流体力学背景下的足球动力学分析

摘要

足球是一项热门的体育运动，其入射角度和入射地点是作为进球的关键原因。本文主要研究足球的入射角度和入射地点，基于数学方法，微分方程法，及流体力学建立数学模型基于流体力学的足球运动数学建模分析。

针对问题一，运用流体动力学分析方法，从基本守恒定律出发，推导空气阻力与升力公式，结合伯努利方程推导马格努斯力公式，探究足球平动动能与转动动能分配对平动阻力的影响机制，明晰转动动能占比通过改变气流状态、阻力系数影响平动阻力的过程 。

针对问题二，在空气为低粘度流体假设下，建立包含空气阻力的足球运动方程，结合球门位置与尺寸约束，以进球时间最短为目标函数，构建初位置和踢球角度未知情况下的优化模型，求解最优进球策略 。又在假设空气为低粘度流体的前提下，构建了包含空气阻力的足球运动方程。结合球门的位置与尺寸约束，以实现进球时间最短为目标函数，建立了在初位置和踢球角度未知情况下的优化模型。通过该模型求解，可得到满足进球条件的最优策略，为初速度一定时的足球射门提供了理论参考。

针对问题三，基于问题一的力公式，运用三维动力学分析与数值求解方法，考虑球体旋转方向对马格努斯力的影响，建立三维运动方程，模拟不同旋转参数和动能分配下的轨迹曲线，总结旋转方向、转动动能占比等对轨迹弧度、落点偏移量的影响规律。

关键词：足球运动；流体力学；运动轨迹优化

目录

一、问题描述

1.我们将先推导出空气阻力的升力和阻力公式以及在动能如何分配对平动方向阻力的影响，马格努斯力公式和伯努利方程。

2.为保证进球的质量，我们要将空气视为低粘度流体的背景下解决足球在初速度一定的情况下进球时间最短，其中初位置和踢球角度未知。

3.我们将在第一个问题的基础上讨论三维空间中足球的运动轨迹与球体旋转方向等的关系。

二、问题分析

1.对于问题一，主要为聚焦于足球运动中空气作用力及动能分配影响。需从流体动力学分析，推导空气阻力和升力公式。基于伯努利方程，分析空气流速与压强关系，进而推导马格努斯力公式，同时，探究足球平动动能与转动动能的分配对平动方向阻力的作用，转动动能占比增加会改变空气绕流状态，通过阻力系数的变化影响平动阻力大小。

2.对于问题二，在将空气视为低粘度流体的前提下，优化足球在初速度固定时的进球时间。由于初位置和踢球角度未知，需建立包含空气阻力的运动方程，结合球门位置与尺寸约束，以进球时间为目标函数构建优化模型。

3.对于问题三，是对于问题 1 的力公式推导，深入探讨三维空间中足球运动轨迹的影响因素。考虑球体旋转方向对马格努斯力方向和大小的影响，通过建立三维动力学方程，分析足球在 x，y，z 三个方向的运动轨迹变化，进行绘图。

三、公式推导和假设

3.1 假设

1.由于流体普遍都具有粘度，而粘度的影响会导致空气阻力的复杂性大大增加，**所以我们假设空气中的流体为低粘度流体**

2.由于空气并非孤立系统，所以必定存在**热交换**，这就会导致流体方程需要联系**热力学状态参量求解问题**，而这种问题对我们队伍所要讨论的问题影响甚微，因为一般情况下大气环境参数变化不会很剧烈，**所以我们将在理想流体的情况下建立模型。理想流体既无热交换并且不可压缩的流体，因此其处于的每一个状态都是热力学平衡态**

