon release.
- **Rev Rate**: The rate of rotation of the ball, affecting its hook potential.
- **Axis of Rotation**: The angle at which the ball spins, influencing its trajectory.
- **Oil Pattern**: The distribution of oil on the lane, which changes over time and affects friction.

By making strategic approximations and developing a mathematical model using differential equations, we aim to create a tool that helps bowlers determine where to stand (the starting board) and where to aim (the target arrow) to consistently hit the pocket. Our model prioritizes usability, allowing for quick adjustments as lane conditions evolve, without the need for extensive measurements or complex calculations. This approach sets our work apart by balancing simplicity with essential accuracy.

2 Methodology

The model employs a system of differential equations to describe the motion of the bowling ball along the lane, accounting for the variables mentioned. The equations relate the starting position on the approach (board number), the target arrow on the lane, and the resulting trajectory of the ball.

2.1 Variables and Parameters

- m: Mass of the bowling ball (kg)
- R: Radius of the bowling ball (m)
- g: Acceleration due to gravity ($9.81\,\mathrm{m/s^2}$)
- $v(t)$: Linear velocity at time t (m/s)
- $\omega(t)$: Angular velocity at time t (rad/s)
- $\phi(t)$: Heading angle at time t (radians)
- $x(t)$: Downlane position at time t (m)
- $y(t)$: Lateral position at time t (m)

- $b(t)$: Board number at time t
- $\mu(x, b)$: Coefficient of friction at position x and board b
- K: Proportionality constant for heading angle change - colloquially, this is similar to how much the ball "hooks"
- Δt: Time step for numerical integration (s)
- L_{oil}: Oil pattern length (m)
- w_b: Board width (0.0254 m)

2.2 Oil Pattern Representation

Oil patterns are universally known to bowlers and are typically provided by bowling centers. They can be encoded using parameters such as *Left Board*, *Right Board*, and *Number of Loads*. These parameters allow bowlers to input any oil pattern into the model. To simplify the model while maintaining accuracy, we approximate the oil distribution by normalizing the number of loads and assuming uniform oil application within specified board ranges.

Table 1: Example Oil Pattern Data

Pass	**Left Board**	**Right Board**	**Number of Loads**
Forward	b_1	b_2	n
...	...	...	...
Reverse	b_3	b_4	n
...	...	...	...

2.3 Coefficient of Friction

The coefficient of friction $\mu(x, b)$ varies based on the oil pattern:

$$\mu(x,b) = \begin{cases} \mu_{\min} + (\mu_{\max} - \mu_{\min})[1 - o_b(b)], & x \leq L_{\text{oil}} \\ \mu_{\max}, & x > L_{\text{oil}} \end{cases} \tag{1}$$

Where:

- $o_b(b)$: Normalized oil volume fraction per board, calculated based on the number of loads (ranging from 0 to 1). This approximation allows us to simplify the complex oil distribution into a manageable function without significant loss of accuracy.
- $\mu_{\min}$: Minimum coefficient of friction on an oiled surface.
- $\mu_{\max}$: Maximum coefficient of friction on a dry surface.

2.4 Equations of Motion

2.4.1 Linear Velocity

$$\frac{dv(t)}{dt} = -\mu(x(t), b(t)) \cdot g \tag{2}$$

This equation models the deceleration of the ball due to friction, where $\mu(x(t), b(t))$ varies according to the oil pattern.

2.4.2 Angular Velocity

$$\frac{d\omega(t)}{dt} = -\frac{5}{2R}\mu(x(t), b(t)) \cdot g \tag{3}$$

This equation represents the change in angular velocity due to frictional torque. We approximate the ball as a solid sphere to simplify the moment of inertia, which is a reasonable assumption for practical purposes.

2.4.3 Heading Angle

$$\frac{d\phi(t)}{dt} = K\left[\omega(t) - \frac{v(t)}{R}\right] \tag{4}$$

The heading angle $\phi(t)$ represents the direction of the ball's motion. The change in heading angle depends on the difference between the ball's angular velocity and the ratio of its linear velocity to its radius. This relationship approximates the ball's tendency to hook when the rotational and translational motions are unbalanced, making the model practical yet sufficiently accurate.

