# Modeling Baseball Batted Ball Trajectories for Realistic Simulation

Building a high-realism baseball **batted ball** module requires understanding the key factors that determine how far and where a hit ball travels. The outcome of a batted ball is primarily set by its **exit velocity** (speed off the bat) and **launch angle**, but many other factors come into play – including where on the bat the ball is struck, the spin imparted, air resistance (drag), the Magnus effect from spin, and environmental conditions like wind and air density. Below, we break down each of these factors and how they can be modeled, using empirical data and physics, to drive a **baseball simulation engine**.

## Exit Velocity and Launch Angle – The Foundations

**Exit velocity (EV)** is the speed of the baseball immediately after it leaves the bat. **Launch angle** is the vertical angle at which the ball comes off the bat (relative to horizontal). These two parameters largely dictate the initial trajectory and potential distance of a batted ball. Generally, higher exit velocities and a moderate launch angle produce the longest hits. Empirical Statcast data confirms what hitters intuitively know: **fly ball distance is maximized at a launch angle around 25–30°**, and this optimum angle shifts slightly downward for higher exit speeds[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed). In other words, an **optimal home run arc** is typically in the high-20s of degrees. If the launch angle is too low, even a hard-hit ball will be a line drive or grounder (likely a single or an out). If the angle is too high, the ball becomes a towering pop-up that falls short. The sweet spot is in between, where the ball has a good mix of upward angle and forward momentum.

Crucially, **every extra mph of exit velocity adds significant distance**. On average, distance increases on the order of **≈5 additional feet per 1 mph increase in exit speed**, for a given angle[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed). For example, a ball hit 100 mph off the bat at ~26° (a near-ideal home run trajectory) travels on the order of **400+ feet** under standard conditions[[2]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=I%20have%20done%20this%20kind,speed%20by%20about%20four%20mph). If you can ramp that up to 105 mph with the same angle, you might gain roughly 25 more feet of carry. This relationship is borne out in Statcast analyses of thousands of batted balls. In one study, 100 mph at 26° carried about 405 ft (under controlled conditions) whereas 96 mph at the same angle would travel roughly 20 ft shorter[[2]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=I%20have%20done%20this%20kind,speed%20by%20about%20four%20mph). The lesson for our simulation: small changes in exit velo or launch angle can dramatically change where the ball lands. We’ll need to model both with high resolution.

**Horizontal spray angle** (where the ball is hit relative to the field – pull side, center, or opposite field) is another consideration. By itself, the horizontal direction doesn’t change distance in a vacuum, but in the real world it correlates with spin differences (discussed later) that can affect carry. A ball hit to dead center with a given EV/angle often carries a bit farther than the “same” ball hit to left or right field, because hits to the center tend to have pure backspin, whereas balls pulled or hit oppo have sidespin that slightly reduces their distance. But the primary inputs for our batted ball module will be the **exit speed** and **vertical launch angle**, which set the stage for everything that follows.

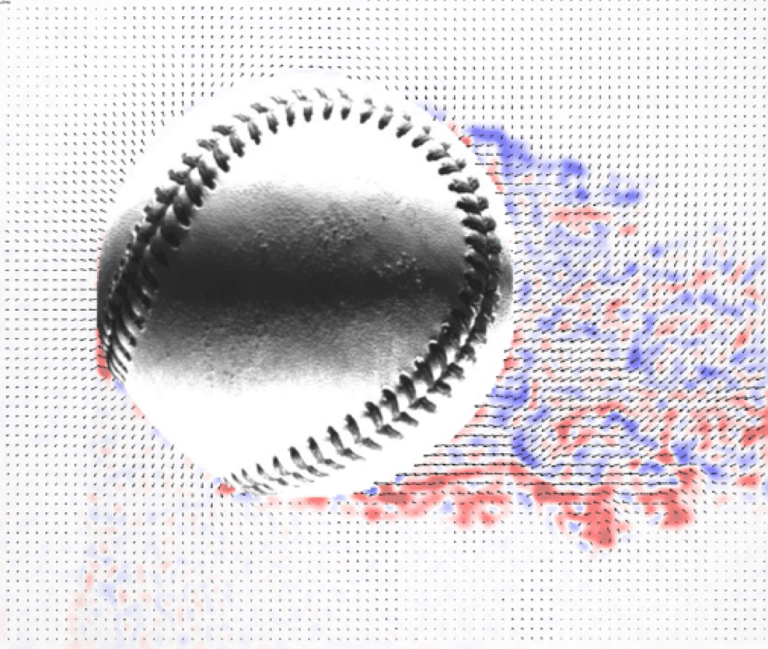
## Bat–Ball Collision and Contact Point (“Sweet Spot”)