3.我们假设流体并不会产生湍流

3.2 Magnus 力推导

在接下来的空气阻力计算中需要通过具体的数值计算速度球体的运动轨迹问题，所以这里需要计算 Magnus 力，不过在推导之前，需要说明一些基本的公式

（1）**空气阻力公式，空气阻力一般包括摩擦阻力与碰撞阻力**，而阻力公式有很多解法去做，例如可以用**量纲分析法或微元法求解**，这里不过多赘述，

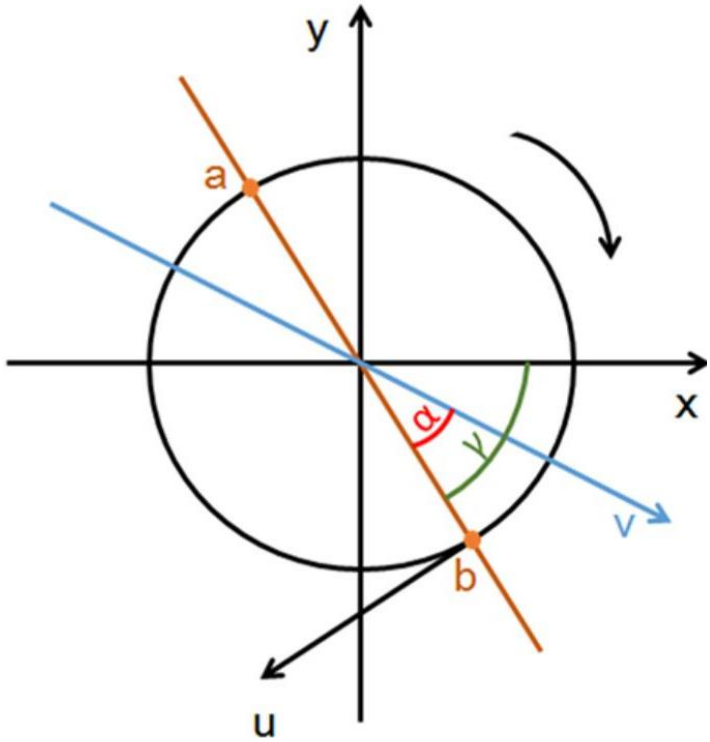
（2）**伯努利定理:**在在流体的无黏无热传导的定常运动中，单位质量流体的总能量沿同一条流线保持不变。其表述为：

$$\boldsymbol{H=\frac{1}{2}v^2+\varepsilon+\frac{p}{\rho}+\Psi=const} \tag{1}$$

其中ε为质点内能，而在不可压缩流中$\frac{d\rho}{dt}=0$，同时由于质点处于热平衡态，由热力学第一定律可知此时$\frac{d\varepsilon}{dt}=0$，做出该判断也是为简化问题所需，因此对于不可压缩定常流在重力场中的伯努利方程可以写为：

$$\boldsymbol{\frac{1}{2}v^2+\frac{p}{\rho}+gz=const} \tag{2}$$

球体旋转时的情况如图：



图表一 球体旋转状态平面分析

此时球体表面的流体流速为$\boldsymbol{u = AR \times \Omega}$，A 是与滑翔机表面性质、内摩擦、粘滞性有关的常数，取值范围[0,1] ，则 a、b 点的表征速度为：

$$\boldsymbol{v_a^2 = v^2 + u^2 - 2vucos\left(\frac{\pi}{2} + \alpha\right) = v^2 + u^2 + 2vusin} \tag{3}$$

$$\boldsymbol{v_b^2 = v^2 + u^2 - 2vucos\left(\frac{\pi}{2} - \alpha\right) = v^2 + u^2 - 2vusin} \tag{4}$$