2.4.4 Position Updates

$$\frac{dx(t)}{dt} = v(t)\cos(\phi(t)) \tag{5}$$

$$\frac{dy(t)}{dt} = v(t)\sin(\phi(t)) \tag{6}$$

These equations update the ball's position based on its speed and heading angle.

2.4.5 Board Number Calculation

$$b(t) = \left\lfloor \frac{y(t)}{w_b} \right\rfloor \tag{7}$$

This approximation allows us to discretize the continuous lateral position into specific boards, simplifying the model while keeping it relevant to how bowlers think about lane positioning.

2.5 Numerical Implementation

To solve the system of differential equations, we will use numerical integration methods, such as the Runge-Kutta method, with a time step Δt. By making these computational approximations, we ensure the model remains practical for real-time application.

2.5.1 Initial Conditions

- $v(0) = v_0$ (initial ball speed)
- $\omega(0) = \frac{\text{Rev Rate} \times 2\pi}{60}$
- $\phi(0) = \phi_0$ (initial heading angle)
- $x(0) = 0$
- $y(0) = y_0$ (based on starting board)

2.5.2 Integration Steps

At each time step t_n:

1. Compute $b(t_n)$ using Equation 7.
2. Retrieve $o_b(b(t_n))$ based on the oil pattern data.
3. Calculate $\mu(x(t_n), b(t_n))$ using Equation 1.
4. Update $v(t_{n+1})$, $\omega(t_{n+1})$, and $\phi(t_{n+1})$ using Equations 2, 3, and 4.
5. Update $x(t_{n+1})$ and $y(t_{n+1})$ using Equations 5 and 6.
6. Increment time: $t_{n+1} = t_n + \Delta t$.

2.6 Adaptability of the Model

The model is designed to accept any oil pattern input provided in the standard format of *Left Board*, *Right Board*, and *Number of Loads*. This universality is achieved by approximating the oil pattern into a normalized function, making the model both adaptable and practical for bowlers on any lane.

3 Expected Outcomes

By solving the system numerically, we can predict:

- The ball's trajectory $x(t)$ and $y(t)$.
- The change in heading angle $\phi(t)$, leading to the entry angle.
- How adjustments in initial conditions affect the entry angle.
- Optimal starting positions and aiming strategies based on lane conditions.

The model should help answer practical questions such as:

- If I move two boards to the left, where should I aim to maintain the optimal entry angle?
- How should I adjust my target as the oil pattern changes over the course of a game?

4 Conclusion

We have developed a mathematical model that incorporates key variables affecting a bowling ball's trajectory, including a detailed representation of the oil pattern. By making strategic approximations, we have simplified complex physics into a practical tool for bowlers, balancing usability with essential accuracy. This approach sets our work apart from previous models that are too intricate for real-time application. Future work will focus on computational implementation, empirical validation, and enhancements to provide real-time recommendations.

Future Work

The current Python implementation, detailed in the appendix, effectively models bowling ball trajectories using differential equations but has limitations. The representation of the oil pattern is simplified and does not fully capture real-world variations, which impacts accuracy. Further refinement of the oil pattern model is necessary to improve its alignment with empirical data.

Additionally, the model does not yet automate practical recommendations for adjustments, such as determining how a bowler's target should change when moving two boards to the left. Future work will focus on enhancing this functionality to provide actionable insights. Collecting empirical data to validate and calibrate the model, along with refining visualizations and user interfaces, will ensure the tool becomes a practical resource for bowlers seeking real-time advice.

References

[1] Frohlich, C. (2004). What Makes Bowling Balls Hook? *American Journal of Physics*, 72(9), 1170–1177.

[2] Stremmel, N. Entry Angle: Part 2. *International Bowling Pro Shop and Instructors Association (IBPSIA)*. Retrieved from `https://ibpsia.com/entry-angle-part-2/`.

[3] Stremmel, N., Ridenour, P., & Sterbenz, S. Identifying the Critical Factors That Contribute to Bowling Ball Motion on a Bowling Lane. *United States Bowling Congress*. Retrieved from `https://bowl.com`.