How are exit velocity and launch angle produced? They are the result of the **bat–ball collision** dynamics. For a given swing, where the ball contacts the bat (especially relative to the bat’s sweet spot and the center of the ball) will determine the exit speed, launch angle, and spin of the batted ball. The **“sweet spot”** of the bat is the point on the barrel that maximizes energy transfer to the ball (while minimizing vibration). Hitting the ball on the sweet spot leads to a more efficient, elastic collision – meaning **higher exit velocity and longer distance** – whereas hitting off the end or near the hands results in lost energy (and that familiar stinging vibration)[[3]](https://smashitsports.com/blogs/smash-it-sports-blog/the-science-behind-sweet-spots-understanding-bat-performance-metrics?srsltid=AfmBOoqA7rcJgPeuHCKGmlSmWmxJW4RYu5E-in4KAAkOpLVXZVpzKB0a#:~:text=The%20sweet%20spot%20on%20a,game%20to%20the%20next%20level). In short, good contact on the sweet spot **“maximizes the trampoline effect, resulting in faster exit velocities and longer hits”**[**[3]**](https://smashitsports.com/blogs/smash-it-sports-blog/the-science-behind-sweet-spots-understanding-bat-performance-metrics?srsltid=AfmBOoqA7rcJgPeuHCKGmlSmWmxJW4RYu5E-in4KAAkOpLVXZVpzKB0a#:~:text=The%20sweet%20spot%20on%20a,game%20to%20the%20next%20level). Our simulation should account for this by giving the best outcomes to perfect, on-the-barrel contact.

**Off-center hits** will not only reduce exit velocity but also influence the launch angle and spin. If the bat strikes the ball **below its center**, the ball will launch upward with **backspin**; if struck above the center, the ball will be driven downward with **topspin**. This is why skilled hitters aiming for a home run try to hit slightly under the baseball – to get that backspin and loft. Conversely, a chopped swing can impart topspin (causing the ball to dive). Similarly, hitting the ball not squarely but with a glancing blow can introduce **sidespin**. For example, a right-handed batter who is a bit in front of the pitch might make contact such that the ball is hit toward left field with a slice of sidespin, whereas if they’re late the ball goes to right field with a hooking spin. These spin components will matter for the flight, as we’ll see.

One important aspect for our module is that **incoming pitch speed and bat speed** also factor into exit velocity. A faster pitch contributes more rebound velocity (all else equal), and a faster swing obviously does as well. The physics of collision (conservation of momentum and the coefficient of restitution between bat and ball) mean that: *Exit Velo* ≈ (some fraction of bat speed) + (some fraction of pitch speed). Typically, the bat speed is the dominant factor for homerun-distance hits, but a pitch’s speed (and spin, if it causes the bat to miss the ideal contact slightly) can modulate the outcome. At this stage, if we are focusing just on the batted ball, we might simply sample exit velocities from a realistic distribution or based on batter skill, but later when the pitching module is integrated, you will want a model for how pitch velocity + swing result in a particular EV and launch angle.

## Aerodynamics: Air Drag and the Magnus Effect (Spin)

  
*Airflow around a back-spinning baseball. The stitches and rotation create a pressure differential: the wake is deflected downward (as seen by the colored airflow disturbance behind the ball), which in turn produces an upward* *Magnus lift* *force on the ball.*