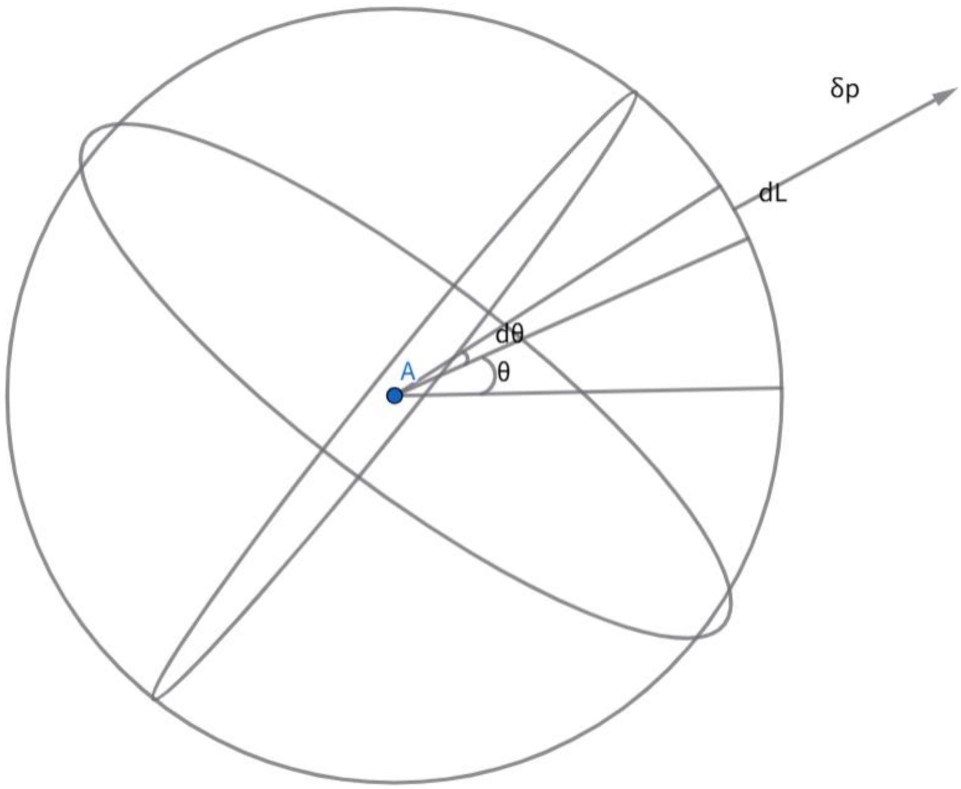
至此用伯努利方程则有 a、b 点的压强差：

$$\begin{aligned} \Delta \boldsymbol{p = p_b - p_a = \frac{1}{2}\rho v_a^2 - \frac{1}{2}\rho v_b^2 + \rho g(h_a - h_b)} \\ \boldsymbol{= 2\rho ARsin\varphi\omega vsin\alpha + 2\rho gRsin\varphi sin(\alpha + \beta)} \end{aligned} \tag{5}$$

而α$\boldsymbol{\alpha}$与$\boldsymbol{\gamma}$之间的夹角是一定值则可以用一β角表示，既$\boldsymbol{\gamma - \alpha = \beta}$

注意到此时球体小面元的受力为：

$$\boldsymbol{dF = \varDelta PdS}$$



图表二 球体旋转状态立体分析

则对上式积分有：

$$F=\int_0^{\pi}\int_0^{R_0}\Delta PdS=R^2\iint_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\Delta Psin\alpha d\alpha d\varphi$$

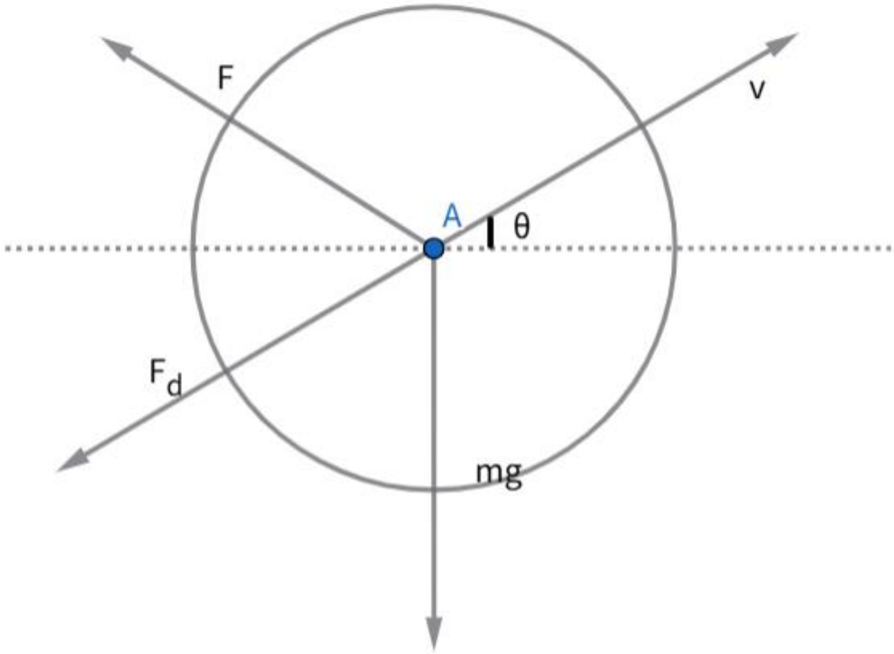
$$=R^2\iint_0^{\frac{\pi}{2}}[2\rho ARsin\varphi\omega vsin\alpha+2\rho gRsin\varphi sin(\alpha+\beta)]d\varphi d\alpha$$

$$=2\rho AR\omega v+2\rho gR(sin\beta+cos\beta) \tag{6}$$

我们发现静止流体中**马格努斯力的方向总是垂直于质心平动方向**，这得益于球体的**高度对称性**。这样我们就只需要考虑一个方向上的马格努斯力证明。

四、微分方程建立与求解

条件与前面一致，此时可以给出球体的受力分析：



图表三 球体旋转状态受力分析

由牛顿第二定律可列出 x，y 分量的运动微分方程(F 是马格努斯力，Fd 是流体的碰撞阻力)：

$$m\frac{d\vec{v}_x}{dt}=-F\sin\theta-F_d\cos\theta \tag{7}$$

$$m\frac{d\vec{v}_y}{dt}=-F\cos\theta-F_d\sin\theta-mg \tag{8}$$

这样只需要解该微分方程就可以导出平面上的运动方程

$$m\frac{\vec{v}_x}{dt} = -[2\rho AR\omega v + 2\rho gR(sin\theta + cos\theta)]\sin\theta$$

$$-\frac{1}{2}\pi R(\sigma_d SV^2\rho)\cos\theta \tag{9}$$

$$m\frac{d\vec{v}_y}{dt} = -[2\rho AR\omega v + 2\rho gR(sin\theta + cos\theta)]\cos\theta$$