[4] United States Bowling Congress Equipment Specifications and Certifications Team. Ball Motion Study: Phase I and II Final Report. Retrieved from `https://bowl.com`.

A Appendix

A.1 Detailed Calculations

Below is a comprehensive and verbose account of all the calculations and derivations involved in developing the mathematical model for the bowling ball trajectory. This detailed exposition covers the definitions of variables, assumptions, derivations of equations, and the incorporation of the oil pattern into the model. The final section include the complete Python code to simulate the model.

B Variables and Parameters

Physical Constants and Parameters:

- m: Mass of the bowling ball (kg)
- R: Radius of the bowling ball (m)
- g: Acceleration due to gravity ($9.81\,\mathrm{m/s^2}$)
- I: Moment of inertia of the ball

  For a solid sphere:

$$I = \frac{2}{5}mR^2$$

State Variables:

- $v(t)$: Linear velocity at time t (m/s)
- $\omega(t)$: Angular velocity at time t (rad/s)
- $\phi(t)$: Heading angle at time t (radians)
- $x(t)$: Position along the lane (downlane distance) at time t (m)
- $y(t)$: Lateral position across the lane at time t (m)
- $b(t)$: Board number at time t (dimensionless)

Oil Pattern Variables:

- $o_b(b)$: Normalized oil volume fraction per board (dimensionless, $0 \leq o_b(b) \leq 1$)
- $o_x(x)$: Normalized oil volume fraction along the lane (dimensionless, $0 \leq o_x(x) \leq 1$)

- $o(x, b)$: Combined normalized oil volume fraction at position x and board b

Coefficient of Friction:

- $\mu(x, b)$: Coefficient of friction at position x along the lane and board b
- μ_{min}: Minimum coefficient of friction (on oiled surface)
- μ_{max}: Maximum coefficient of friction (on dry surface)

Constants and Empirical Parameters:

- K: Proportionality constant for the rate of change of heading angle (empirical)
- Δt: Time step for numerical integration (s)
- L_{oil}: Length of the oiled section of the lane (m)
- w_b: Width of one board on the lane (0.0254 m, or 1 inch)

C Assumptions

1. **Rigid Body:** The bowling ball is considered a rigid solid sphere.
2. **No Air Resistance:** Air resistance is negligible compared to frictional forces.
3. **Friction Model:** The friction between the ball and the lane depends on the coefficient of friction, which varies due to the oil pattern.
4. **Oil Pattern Symmetry:** The oil pattern is symmetrical about the lane's centerline.

5. **Rolling and Sliding:** The ball may slide initially but transitions to rolling without slipping as it moves down the lane.

6. **Lane Surface:** The lane surface is flat and horizontal.

D Derivation of Equations of Motion

D.1 Translational Motion (Linear Velocity)

Newton's Second Law for Translation:

The net force acting on the ball in the horizontal direction is due to friction:

$$F_{\text{net}} = m\frac{dv}{dt} = -F_f$$

Where:

- F_f: Frictional force.

Frictional Force:

$$F_f = \mu(x, b) \cdot N$$

Where:

- $N = m \cdot g$ (normal force, since the lane is horizontal).

Therefore:

$$F_f = \mu(x, b) \cdot m \cdot g$$

Substituting Back into Newton's Second Law:

$$m\frac{dv}{dt} = -\mu(x, b) \cdot m \cdot g$$

Simplify:

$$\frac{dv}{dt} = -\mu(x, b) \cdot g$$

Final Equation for Linear Velocity:

$$\frac{dv(t)}{dt} = -\mu(x(t), b(t)) \cdot g$$

D.2 Rotational Motion (Angular Velocity)

Newton's Second Law for Rotation:

$$\tau = I\frac{d\omega}{dt}$$

Where:

- τ: Torque acting on the ball.
- I: Moment of inertia.