Once the ball is airborne, **physics of motion through air** take over. In a vacuum (no air), a baseball would follow a simple parabolic trajectory under gravity, but in reality **air resistance (drag)** and **Magnus forces** (from spin) profoundly influence the flight. **Drag** is the force of air friction that opposes the ball’s motion and continuously slows it down. A baseball moves fast enough that drag is a significant factor – proportional to the square of speed. This means that a 110 mph line drive experiences much more air resistance than a 70 mph bloop. Drag causes the ball to lose horizontal speed and fall shorter than it would in a vacuum. In fact, the familiar fact that 45° is the optimal launch angle for maximum distance only holds true *without* air resistance. With drag, the optimal angle is much lower (around 30° or less) because a higher arc keeps the ball in the air longer where drag can eat away its speed. Empirical data confirms that **fly balls in MLB peak in distance at ~25–30° launch angle** (not 45°), precisely due to drag slowing high-angled balls more[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed)[[4]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20are%20extremely%20valuable,exit%20speed%20and%20launch%20angle). And the faster the ball, the more pronounced the drag: the optimal angle actually shifts down a couple degrees for very hard-hit balls (e.g. 110+ mph) compared to slower hits[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed).

In simulation terms, we will model drag as a force $F\_d = \frac{1}{2} C\_d \rho A v^2$ acting opposite to the velocity vector (where $C\_d$ is the drag coefficient, $\rho$ air density, $A$ cross-sectional area, and $v$ speed). For a baseball, experimental and wind tunnel data show $C\_d$ is around 0.3–0.4 at typical speeds (it can vary with seam orientation and velocity). This drag significantly reduces a ball’s speed in flight – for example, a ball leaving the bat at 100 mph might be going only ~50–60 mph when it lands due to drag. The effect on distance is huge: **more drag = shorter distance,** and this is why a ball hit at Coors Field (thin air) travels farther than the same ball at sea level (denser air). We’ll quantify those environmental effects shortly.

**The Magnus effect (lift and curve from spin)** is the other big aerodynamic factor. A spinning ball generates an aerodynamic force **perpendicular to its flight path**, due to pressure differences in the airflow (as depicted in the image above). In baseball terms: **backspin** on a fly ball creates an upward lift force that counteracts gravity a bit and lets the ball stay aloft longer (resulting in further travel), whereas **topspin** creates a downward force that makes the ball drop quicker. This is the same Magnus lift that makes a curveball break downward – for a batted ball, it can make a well-hit fly ball carry an extra few dozen feet if there’s plenty of backspin. In our module, we will want to incorporate Magnus force $F\_l = \frac{1}{2} C\_l \rho A v^2$ in the upward direction (for backspin) or downward (for topspin), where $C\_l$ is a lift coefficient depending on spin rate. A highly spinning ball can have a lift coefficient on the order of 0.2.

How much does spin really matter? Quantitatively, a **ball with backspin will travel farther** than the same ball (same EV/angle) with no spin. Professor Alan Nathan analyzed Statcast data and found that at **103 mph and 27° launch**, increasing backspin from 0 up to ~2000 rpm can increase distance by ~20% (from roughly 336 ft to 400+ ft)[[5]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=the%20amount%20of%20backspin%2C%20for,increases%20with%20increasing%20spin%2C%20essentially)[[6]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=0%20336%20500%20368%201000,400%202500%20403%203000%20403). The first bit of backspin has the biggest effect: e.g. going from 0 spin (knuckled) to 1500 rpm might add ~60 feet (!) to a deep fly[[5]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=the%20amount%20of%20backspin%2C%20for,increases%20with%20increasing%20spin%2C%20essentially)[[6]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=0%20336%20500%20368%201000,400%202500%20403%203000%20403). However, there are **diminishing returns** to spin. Beyond a certain spin rate (around 1500–2000 rpm for that same example), additional backspin yields little extra carry[[5]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=the%20amount%20of%20backspin%2C%20for,increases%20with%20increasing%20spin%2C%20essentially)[[6]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=0%20336%20500%20368%201000,400%202500%20403%203000%20403). The distance gains **saturate** because while more backspin does increase lift, it also **increases drag** on the ball. In fact, Nathan notes that beyond about 2500–3000 rpm of spin, the **extra drag cancels out the extra lift**, so you don’t get any further carry[[7]](https://blogs.fangraphs.com/more-fun-with-batted-ball-spin-data/#:~:text=,%E2%80%9D). This is an important realism detail: our simulator should not just assume “more spin = always farther.” There’s an optimal range of backspin. Real MLB home runs typically have backspin on the order of 1500–2500 rpm. Very few well-hit balls have topspin, because to launch at a HR angle you usually impart backspin (as Dr. Nathan quipped, “it is very difficult for a ball hit at 25–30° to have topspin” – those high drives **almost always have backspin**[[8]](https://blogs.fangraphs.com/more-fun-with-batted-ball-spin-data/#:~:text=As%20well%20as%20this%3A)).