$$-\frac{1}{2}\pi R(\sigma_d SV^2\rho)\sin\theta - mg \tag{10}$$

接下来就是对微分方程的解进行讨论。

原方程可整理为一阶非线性常微分方程（二次型）：为了方便计算我们对国际足协给出的标准进行赋值，其中国际足联规定的标准足球周长在 68.5 至 69.5 **厘米之间**，重量在 420 至 445 **克**之间，直径为 22.1cm。这样我们便进行以下赋值并且进行计算 $\rho = 1.225\,kg/m^3$，$g = 9.8\,m/s^2$，$R = 0.1105\,m$，$\sigma d = 0.5$，$\omega = 1\,rad/s$，$m = 0.430\,kg$，$A = S = \pi R^2$

其中各系数定义为：

$$\text{二次项系数：} k2 = -0.5\,\pi Rodsp\;sin\;\theta \div m\;\approx -0.247\;\;sin\;\;\theta$$

$$\text{线性项系数：} k1 = -2\,\rho A\,R\,\omega\cos\,\theta \approx -0.630\;\cos\;\theta$$

$$\text{常数项：} k0 = -2\rho gR(\sin\theta + \cos\theta)\cos\theta$$

$$\approx -6.17(\sin\;\theta + \;\cos\;\theta)\;\cos\;\theta - 9.8$$

这样便可简化问题，由于对于有理函数的实根情况不同会有不同的解，所以我们可以对${k_1}^2 - k_2k_0$的取值不同运用不同的积分公式得到求解。

$\Delta(\theta) \approx 0.3969\cos^2\theta - 6.10\sin\theta\,(\sin\theta\cos\theta + \cos^2\theta) - 9.68\sin\theta$

当 ${k_1}^2 - k_2k_0 < 0$ 时其速度变化率小于 0，既此时在 y 方向上的速度不断减小，运用积分公式并且将初始条件 $t = 0$，$v = V\sin\theta$ 有。

$$t(V) = -\frac{2}{\sqrt{k_2k_0 - k_1^2}}\ln\left|\frac{2k_2V + k_0 - \sqrt{k_1^2 - k_2k_0}}{2k_2V + k_0 + \sqrt{k_1^2 - k_2k_0}}\right| + C \tag{11}$$

其中，

$$C=\frac{2}{\sqrt{k_2k_0-k_1^2}}\ln\left|\frac{2k_2V\sin\theta+k_0-\sqrt{k_1^2-k_2k_0}}{2k_2V\sin\theta+k_0+\sqrt{k_1^2-k_2k_0}}\right| \tag{12}$$

对于 x 方向的速度也是同样的解法只是在 x 分量上少了常数系数，这样方程就是一个一阶线性微分方程，与 y 方向的解法类似因此我们便不再重复，直接给出结果，用参数 m 表示方程的系数。

二次项系数：$\boldsymbol{m_2=-\frac{1}{2}\pi R\sigma_d S\rho\cos\theta=0.0041cos\theta}$

线性项系数：$\boldsymbol{m_1=-2\rho AR\omega sin\theta=0.0938sin\theta}$

常数项系数；$\boldsymbol{m_0=2\rho gR(sin\theta+cos\theta)\sin\theta=3.045(sin\theta+cos\theta)\sin\theta}$

这样便可以得到形式类似的函数

接下来我们可以尝试对角度 θ 给与不同的数值，以更好地求解该方程，为了简化运算又不缺代表性，我们将赋值并列出表格：

图表四 不同角度下 x 方向上的求解

θ	cos θ	sin θ	a（v² 项系数）	b（v 项系数）	c（常数项）	判别式 $D=b2-4ac$	D
1°	0.9998	0.0175	-0.6299	-0.00431	-15.041	≈ 0.1372	$D>0$
15°	0.9659	0.2588	-0.00916	-0.00624	-0.4245	≈ -0.0155	$D<0$
30°	0.8660	0.5000	-0.00821	-0.01205	-2.6153	≈ -0.0855	$D<0$
45°	0.7071	0.7071	-0.00670	-0.01704	-5.7925	≈ -0.153	$D<0$
60°	0.5000	0.8660	-0.00474	-0.02087	-8.5636	≈ -0.161	$D<0$
75°	0.2588	0.9659	-0.00245	-0.02327	-9.9421	≈ -0.096	$D<0$
90°	0	1	0	-0.0241	-6.17	≈-0.00058	$D<0$