Torque Due to Friction:

The frictional force creates a torque about the center of the ball:

$$\tau = F_f \cdot R$$

Substituting the Expression for Frictional Force:

$$\tau = \mu(x, b) \cdot m \cdot g \cdot R$$

Equate Torque and Angular Acceleration:

$$\mu(x, b) \cdot m \cdot g \cdot R = I \frac{d\omega}{dt}$$

Substitute Moment of Inertia for a Solid Sphere:

$$I = \frac{2}{5} m R^2$$

Therefore:

$$\mu(x, b) \cdot m \cdot g \cdot R = \left(\frac{2}{5} m R^2 \right) \frac{d\omega}{dt}$$

Simplify the Equation:

1. Cancel m from both sides.
2. Cancel one R from both sides.

$$\mu(x, b) \cdot g = \left(\frac{2}{5} R \right) \frac{d\omega}{dt}$$

Solve for $\frac{d\omega}{dt}$:

$$\frac{d\omega}{dt} = \frac{5}{2R}\mu(x, b) \cdot g$$

Note that the torque acts to increase the angular velocity, but since friction opposes motion, we include a negative sign:

$$\frac{d\omega}{dt} = -\frac{5}{2R}\mu(x, b) \cdot g$$

Final Equation for Angular Velocity:

$$\frac{d\omega(t)}{dt} = -\frac{5}{2R}\mu(x(t), b(t)) \cdot g$$

D.3 Heading Angle (Change in Direction)

Assumption:

The rate of change of the heading angle $\phi(t)$ is proportional to the difference between the angular velocity and the linear velocity divided by the radius.

Define Slip:

The slip between the ball and the lane is given by:

$$\text{Slip} = \omega(t) \cdot R - v(t)$$

Rate of Change of Heading Angle:

We model the rate of change of the heading angle as:

$$\frac{d\phi(t)}{dt} = K\left[\omega(t) - \frac{v(t)}{R}\right]$$

Where:

- K is an empirical proportionality constant.

Explanation:

- When $\omega(t) = \frac{v(t)}{R}$, the ball rolls without slipping, and there is no change in the heading angle.
- When there is a difference, the ball experiences a torque causing it to hook, changing the

  heading angle.

D.4 Position Equations

Components of Velocity:

The velocity components along the x and y axes are:

- $v_x(t) = v(t)\cos(\phi(t))$
- $v_y(t) = v(t)\sin(\phi(t))$

Differential Equations for Position:

$$\frac{dx(t)}{dt} = v_x(t) = v(t)\cos(\phi(t))$$

$$\frac{dy(t)}{dt} = v_y(t) = v(t)\sin(\phi(t))$$

D.5 Board Number Calculation

To determine the board number at time t:

$$b(t) = \left\lfloor \frac{y(t)}{w_b} \right\rfloor$$

Where:

- $w_b = 0.0254\,\mathrm{m}$ (1 inch, standard width of a board)
- $\lfloor \cdot \rfloor$ denotes the floor function, ensuring $b(t)$ is an integer.

D.6 Oil Volume Fraction Along the Lane ($o_x(x)$)

Define the oil pattern length:

$$L_{\mathrm{oil}} = 12.19\,\mathrm{m}(40 feet)$$

The normalized oil volume fraction along the lane is:

$$o_x(x) = \begin{cases} 1, & x \leq L_{\mathrm{oil}} \\ 0, & x > L_{\mathrm{oil}} \end{cases}$$

D.7 Combined Normalized Oil Volume Fraction ($o(x, b)$)

$$o(x, b) = o_b(b) \cdot o_x(x)$$

This represents the oil volume fraction at a specific position along the lane and across the boards.

D.8 Coefficient of Friction ($\mu(x, b)$)

The coefficient of friction depends on the oil volume fraction:

$$\mu(x, b) = \begin{cases} \mu_{\min} + (\mu_{\max} - \mu_{\min})[1 - o_b(b)], & x \leq L_{\text{oil}} \\ \mu_{\max}, & x > L_{\text{oil}} \end{cases}$$

Explanation:

- When $x \leq L_{\text{oil}}$:
  - μ varies between $\mu_{\min}$ and $\mu_{\max}$ based on $o_b(b)$.
  - If $o_b(b) = 1$ (maximum oil), $\mu = \mu_{\min}$.
  - If $o_b(b) = 0$ (no oil), $\mu = \mu_{\max}$.
- When $x > L_{\text{oil}}$:
  - The lane is considered dry, so $\mu = \mu_{\max}$.

