Appendix I

Here's a simplified model in Mathematics to understand the basic principles.[3–5]

1. Potential Flow: Instead of solving the full Navier-Stokes equations, we can assume a potential flow, where the velocity field can be derived from a scalar potential function. This simplifies the math considerably.
2. Magnus Effect: We can model the Magnus effect by introducing a lift force proportional to the ball's spin and velocity.
3.

```
(*Parameters*)ballRadius = 0.11;
(*meters*)ballMass = 0.45;
(*kg*)airDensity = 1.225;
(*kg/m^3*)dragCoefficient = 0.2;
(*approximate for a sphere*)g = 9.81;
(*m/s^2*)(*Functions*)magnusForce[v_, w_] := airDensity * Pi * ballRadius^2 * Cross[v, w]
dragForce[v_] := -0.5 * airDensity * dragCoefficient * Pi * ballRadius^2 * Norm[v] * v
acceleration[v_, w_] := (magnusForce[v, w] + dragForce[v]) / ballMass

(*Initial conditions*)
initialPosition = {0, 0, 0};
initialVelocity = {20, 0, 15};
(*m/s*)initialAngularVelocity = {0, 0, 100};
(*rad/s*)(*Simulation*)tMax = 5;
dt = 0.01;
tValues = Range[0, tMax, dt];

v = initialVelocity;
w = initialAngularVelocity;
r = initialPosition;

data = {};
For[t = 0, t ≤ tMax, t += dt, a = acceleration[v, w];
 v += a * dt;
 w = w; (*Assuming constant angular velocity for simplicity*)r += v * dt;
 AppendTo[data, {t, r}];]

(*Visualization*)
ParametricPlot3D[Evaluate[r /. data], {t, 0, tMax}, PlotRange → All]
```

Code 1 Mathematics code to illustrate interaction between turn, speed and Magnus effect to bend of football and plot the diagram.

Appendix II

Here is a sample Mathematics code for simulating the trajectory of a football under sidekick with vortex shedding and negative Magnus effect, along with plotting the result:

```
(*Define parameters*)m = 0.42;
(*Mass of the football (kg)*)R = 0.11;
(*Radius of the football (m)*)v0 = 20;
(*Initial velocity (m/s)*)omega = -100;
(*Angular velocity (rad/s)-negative for negative Magnus effect*)rho = 1.225;
(*Air density (kg/m^3)*)Cl = 0.2;
(*Lift coefficient*)Cd = 0.1;
(*Drag coefficient*)(*Define functions*)dragForce[v_] := 0.5 * Cd * rho * Pi * R^2 * v^2;
liftForce[v_] := 0.5 * Cl * rho * Pi * R^2 * v * Abs[omega];
MagnusForce[v_] := liftForce[v];
(*Negative sign included in liftForce definition*)dt = 0.01;
(*Time step (s)*)t = Range[0, 2, dt];
(*Time range (s)*)(*Initialize position and velocity*)x = 0;
y = 0;
vx = v0;
vy = 0;

(*Store trajectory data*)
xPos = {x};
yPos = {y};

(*Loop through time steps*)
Do[(*Calculate forces*)fdrag = dragForce[vx];
  flift = MagnusForce[vx];
  (*Update accelerations*)ax = (flift - fdrag) / m;
  ay = -9.81;
  (*Update velocities*)vx = vx + ax * dt;
  vy = vy + ay * dt;
  (*Update positions*)x = x + vx * dt;
  y = y + vy * dt;
  (*Store trajectory data*)AppendTo[xPos, x];
  AppendTo[yPos, y];, {i, 1, Length[t]}];

(*Plot the trajectory*)
Plot[{y}, {x, 0, xPos[[Last]]}, PlotRange → {{0, Max[yPos] + 1}, {0, xPos[[Last]] + 10}},
  AxesLabel → {"Y (m)", "X (m)"}, PlotLabel → "Football Trajectory with Magnus Effect"];
```

Code 2 Mathematics code for simulating the trajectory of a football under sidekick with vortex shedding and negative Magnus effect.