图表五 不同角度下 y 方向上的求解

θ	a（v² 项系数）	b（v 项系数）	c（常数项）	判别式 $D=b2-4ac$	D
1°	-0.000166	-0.0241	-15.96	≈ -0.0102	D<0
15°	-0.00245	-0.0233	-15.64	≈ -0.151	D<0
30°	-0.00474	-0.0208	-14.68	≈ -0.276	D<0
45°	-0.00670	-0.0170	-12.86	≈ -0.340	D<0
60°	-0.00821	-0.0120	-10.43	≈ -0.341	D<0
75°	-0.00916	-0.00624	-7.82	≈ -0.287	D<0
90°	-0.00948	0	-9.8	≈ -0.371	D<0

在我们便得到了速度与时间的关系，但是如果要继续求积分的话就会比较困难，所以我们可以进行变换，

$$\frac{dv}{dt}=\frac{dv}{dy}\frac{dy}{dt}=\frac{dV}{dy}V$$

则可以得到

$$V\frac{dV}{k_2V^2+k_1V+k_0}=-dy$$

积分得

$$y=\frac{1}{2k_2}\ln\left|k_2V^2+k_1V+k_0\right|+\frac{k_1}{2k_2}t(V)+C_2 \tag{13}$$

其中，$t=o$ 时 $y=0$，$v=Vsin\theta$

此时便可以解得积分系数

$$C_2=-\frac{1}{2k_2}\ln\left|k_2Vsin\theta^2+k_1Vsin\theta+k_0\right| \tag{14}$$

这样便能得到关于速度时间与位移之间的关系，接下来只需要给定边界条件即可得到问题的解。(为了方便表达下面的式子就沿用 C 的常数表达式)

五、模型建立与求解

现在我们假设有一个**高 2.44 米，高为 7.32 米**的足球网，有人在其前面未知位置（设为 d）以不固定的初速度和角速度，以某位置角度出射，**以人为惯性参考系，**当然这里暂时只讨论 XoY 坐标系的情况。

球若想进入球门，可以在 x=d 处是恰好速度等于 0，然后在球门正上方做有空气阻力的自由落体运动，但是该情况不符合实际所以我们将不讨论该情况。

接下来先讨论**进球时间与出球角度的关系**

首先我们令 **x=d**，这样我们便可得到**球到达球门的时间** *t0*，这样我们便可以得到球到达 x=d 时的 y 坐标，此时给出限定条件：

$$2\geq y\geq 0, v_y>0, v_x>0$$

即可得到初速度与角度的范围，由于通过一般方法求解比较困难，所以我们可以运用计算机进行参数逼近或者是进行求解，

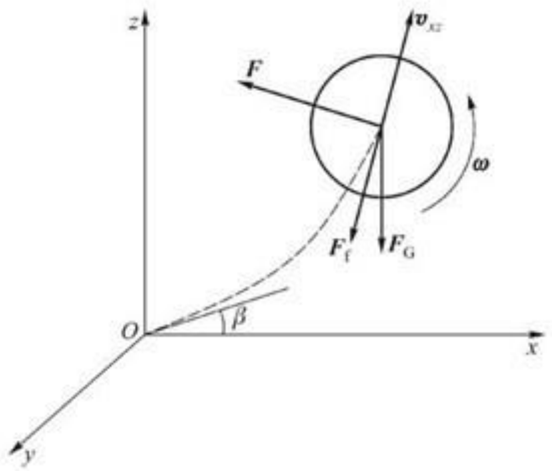
此时我们可以联立函数 x(V)与函数 V(t)从而得到函数 x(t)，这样便可以令 x(t， θ)=d 便可以得到进球时间与出球角度的关系，接下来只要对函数 y（t，θ ）的驻值条件进行讨论就可以得到位移随角度的极值。

对角度的驻值（$\partial\theta/\partial y = 0$）通过数值计算（固定$t = 1\text{s}$，位移$y \approx -10\text{m}$），得到驻值对应的角度约为$\theta \approx 53°$。

六、三维坐标系下的模型优化

6.1 模型建立

对比模型 2 我们将多一条侧边的坐标轴进行讨论，并且除了马格努斯力与空气阻力外会多一个偏转力，如图：



图表六 三维坐标下球体旋转分析

该力$\boldsymbol{F_G} = \boldsymbol{\mu}(\boldsymbol{\omega} \times \boldsymbol{v})$，接下来可以得出新的微分方程：