E Complete System of Differential Equations

1. Linear Velocity:

$$\frac{dv(t)}{dt} = -\mu(x(t), b(t)) \cdot g$$

2. Angular Velocity:

$$\frac{d\omega(t)}{dt} = -\frac{5}{2R}\mu(x(t), b(t)) \cdot g$$

3. Heading Angle:

$$\frac{d\phi(t)}{dt} = K\left[\omega(t) - \frac{v(t)}{R}\right]$$

4. Positions:

$$\frac{dx(t)}{dt} = v(t)\cos(\phi(t))$$

$$\frac{dy(t)}{dt} = v(t)\sin(\phi(t))$$

5. Board Number:

$$b(t) = \left\lfloor \frac{y(t)}{w_b} \right\rfloor$$

F Initial Conditions

At $t = 0$:

- **Linear Velocity:**

$$v(0) = v_0$$

- **Angular Velocity:**

$$\omega(0) = \frac{\text{Rev Rate} \times 2\pi}{60}$$

(Rev Rate is in RPM)

- **Heading Angle:**

$$\phi(0) = \phi_0$$

- **Positions:**

$$x(0) = 0$$

$$y(0) = y_0 = \text{Starting Board} \times w_b$$

G Numerical Integration Procedure

To solve the system numerically, we use a suitable numerical method (e.g. Runge-Kutta methods).

Time Steps:

- Choose a time step Δt (e.g., 0.01 s).

Algorithm:

At each time step t_n:

1. **Calculate Board Number $b(t_n)$:**

$$b(t_n) = \left\lfloor \frac{y(t_n)}{w_b} \right\rfloor$$

2. **Retrieve $o_b(b(t_n))$:**

    - If $b(t_n)$ is in the oil pattern data, use the corresponding $o_b(b)$.
    - If not, $o_b(b(t_n)) = 0$.

3. **Compute $o(x(t_n), b(t_n))$:**

   - Determine $o_x(x(t_n))$:

$$o_x(x(t_n)) = \begin{cases} 1, & x(t_n) \leq L_{\text{oil}} \\ 0, & x(t_n) > L_{\text{oil}} \end{cases}$$

   - Compute $o(x(t_n), b(t_n)) = o_b(b(t_n)) \cdot o_x(x(t_n))$.

4. **Calculate Coefficient of Friction $\mu(x(t_n), b(t_n))$:**

   - If $x(t_n) \leq L_{\text{oil}}$:

$$\mu = \mu_{\min} + (\mu_{\max} - \mu_{\min})[1 - o_b(b(t_n))]$$

   - If $x(t_n) > L_{\text{oil}}$:

$$\mu = \mu_{\max}$$

5. **Update Linear Velocity $v(t_{n+1})$:**

$$v(t_{n+1}) = v(t_n) + \frac{dv(t_n)}{dt} \cdot \Delta t = v(t_n) - \mu \cdot g \cdot \Delta t$$

6. **Update Angular Velocity $\omega(t_{n+1})$:**

$$\omega(t_{n+1}) = \omega(t_n) + \frac{d\omega(t_n)}{dt} \cdot \Delta t = \omega(t_n) - \frac{5}{2R}\mu \cdot g \cdot \Delta t$$

7. **Update Heading Angle $\phi(t_{n+1})$:**

$$\phi(t_{n+1}) = \phi(t_n) + \frac{d\phi(t_n)}{dt} \cdot \Delta t = \phi(t_n) + K\left[\omega(t_n) - \frac{v(t_n)}{R}\right]\Delta t$$

8. **Update Positions:**

   - **Update $x(t_{n+1})$:**

$$x(t_{n+1}) = x(t_n) + \frac{dx(t_n)}{dt} \cdot \Delta t = x(t_n) + v(t_n)\cos(\phi(t_n)) \cdot \Delta t$$

   - **Update $y(t_{n+1})$:**

$$y(t_{n+1}) = y(t_n) + \frac{dy(t_n)}{dt} \cdot \Delta t = y(t_n) + v(t_n)\sin(\phi(t_n)) \cdot \Delta t$$

9. **Increment Time:**

$$t_{n+1} = t_n + \Delta t$$

10. **Check Termination Condition:**

   - If $x(t_{n+1}) \geq 18.29\,\text{m}$ (length of the lane), stop.
   - Or if $v(t_{n+1})$ becomes negligible.