What about **sidespin**? Sidespin (spin around a vertical axis) doesn’t give lift, but it does cause the ball to curve sideways in flight – the Magnus force in that case pushes it left/right. A ball hit with sidespin will **“hook” or “slice”** similarly to a golf shot or a slice in tennis. For a right-handed batter, a ball hit to the pull side (left field) often has hooking sidespin (curving toward the foul line), while an opposite-field hit has slicing sidespin (tailing toward right field foul line). In our model, sidespin can be included by splitting the spin vector into its components: backspin (creates vertical lift) vs sidespin (creates horizontal curve). **Sidespin tends to reduce distance slightly** because it doesn’t help keep the ball up, and it actually increases total spin (hence drag) without adding lift. A research study using a trajectory calculator showed that adding a moderate amount of sidespin (~1500 rpm) to a ball with backspin can cut about **10–15 feet off** its distance, due to the extra drag and some complicated physics of how the spin axis tilts during flight[[9]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=those%20sum%02marized%20below%20were%20for,feature%20of%20the%20drag%20coefficient)[[10]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=that%20it%20increases%20with%20the,399%2C%20and%20a%20reduction). In sum, a ball hit with pure backspin (like one hit to dead center field for a righty) will carry farther than an equivalent ball hit with a lot of sidespin (common on flies down the foul lines). Our simulation can incorporate this by reducing carry for heavy sidespin or simply directly simulating the Magnus forces.

To implement spin effects, we will likely simulate the Magnus force dynamically. The spin rate and spin axis (direction) are determined at contact based on how the bat met the ball. For simplicity, one could start by assuming a typical backspin rate for fly balls (say 2000 rpm for a well-hit fly, lower for line drives) and maybe some sidespin if the spray angle is not center. A fully realistic model might even use the physics of the collision (the offset of the ball on the bat vertically and horizontally) to compute spin, but that can get complex. It might suffice to say: *if contact is made slightly under the center of the ball, assign a backspin in proportion to how much under; if contact is made with bat not perfectly squared (e.g., a bit of side cut), assign some sidespin.* Empirical evidence suggests opposite-field hits tend to have more spin (especially sidespin) than pulled hits[[11]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=4%20%E2%80%A2%20As%20a%20consequence,of%2010%E2%97%A6%20break%20toward%20the)[[12]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=hit%20at%20the%20same%20spray,spray%20angle%2C%20the%20amount%20of). For now, the main takeaway is that including spin (Magnus effect) will add a lot of realism: it explains why two hits with the same EV/launch can have different distances and landing spots.

## Environmental Factors: Wind, Altitude, Temperature, Humidity

Real baseballs are affected by the environment, and our simulator can incorporate these factors without needing an overly complex weather model. The **air density** through which the ball flies is crucial because it directly scales the drag and Magnus forces. **Thinner air (lower density)** means less drag and lift, so the ball goes farther (and curves less); **denser air** shortens distances. Several conditions influence air density:

* **Altitude:** Higher elevation = thinner air. For example, at **1,000 ft higher elevation, a fly ball travels about 6 feet farther** on average[[13]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=CHANGE%20IN%20FLYBALL%20DISTANCE%20WITH,CERTAIN%20ATMOSPHERIC%20EFFECTS). Denver’s Coors Field (about 5,200 ft elevation) is famous for yielding extra distance – on the order of 30+ feet more for a typical home run ball compared to sea level parks[[14]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=temperature,The%20Denver%20effect%20is%20huge)[[15]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=From%20Figure%204%20we%20learn,for%20both%20of%20these%20features). Our model should scale drag by air density for the stadium’s altitude.
* **Temperature:** Warm air is less dense than cold air. A **10°F increase in temperature adds roughly 3–4 feet** of carry distance[[13]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=CHANGE%20IN%20FLYBALL%20DISTANCE%20WITH,CERTAIN%20ATMOSPHERIC%20EFFECTS). So a hot summer day helps the ball fly a bit farther than a chilly day. (We could incorporate this by adjusting $\rho$ based on temperature in our physics calculations.)
* **Humidity:** Counterintuitively, more humid air is slightly less dense than dry air (water vapor displaces heavier molecules). However, the effect is *very small*. A **50% increase in relative humidity might add ~1 foot** of distance at most[[16]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Atmospheric%20Effect%20Change%20in%20Distance,blowing%20wind%2018.8%20ft). In other words, humid vs dry air won’t noticeably change the flight for our purposes (though humidity can affect the ball’s elasticity and seam, but that’s an advanced detail).
* **Wind:** Wind can have a dramatic impact. A wind blowing *out* (from home plate toward the outfield) effectively gives the ball extra push or reduces its relative airspeed, allowing it to carry much farther. Even a modest **5 mph tailwind adds on the order of 18–20 feet** of distance to a fly ball[[16]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Atmospheric%20Effect%20Change%20in%20Distance,blowing%20wind%2018.8%20ft). A headwind of the same speed would conversely rob ~15–20 feet. Crosswinds won’t change the distance as much, but will alter the ball’s horizontal path – potentially pushing a would-be fair ball foul or vice versa. In a simulation, wind can be modeled as an added velocity to the air relative to the ball. For simplicity, one can assume a constant wind vector during the flight (swirling or gusting winds would be more complex, and the user noted we probably don’t need a super detailed weather system). But including wind adds a lot of realism – e.g., players “know” in Wrigley Field a stiff wind can turn routine flies into homers or knock down balls that would have gone out on a calm day.

In practical terms, our module can allow environmental parameters such as air density (or equivalently, altitude + temp + humidity) and wind speed/direction as inputs. The trajectory calculations (drag force) use air density, and an effective **wind vector** would be subtracted from the ball’s velocity when computing drag/lift (since those forces depend on relative motion of ball vs air). This way, a tailwind reduces the relative speed through the air, thus reducing drag and keeping the ball’s speed higher for longer (more distance), whereas a headwind increases relative speed and drag. We have solid data to calibrate these effects – for instance, we can trust that ~5 mph of tailwind ≈ +19 ft distance as a reasonable rule[[16]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Atmospheric%20Effect%20Change%20in%20Distance,blowing%20wind%2018.8%20ft).

Finally, note that *other* environmental factors like air pressure (weather systems) or even slight differences in gravity (negligible across locations) could be considered, but they are second-order. The big ones are altitude, temperature, and wind.

## Putting It All Together – Trajectory Modeling for the Simulator

With all these factors identified, we can design our batted ball module to simulate the flight of a baseball from bat to landing. The general approach is to treat the ball as a projectile under forces and integrate its motion over time. **Inputs to the model** (from the collision module or from random sampling) will include: initial speed (exit velocity), launch angle (vertical), spray angle (horizontal direction), and spin (both backspin and sidespin rates). **Environmental inputs** include air density (which we can derive from park altitude and weather) and wind vector.

**Trajectory computation:** At each small time step $\Delta t$, we update the ball’s velocity and position. The forces acting on the ball are: gravity (constant downward), drag (opposing the velocity), and Magnus lift (perpendicular to velocity, direction determined by spin axis using right-hand rule). The equations of motion (in 2D or 3D) can be numerically integrated. This isn’t as daunting as it sounds – even a simple Euler method with small time steps (like 1–5 milliseconds per step) will yield a realistic trajectory. Alternatively, one can use a more refined integrator or even an analytical model if available. But given modern computing, it’s very feasible to simulate the flight in real-time in a game.

We will use known physics constants: mass of baseball (~0.145 kg), diameter ~0.074 m (giving cross-sectional area ~0.0043 m²), and empirically measured drag and lift coefficients. It’s worth noting that $C\_d$ (drag coefficient) for a baseball isn’t truly constant – it can vary with speed and spin (as mentioned, spin can slightly increase $C\_d$). In Alan Nathan’s model, he actually used a speed-dependent drag and included the increase of drag with spin to fit the data[[17]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20are%20extremely%20valuable,exit%20speed%20and%20launch%20angle)[[18]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Finally%2C%20I%20want%20to%20take,canceling%20the%20increase%20in%20lift). For simplicity, we might choose an average $C\_d \approx 0.35$ and then optionally add a small increment if spin is above a threshold. The **Magnus (lift) coefficient $C\_l$** can be related to the spin rate and velocity via a dimensionless spin factor (often denoted $\sigma$ or $S$). A rough approach: $C\_l$ increases with spin rate up to a point and then levels off (as we discussed, beyond ~2500 rpm no further lift gain). We could implement a formula or just cap the beneficial lift at that spin.