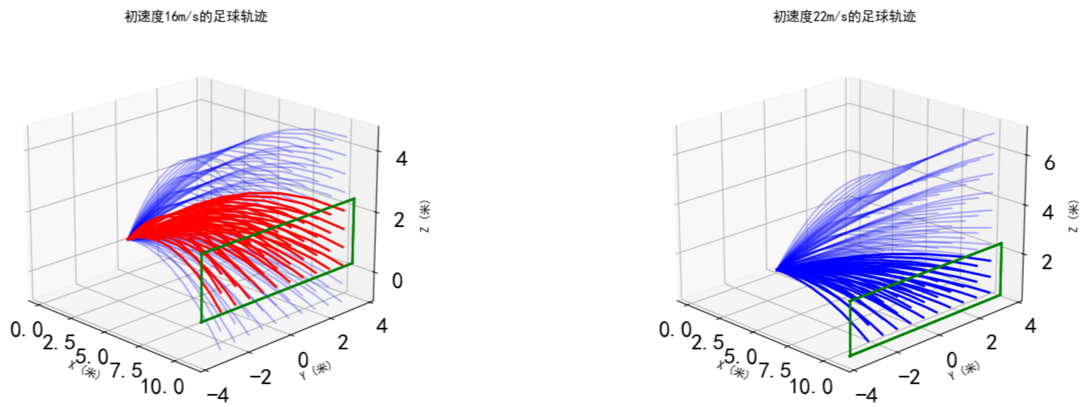
$$\boldsymbol{m}\frac{\boldsymbol{dv_x}}{\boldsymbol{dt}} = \boldsymbol{F_{d,x}} + \boldsymbol{F_{M,x}} + \boldsymbol{F_{G,x}} \tag{15}$$

$$\boldsymbol{m}\frac{\boldsymbol{dv_y}}{\boldsymbol{dt}} = \boldsymbol{F_{d,y}} + \boldsymbol{F_{M,y}} + \boldsymbol{F_{G,y}} - \boldsymbol{mg} \tag{16}$$

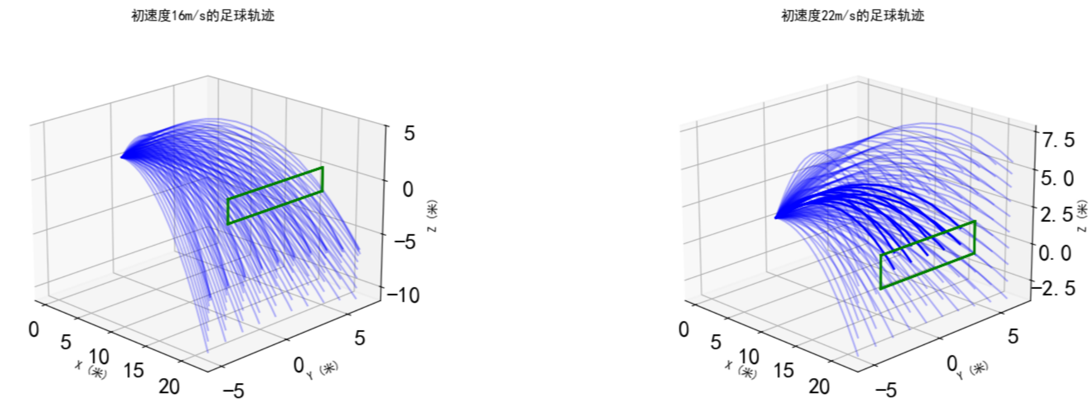
$$\boldsymbol{m}\frac{\boldsymbol{dv_x}}{\boldsymbol{dt}} = \boldsymbol{F_{d,z}} + \boldsymbol{F_{M,z}} + \boldsymbol{F_{G,z}} \tag{17}$$

6.2 模拟结果

接下来我们根据微分方程对不同的出射进行赋值，得出结果图



图表七 距离为 11m 时的射门模拟情况



图表七 距离为 22m 时的射门模拟情况

不难看出当射门距离变长后精准度大大下降，当速度增大时进球概率和精准度也会减小

模拟结果表明：旋转方向通过改变马格努斯力的方向显著影响轨迹弧度——侧旋导致侧向偏移，上旋使轨迹更“平”，下旋增加下落弧度；同时，转动动能占比与射门距离共同影响落点精度，当射门距离从 11m 增至 22m 时，轨迹偏差明显增大，精准度显著下降，验证了旋转参数和距离对足球运动轨迹的关键作用。

七、结论

首先我们利用空气阻力公式和伯努利公式推导了球体的马格努斯力公式，然后利用公式进行受力分析并且构建微分方程，然后对微分方程进行求解得到了速度函数。

根据国际足协标准对参数进行赋值给出不同的函数表达式情况，在低粘度流体假设下，建立了含空气阻力的足球二维运动方程，结合球门尺寸（高 2.2 米）

与位置约束，构建了以“进球时间最短”为目标的优化模型。

通过数值求解发现，当出射角度为 53° 时，位移对角度的变化率为临界值，最后对三维坐标系情况下模型优化和不同的射门情况进行绘图。

基于三维动力学分析，建立了考虑球体旋转的运动方程。

八、参考文献

[1] 庄礼贤 马晖扬 伊协远 流体力学[M]第二版

[2] 朗道 流体力学[M]高等教育出版社

[3] 刘孝贤编著. 足球中的物理学. 济南：山东科学技术出版社, 2009.09.