H Example Calculation

Suppose we have the following initial conditions and parameters:

- **Mass of the Ball (m)**: $6.8\,\text{kg}$ (15-pound ball)
- **Radius of the Ball (R)**: $0.1085\,\text{m}$ (8.5 inches in diameter)

- **Initial Linear Velocity ($v(0)$)**: $6\,\mathrm{m/s}$

  Compute $\omega(0)$:

  $$\omega(0) = \frac{300\,\mathrm{RPM} \times 2\pi}{60} = 31.42\,\mathrm{rad/s}$$

- **Initial Heading Angle ($\phi(0)$)**: 0 radians

- **Starting Board**: Board 10

  Compute $y(0)$:

  $$y(0) = 10 \times w_b = 10 \times 0.0254\,\mathrm{m} = 0.254\,\mathrm{m}$$

- **Coefficients of Friction:**

  - $\mu_{\mathrm{min}} = 0.02$
  - $\mu_{\mathrm{max}} = 0.3$

- **Empirical Constant (K)**: 0.1

- **Time Step (Δt)**: $0.01\,\mathrm{s}$

- **Oil Pattern Length ($L_{\mathbf{oil}}$)**: $12.19\,\mathrm{m}$

At Time $t = 0$:

1. **Calculate Board Number:**

   $$b(0) = \left\lfloor \frac{y(0)}{w_b} \right\rfloor = \left\lfloor \frac{0.254}{0.0254} \right\rfloor = 10$$

2. **Retrieve $o_b(b(0))$:**

   - For Board 10, $o_b(10) = 1$

3. **Compute $o(x(0), b(0))$:**

- Since $x(0) = 0 \leq L_{\text{oil}}$, $o_x(0) = 1$
- Compute:

$$o(0, 10) = o_b(10) \cdot o_x(0) = 1 \cdot 1 = 1$$

4. **Calculate Coefficient of Friction $\mu(x(0), b(0))$:**

$$\mu = \mu_{\min} + (\mu_{\max} - \mu_{\min})[1 - o_b(10)] = 0.02 + (0.3 - 0.02)(1 - 1) = 0.02$$

5. **Compute $\frac{dv(0)}{dt}$:**

$$\frac{dv(0)}{dt} = -\mu \cdot g = -0.02 \cdot 9.81 = -0.1962\,\text{m/s}^2$$

6. **Compute $\frac{d\omega(0)}{dt}$:**

$$\frac{d\omega(0)}{dt} = -\frac{5}{2R}\mu \cdot g = -\frac{5}{2 \times 0.1085} \times 0.02 \times 9.81 = -4.527\,\text{rad/s}^2$$

7. **Compute $\frac{d\phi(0)}{dt}$:**

$$\frac{d\phi(0)}{dt} = K\left[\omega(0) - \frac{v(0)}{R}\right] = 0.1\left[31.42 - \frac{6}{0.1085}\right] = 0.1\,[31.42 - 55.30] = -2.388\,\text{rad/s}$$

8. **Update Linear Velocity $v(0.01)$:**

$$v(0.01) = v(0) + \frac{dv(0)}{dt} \cdot \Delta t = 6 - 0.1962 \times 0.01 = 5.9980\,\text{m/s}$$

9. **Update Angular Velocity $\omega(0.01)$:**

$$\omega(0.01) = \omega(0) + \frac{d\omega(0)}{dt} \cdot \Delta t = 31.42 - 4.527 \times 0.01 = 31.3747\,\text{rad/s}$$

10. **Update Heading Angle $\phi(0.01)$:**

$$\phi(0.01) = \phi(0) + \frac{d\phi(0)}{dt} \cdot \Delta t = 0 - 2.388 \times 0.01 = -0.02388\,\text{rad}$$

11. **Update Positions:**

    - **Update $x(0.01)$:**

$$x(0.01) = x(0) + v(0)\cos(\phi(0)) \cdot \Delta t = 0 + 6\cos(0) \times 0.01 = 0.06\,\text{m}$$

    - **Update $y(0.01)$:**

$$y(0.01) = y(0) + v(0)\sin(\phi(0)) \cdot \Delta t = 0.254 + 6\sin(0) \times 0.01 = 0.254\,\text{m}$$

Repeat the steps for each time increment.