**Validation with empirical data:** We have plenty of data points to ensure our simulation is realistic. For instance, if we input a ball hit 100 mph at 28° with, say, 1800 rpm backspin, at sea level, no wind, we should get a carry on the order of 390–400 feet. If our simulation yields something wildly different, we’d tweak $C\_d$ or $C\_l$ until it matches known figures (like the 5 ft/mph rule and known distances). We can also incorporate the atmospheric adjustments from the earlier table: e.g. set density lower for Coors Field and confirm the ball flies ~30 ft farther.

In a simpler “empirical model” approach (if we didn’t integrate physics every time), one could use a precomputed lookup or equation fit for distance given EV, launch angle, etc. For example, using a polynomial or a machine learning model trained on Statcast data to predict distance and hang time. However, the physics-based approach has advantages: it naturally handles wind and spin and gives you the full trajectory (not just distance). Given that this will be part of a **game simulation**, having the actual trajectory (for determining if a ball clears a wall, how high it is, whether a fielder can catch it, etc.) is important – so doing the stepwise physics is worthwhile.

**Where on the field the ball lands** will be determined by both the distance traveled and the horizontal deflection. The distance along the ground from home plate comes from the combination of the horizontal velocity components and the flight time. Including drag means the ball’s horizontal speed is decelerating, so the range is less than a simple range formula. Including wind and Magnus means the range can’t be solved analytically in a trivial way – hence the simulation. The lateral (left-right) motion comes from the initial sideways velocity (if any, based on spray angle) plus any sideways push from wind or Magnus (sidespin). We should be able to calculate the landing coordinates (x, y distance in the field) from the trajectory.

To summarize, the batted ball module will:

1. **Take initial conditions** (EV, vertical angle, horizontal direction, spin, environment).
2. **Compute trajectory** under gravity, drag, and Magnus forces, using appropriate coefficients.
3. **Output** the ball’s flight path and landing spot (or wall impact, etc.).

Along the way, we’ve built in **realistic behaviors**: Hits on the sweet spot will tend to have higher EV and maybe optimal spin. Off-center hits could be modeled to have more sidespin or topspin and thus shorter carry. High-altitude or hot weather games will see the ball carry farther (which the module will reflect by reduced air resistance). A windy day will visibly push fly balls around. All of these factors are grounded in empirical research and physics data, so the simulation should feel true-to-life.

Finally, keep in mind that after we nail the batted ball flight, the next step is to integrate with the pitching module. That will involve modeling how pitch velocity, spin, and batter swing mechanics produce those initial conditions (EV, angle, etc.). But by structuring the batted ball module with the inputs mentioned, we make it straightforward to connect the two – e.g., the collision model would supply the exit velo and spin based on the pitch. At that stage, you might also incorporate probability distributions (for hit types, mishits, etc.). For now, with a solid batted ball flight model in place, you’ll have the core of the **simulation engine** ready for further expansion into a full-fledged baseball game.

**Sources:** The above analysis is supported by physics research and Statcast data on baseball trajectories. Key references include Alan Nathan’s studies of fly ball distance versus exit velocity/angle[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed)[[2]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=I%20have%20done%20this%20kind,speed%20by%20about%20four%20mph), his aerodynamic model incorporating drag and lift[[17]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20are%20extremely%20valuable,exit%20speed%20and%20launch%20angle), and his findings on backspin’s influence on carry (showing diminishing returns beyond ~1500 rpm)[[18]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Finally%2C%20I%20want%20to%20take,canceling%20the%20increase%20in%20lift)[[5]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=the%20amount%20of%20backspin%2C%20for,increases%20with%20increasing%20spin%2C%20essentially). Empirical figures for environmental effects (altitude, temperature, humidity, wind) are drawn from the same data-driven model[[13]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=CHANGE%20IN%20FLYBALL%20DISTANCE%20WITH,CERTAIN%20ATMOSPHERIC%20EFFECTS). The importance of contact quality and sweet-spot hits for maximizing exit velocity is well-documented in batting research[[3]](https://smashitsports.com/blogs/smash-it-sports-blog/the-science-behind-sweet-spots-understanding-bat-performance-metrics?srsltid=AfmBOoqA7rcJgPeuHCKGmlSmWmxJW4RYu5E-in4KAAkOpLVXZVpzKB0a#:~:text=The%20sweet%20spot%20on%20a,game%20to%20the%20next%20level). All these insights have been combined to inform the design of a realistic batted ball physics module. With these in hand, we can confidently simulate where a baseball will go when it’s hit – from the crack of the bat to the final landing spot.[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed)[[13]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=CHANGE%20IN%20FLYBALL%20DISTANCE%20WITH,CERTAIN%20ATMOSPHERIC%20EFFECTS)