[4]高明向.虚拟环境中球形物体的动力学行为建模及其应用研究[D].武汉理工大学,2008.

九、附录

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from matplotlib import cm
import pandas as pd
plt.rcParams['font.family'] = 'SimHei'
plt.rcParams['axes.unicode_minus'] = False
plt.rcParams.update({'font.size': 20})
# 物理参数设置
m = 0.43            # 足球质量(kg)
R = 0.11            # 足球半径(m)
rho = 1.225         # 空气密度(kg/m³)
Cd = 0.4            # 阻力系数
g = 9.81            # 重力加速度(m/s²)
# 球门参数
goal_x = 22                     # 球门距离(m)
goal_width = 7.32               # 球门宽度(m)
goal_height = 2.44              # 球门高度(m)
goal_bottom = 0.3               # 球门底部高度(m)
# 旋转参数
omega_magnitude = 10
spin_direction = np.array([0, 0, 1])
def football_dynamics_3d(t, state, theta, phi):
    """
    定义三维空间中的足球运动微分方程

    参数:
        t: 时间
        state: 状态向量 [x, vx, y, vy, z, vz]
        theta: 仰角(度)
        phi: 方位角(度)
    """
    x, vx, y, vy, z, vz = state
```

```
    # 转换为弧度
    theta_rad = np.radians(theta)
    phi_rad = np.radians(phi)

    # 计算速度大小和方向
    v = np.sqrt(vx**2 + vy**2 + vz**2)
    if v < 1e-10:
        return [vx, 0, vy, 0, vz, 0]

    # 阻力系数
    k_drag = -0.5 * np.pi * R**2 * Cd * rho / m

    # 马格努斯力系数
    omega_vector = omega_magnitude * spin_direction
    v_vector = np.array([vx, vy, vz])
    cross_product = np.cross(omega_vector, v_vector)
    k_magnus = 1.0 * R**3 * rho / m    # 简化

    # 计算加速度 (dv/dt)
    ax = k_drag * v * vx + k_magnus * cross_product[0]
    ay = k_drag * v * vy + k_magnus * cross_product[1]
    az = k_drag * v * vz + k_magnus * cross_product[2] - g

    return [vx, ax, vy, ay, vz, az]
def goal_event_3d(t, state, theta, phi):
    """事件函数：当足球到达球门 x 位置时终止计算"""
    return state[0] - goal_x
goal_event_3d.terminal = True    # 到达球门后终止积分
goal_event_3d.direction = 1       # 仅检测正向穿越
def is_goal(state):
    """判断足球是否进球"""
    x, _, y, _, z, _ = state
    in_width = abs(y) <= goal_width / 2
    in_height = goal_bottom <= z <= goal_height
    return in_width and in_height and (x >= goal_x - 0.1)
def calculate_trajectory(theta, phi, v0):
    """计算给定角度和初速度下的足球轨迹"""
    # 初始状态：[x0, vx0, y0, vy0, z0, vz0]
    theta_rad = np.radians(theta)
    phi_rad = np.radians(phi)

    vx0 = v0 * np.cos(theta_rad) * np.cos(phi_rad)
    vy0 = v0 * np.cos(theta_rad) * np.sin(phi_rad)
    vz0 = v0 * np.sin(theta_rad)

    initial_state = [0, vx0, 0, vy0, 0.2, vz0]    # 高度 0.2m

    # 求解微分方程
    sol = solve_ivp(
        football_dynamics_3d, [0, 5], initial_state,
        args=(theta, phi), events=goal_event_3d,
        method='RK45', rtol=1e-8, atol=1e-8
    )

    return sol
def visualize_trajectories(results_dict):
    """3D 可视化不同初速度下的足球轨迹"""
```

```
    fig = plt.figure(figsize=(16, 12))

    # 定义颜色映射
    colors = ['red', 'blue', 'green', 'purple']
    v0_values = list(results_dict.keys())

    for i, v0 in enumerate(v0_values):
        results = results_dict[v0]
        ax = fig.add_subplot(2, 2, i+1, projection='3d')

        # 绘制进球轨迹
        goal_trajectories = [t for t in results if t['is_goal']]
        non_goal_trajectories = [t for t in results if not t['is_goal']]

        # 绘制非进球轨迹
        for traj in non_goal_trajectories:
            sol = traj['solution']
            ax.plot(sol.y[0], sol.y[2], sol.y[4], 'b-', alpha=0.3)

        # 绘制进球轨迹
        for traj in goal_trajectories:
            sol = traj['solution']
            ax.plot(sol.y[0], sol.y[2], sol.y[4], colors[i], linewidth=2)