I Calculations for Subsequent Time Steps

For each time step, the same procedure is followed:

1. Use the updated values of $v(t_n)$, $\omega(t_n)$, $\phi(t_n)$, $x(t_n)$, $y(t_n)$.
2. Calculate $b(t_n)$ based on $y(t_n)$.
3. Retrieve $o_b(b(t_n))$ and compute $o(x(t_n), b(t_n))$.
4. Calculate $\mu(x(t_n), b(t_n))$.
5. Compute derivatives and update the state variables.

6. Continue iterating until the ball reaches the pins or other termination conditions are met.

J Potential Outputs and Analysis

Trajectory Plot:

- Plot $y(t)$ versus $x(t)$ to visualize the ball's path down the lane.

Entry Angle:

- Calculate the final heading angle $\phi_{\text{entry}} = \phi(t_{\text{end}})$ when $x(t_{\text{end}}) = 18.29\,\text{m}$.

Effect of Variables:

- Analyze how changes in $v(0)$, $\omega(0)$, $\phi(0)$, and the oil pattern affect the trajectory and entry angle.

Optimization:

- Use the model to determine the optimal starting position and aiming strategy to achieve the desired entry angle.

K Python Implementation

This section provides the Python code implementation of the bowling ball motion model discussed in the previous sections. The code includes numerical methods for solving the differential equations representing the ball's motion on a bowling lane.