[[1]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20show%20quantitatively%20what,mph%20increase%20in%20exit%20speed) [[2]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=I%20have%20done%20this%20kind,speed%20by%20about%20four%20mph) [[4]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20are%20extremely%20valuable,exit%20speed%20and%20launch%20angle) [[13]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=CHANGE%20IN%20FLYBALL%20DISTANCE%20WITH,CERTAIN%20ATMOSPHERIC%20EFFECTS) [[14]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=temperature,The%20Denver%20effect%20is%20huge) [[15]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=From%20Figure%204%20we%20learn,for%20both%20of%20these%20features) [[16]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Atmospheric%20Effect%20Change%20in%20Distance,blowing%20wind%2018.8%20ft) [[17]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=These%20data%20are%20extremely%20valuable,exit%20speed%20and%20launch%20angle) [[18]](https://tht.fangraphs.com/going-deep-on-goin-deep/#:~:text=Finally%2C%20I%20want%20to%20take,canceling%20the%20increase%20in%20lift) Going Deep on Goin’ Deep | The Hardball Times

<https://tht.fangraphs.com/going-deep-on-goin-deep/>

[[3]](https://smashitsports.com/blogs/smash-it-sports-blog/the-science-behind-sweet-spots-understanding-bat-performance-metrics?srsltid=AfmBOoqA7rcJgPeuHCKGmlSmWmxJW4RYu5E-in4KAAkOpLVXZVpzKB0a#:~:text=The%20sweet%20spot%20on%20a,game%20to%20the%20next%20level) The Science Behind Sweet Spots: Understanding Bat Performance Metrics – Smash It Sports

<https://smashitsports.com/blogs/smash-it-sports-blog/the-science-behind-sweet-spots-understanding-bat-performance-metrics?srsltid=AfmBOoqA7rcJgPeuHCKGmlSmWmxJW4RYu5E-in4KAAkOpLVXZVpzKB0a>

[[5]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=the%20amount%20of%20backspin%2C%20for,increases%20with%20increasing%20spin%2C%20essentially) [[6]](https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf#:~:text=0%20336%20500%20368%201000,400%202500%20403%203000%20403) baseball.physics.illinois.edu

<https://baseball.physics.illinois.edu/THT-Fly-Ball-Distances-Statcast.pdf>

[[7]](https://blogs.fangraphs.com/more-fun-with-batted-ball-spin-data/#:~:text=,%E2%80%9D) [[8]](https://blogs.fangraphs.com/more-fun-with-batted-ball-spin-data/#:~:text=As%20well%20as%20this%3A) More Fun With Batted Ball Spin Data | FanGraphs Baseball

<https://blogs.fangraphs.com/more-fun-with-batted-ball-spin-data/>

[[9]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=those%20sum%02marized%20below%20were%20for,feature%20of%20the%20drag%20coefficient) [[10]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=that%20it%20increases%20with%20the,399%2C%20and%20a%20reduction) [[11]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=4%20%E2%80%A2%20As%20a%20consequence,of%2010%E2%97%A6%20break%20toward%20the) [[12]](https://baseball.physics.illinois.edu/carry-v2.pdf#:~:text=hit%20at%20the%20same%20spray,spray%20angle%2C%20the%20amount%20of) baseball.physics.illinois.edu

<https://baseball.physics.illinois.edu/carry-v2.pdf>