        # 绘制球门
        goal_x_plane = goal_x
        goal_y = [-goal_width/2, goal_width/2, goal_width/2, -goal_width/2,
-goal_width/2]
        goal_z_bottom = [goal_bottom] * 5
        goal_z_top = [goal_height] * 5

        ax.plot([goal_x_plane]*5, goal_y, goal_z_bottom, 'g-', linewidth=2)
        ax.plot([goal_x_plane]*5, goal_y, goal_z_top, 'g-', linewidth=2)

        for i in range(4):
            ax.plot([goal_x_plane, goal_x_plane],
                    [goal_y[i], goal_y[i]],
                    [goal_bottom, goal_height], 'g-', linewidth=2)

        # 设置坐标轴标签和标题
        ax.set_xlabel('X (米)', fontsize=10)
        ax.set_ylabel('Y (米)', fontsize=10)
        ax.set_zlabel('Z (米)', fontsize=10)
        ax.set_title(f'初速度{v0}m/s 的足球轨迹', fontsize=12)

        # 设置视角
        ax.view_init(elev=20, azim=-45)

    plt.tight_layout()
    plt.show()
def visualize_success_rate_comparison(results_dict):
    """对比不同初速度下的进球成功率"""
    plt.figure(figsize=(12, 8))

    v0_values = list(results_dict.keys())
    success_rates = []

    for v0 in v0_values:
```

```
        results = results_dict[v0]
        total = len(results)
        success = sum(1 for r in results if r['is_goal'])
        success_rate = success / total * 100
        success_rates.append(success_rate)

        print(f"初速度 {v0}m/s 的进球成功率: {success_rate:.2f}%")

    # 绘制柱状图
    bars = plt.bar([str(v) for v in v0_values], success_rates, color='skyblue')

    # 添加数值标签
    for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2., height,
                 f'{height:.2f}%',
                 ha='center', va='bottom')

    plt.xlabel('初速度 (m/s)', fontsize=12)
    plt.ylabel('进球成功率 (%)', fontsize=12)
    plt.title('不同初速度下的进球成功率对比', fontsize=16)
    plt.grid(axis='y', linestyle='--', alpha=0.7)

    plt.tight_layout()
    plt.show()
def main():
    """主函数：执行模拟并分析结果"""
    # 要测试的初速度列表
    initial_velocities = [16, 22]  # 单位：m/s

    # 角度范围设置
    theta_min, theta_max = 10, 40  # 仰角范围(度)
    phi_min, phi_max = -15, 15     # 方位角范围(度)
    theta_steps = 10
    phi_steps = 10

    theta_values = np.linspace(theta_min, theta_max, theta_steps)
    phi_values = np.linspace(phi_min, phi_max, phi_steps)

    # 存储不同初速度的结果
    all_results = {}

    for v0 in initial_velocities:
        print(f"\n 正在计算初速度 {v0}m/s 的足球轨迹...")
        results = []
        trajectories = []

        total_calculations = theta_steps * phi_steps
        calculations_done = 0

        for theta in theta_values:
            for phi in phi_values:
                # 计算轨迹
                sol = calculate_trajectory(theta, phi, v0)

                # 检查是否进球
                final_state = sol.y[:, -1]
                success = is_goal(final_state)
```

```
                # 提取结果
                t_goal = sol.t_events[0][0] if len(sol.t_events[0]) > 0 else np.nan
                y_goal = sol.y_events[0][0][2] if len(sol.t_events[0]) > 0 else
np.nan
                z_goal = sol.y_events[0][0][4] if len(sol.t_events[0]) > 0 else
np.nan

                results.append({
                    '仰角(°)': theta,
                    '方位角(°)': phi,
                    '进球时间(s)': round(t_goal, 3),
                    '球门 Y 坐标(m)': round(y_goal, 3),
                    '球门 Z 坐标(m)': round(z_goal, 3),
                    '是否进球': success,
                    '成功率': 100 if success else 0
                })

                trajectories.append({
                    'theta': theta,
                    'phi': phi,
                    'solution': sol,
                    'is_goal': success
                })

                # 更新进度
                calculations_done += 1
                if calculations_done % 10 == 0:
                    progress = calculations_done / total_calculations * 100
                    print(f"进度: {progress:.1f}%")

        all_results[v0] = trajectories

        # 转换为 DataFrame 并打印进球情况
        df = pd.DataFrame(results)
        print(f"\n 初速度 {v0}m/s 的进球情况统计:")
        print(df[df['是否进球'] == True])

    # 可视化
    visualize_trajectories(all_results)
    visualize_success_rate_comparison(all_results)
if __name__ == "__main__":
    main()
```