K.1 Python Code

```
import math
import matplotlib.pyplot as plt

# =========================================
# Constants and Parameters
# =========================================
g = 9.81
w_b = 0.0254    # Width of one board in meters

lane_length = 18.29  # 60 ft in meters
L_oil = 12.19         # Oil pattern length in meters

oil_pattern_data = [
    {'direction': 'Forward', 'left_board': 3,  'right_board': 3,  'loads': 1},
    {'direction': 'Forward', 'left_board': 7,  'right_board': 7,  'loads': 1},
    {'direction': 'Forward', 'left_board': 8,  'right_board': 8,  'loads': 2},
    {'direction': 'Forward', 'left_board': 10, 'right_board': 10, 'loads': 3},
    {'direction': 'Forward', 'left_board': 11, 'right_board': 11, 'loads': 3},
```

```
    {'direction': 'Reverse', 'left_board': 6,  'right_board': 6, 'loads': 1},
    {'direction': 'Reverse', 'left_board': 10, 'right_board': 10, 'loads': 3},
    {'direction': 'Reverse', 'left_board': 11, 'right_board': 11, 'loads': 3},
    {'direction': 'Reverse', 'left_board': 13, 'right_board': 13, 'loads': 3},
]

mu_min = 0.0001
mu_max = 0.02

K = 0.00025
# initial_ball_speed = 9.0   # m/s
phi0 = math.radians(-1.0)
# starting_board = 10
# y0 = starting_board * w_b

dt = 0.01

# Ball parameters
m = 6.8
diameter = 0.2159
R = diameter / 2
I = (2/5)*m*(R**2)  # Moment of inertia for a solid sphere

def compute_oil_distribution(oil_data):
    load_map = {}
    # Assume each entry affects the same length down the lane, only oil thickness varies
    for entry in oil_data:
```

```
        board_range = range(entry['left_board'], entry['right_board'] + 1)
        for b in board_range:
            if entry['direction'] == 'Forward':
                # Forward means adding oil from front to back, increasing load towards t
                increment = (b - min(board_range) + 1) * entry['loads']
            else:
                # Reverse means adding oil from back to front, increasing load towards t
                increment = (max(board_range) - b + 1) * entry['loads']
            load_map[b] = load_map.get(b, 0) + increment

    # Normalize the loads by the maximum value to keep values between 0 and 1
    if load_map:
        max_loads = max(load_map.values())
        for b in load_map:
            load_map[b] = load_map[b] / max_loads
    else:
        load_map = {}

    return load_map

oil_map = compute_oil_distribution(oil_pattern_data)

def oil_fraction(x, y):
    # Compute board number b(t)
    b = int(math.floor(y / w_b))
    # Clamp board number between 1 and 39
    b = min(max(b, 1), 39)
```

```
    o_b = oil_map.get(b, 0.0)

    # o_x(x) = 1 if x <= L_oil else 0
    if x <= L_oil:
        o_x = 1.0
    else:
        o_x = 0.0

    # o(x,b) = o_b(b)*o_x(x)
    return o_b * o_x

def mu_of_position(x, y):
    o_val = oil_fraction(x, y)
    if x <= L_oil:
        return mu_min + (mu_max - mu_min)*(1 - o_val)
    else:
        return mu_max

def equations_of_motion(t, state):
    # State: [v, omega, phi, x, y]
    v, omega, phi, x, y = state
    mu = mu_of_position(x, y)

    dv_dt = -mu * g
    domega_dt = -(mu*m*g*R)/I
    dphi_dt = K * (omega**1.3 + (v/R))
    dx_dt = v * math.cos(phi)
```

```
    dy_dt = v * math.sin(phi)

    return [dv_dt, domega_dt, dphi_dt, dx_dt, dy_dt]

def runge_kutta_4_step(t, state, dt):
    k1 = equations_of_motion(t, state)
    k1_state = [s + (dt/2)*kk for s, kk in zip(state, k1)]

    k2 = equations_of_motion(t + dt/2, k1_state)
    k2_state = [s + (dt/2)*kk for s, kk in zip(state, k2)]

    k3 = equations_of_motion(t + dt/2, k2_state)
    k3_state = [s + dt*kk for s, kk in zip(state, k3)]

    k4 = equations_of_motion(t + dt, k3_state)

    new_state = [
        s + (dt/6)*(k1_i + 2*k2_i + 2*k3_i + k4_i)
        for s, k1_i, k2_i, k3_i, k4_i in zip(state, k1, k2, k3, k4)
    ]
    return new_state

def simulate(initial_ball_speed, phi0, initial_rev_rate, starting_board):
    y0 = starting_board * w_b

    # initial_ball_speed = 9.0   # m/s
    # phi0 = math.radians(-1.0)
```

```
    # starting_board = 10
    # y0 = starting_board * w_b
    #

    omega0 = (initial_rev_rate * 2 * math.pi) / 60.0
    state = [initial_ball_speed, omega0, phi0, 0.0, y0]
    t = 0.0
    trajectory = [(t, *state)]  # (t, v, omega, phi, x, y)



    while state[3] < lane_length and state[0] > 0.1:
        state = runge_kutta_4_step(t, state, dt)
        t += dt
        trajectory.append((t, *state))

    final_mu = mu_of_position(state[3], state[4])


    return state, trajectory

def main():
    # (initial_ball_speed, phi0, initial_rev_rate, starting_board)
    trials = [(9, math.radians(-1.0), 100, 10), (6, math.radians(-2.0), 200, 15), (11, m
    plt.figure(figsize=(6, 8))

    for trial in trials:
```

```
        state, trajectory = simulate(*trial)
        v_final, omega_final, phi_final, x_final, y_final = state
        final_angle_deg = math.degrees(phi_final)

        xs = [p[4] for p in trajectory]
        ys = [p[5] for p in trajectory]
        boards = [y / w_b for y in ys]

        plt.plot(boards, xs, label=f'{trial[0]} m/s, {round(trial[1], 2)} Heading Angle,

    plt.scatter([20], [lane_length], s=100, color='red', label='Headpin')
    plt.xlabel('Board Number')
    plt.ylabel('Downlane Distance (m)')
    plt.title('Simulated Trajectories of Bowling Balls')
    plt.xlim(39, 0)
    plt.ylim(0, 20)
    plt.grid(True)
    plt.legend(loc='lower left', borderaxespad=4)
    plt.show()

if __name__ == "__main__":
    main()
```