```
%                                问题二：板凳龙碰撞                                %
%  (运行前请clear工作区,避免出现错误!)

% 参数设置
benches_num = 224;
head_long = 3.41;
body_long = 2.20;
benches_width = 0.3;
hole_to_head = 0.275;
p = 0.55;
v_head = 1.0;


%由问题一知前300s不会碰撞，假设再过200秒之内会碰撞
T = 500;    % 模拟时间
%缩小时间步长可以提高精度
dt = 0.01; % 时间步长

%需要实现龙在螺旋线上的动态显示，与第一问类似，不再注释
theta0 = 2*pi*16;
r0 = p*16;
positions(1, :, 1) = [r0 * cos(theta0), r0 * sin(theta0) ];
L = [head_long - 2 * hole_to_head; ...
    repmat(body_long - 2 * hole_to_head, benches_num-1, 1)];
initial_theta = theta0;
initial_r = r0;

for i = 2:benches_num
    delta_theta(i) = L(i-1) / initial_r;
    initial_theta = initial_theta + delta_theta(i);
    initial_r = p / (2 * pi) * initial_theta;
    positions(i, 1, 1) = initial_r * cos(initial_theta);
    positions(i, 2, 1) = initial_r * sin(initial_theta);
end

velocities = zeros(224, 301);
current_theta = theta0;
current_r = r0;

%初始化标志位为false
stop_signal = false;
for j = dt:dt:T
    %如果停止标志出现，立刻停止循环
    if stop_signal
        break;
    end
    t = round(j / dt);
    theta_head = current_theta - v_head * dt / current_r;
    r_head = p / (2 * pi) * theta_head;
    positions(1, :, t+1) = [r_head * cos(theta_head), ...
                r_head * sin(theta_head)];
    current_theta = theta_head;
    current_r = r_head;
    initial_theta = current_theta;
    initial_r = current_r;

    for i = 2:benches_num
        delta_theta(i) = L(i-1) / initial_r;
```

```
    initial_theta = initial_theta + delta_theta(i);
    initial_r = p / (2 * pi) * initial_theta;
    positions(i, 1, t+1) = initial_r * cos(initial_theta);
    positions(i, 2, t+1) = initial_r * sin(initial_theta);
end

if t > 0
    a = (positions(:, 1, t+1) - positions(:, 1, t)) / dt;  % vx
    b = (positions(:, 2, t+1) - positions(:, 2, t)) / dt;  % vy
    velocities(:, t+1) = sqrt(a.^2 + b.^2);
end


% %图像显示占用电脑资源影响运行速度，可以选择注释提高运行效率
% pause(0.01);
% clf;
% hold on;
% axis equal;
% xlabel('X (米)');
% ylabel('Y (米)');
% xlim([-12, 12]);
% ylim([-12, 12]);
% title(['板凳龙行进示意图 (t = ', num2str(j), 's)']);
% grid on;
% % 画背景螺线图
% theta_spiral = linspace(0, -32*pi, 10000);
% r_spiral = 0.55 * 16 + (0.55 / (2 * pi)) * theta_spiral;
% x_spiral = r_spiral .* cos(theta_spiral);
% y_spiral = r_spiral .* sin(theta_spiral);
% plot(x_spiral, y_spiral,'LineWidth', 2, 'Color', [0 0.4470 0.7410]);


%  key:考虑龙的宽度
%绘制出带有宽度的板凳龙
for i = 1:(benches_num-1)
    % 计算方向向量
    dx = positions(i+1, 1, t+1) - positions(i, 1, t+1);
    dy = positions(i+1, 2, t+1) - positions(i, 2, t+1);
    length = sqrt(dx^2 + dy^2);
    ux = -dy / length;  % 垂直方向的x分量
    uy = dx / length;   % 垂直方向的y分量

    % 计算每段线段两侧的四个顶点
    x_left1 = positions(i, 1, t+1) + ux * benches_width / 2;
    y_left1 = positions(i, 2, t+1) + uy * benches_width / 2;
    x_right1 = positions(i, 1, t+1) - ux * benches_width / 2;
    y_right1 = positions(i, 2, t+1) - uy * benches_width / 2;

    x_left2 = positions(i+1, 1, t+1) + ux * benches_width / 2;
    y_left2 = positions(i+1, 2, t+1) + uy * benches_width / 2;
    x_right2 = positions(i+1, 1, t+1) - ux * benches_width / 2;
    y_right2 = positions(i+1, 2, t+1) - uy * benches_width / 2;

    % 计算龙头方向向量
    dx = positions(2, 1, t+1) - positions(1, 1, t+1);
    dy = positions(2, 2, t+1) - positions(1, 2, t+1);
    length = sqrt(dx^2 + dy^2);
    ux = -dy / length;  % 垂直方向的x分量
```

```
    uy = dx / length;    % 垂直方向的y分量
    % 定义延长的倍数
    m = (3.41/2.86-1)/2;  % 延长1.5倍长度
    % 计算延长后的新的两个端点
    new_x1 = positions(1, 1, t+1) - m * dx;  % 起点向外延长
    new_y1 = positions(1, 2, t+1) - m * dy;
    new_x2 = positions(2, 1, t+1) + m * dx;  % 终点向外延长
    new_y2 = positions(2, 2, t+1) + m * dy;
    % 计算延长后的四个顶点
    new_x_left1 = new_x1 + ux * benches_width / 2;
    new_y_left1 = new_y1 + uy * benches_width / 2;
    new_x_right1 = new_x1 - ux * benches_width / 2;
    new_y_right1 = new_y1 - uy * benches_width / 2;

    %龙头的两个顶点
    P1_head=[new_x_left1,new_y_left1];  %龙头的起点
    P2_head=[new_x_right1,new_y_right1]; %龙头的终点

    %龙身的四个顶点
    Q1_body=[x_left1,y_left1];
    Q2_body=[x_right1,y_right1];
    Q3_body=[x_left2,y_left2];
    Q4_body=[x_right2,y_right2];

    new_x_left2 = new_x2 + ux * benches_width / 2;
    new_y_left2 = new_y2 + uy * benches_width / 2;
    new_x_right2 = new_x2 - ux * benches_width / 2;
    new_y_right2 = new_y2 - uy * benches_width / 2;

    if check_intersection(P1_head, P2_head, Q1_body, Q2_body) || ...
       check_intersection(P1_head, P2_head, Q2_body, Q4_body) || ...
       check_intersection(P1_head, P2_head, Q4_body, Q3_body) || ...
       check_intersection(P1_head, P2_head, Q3_body, Q1_body)

        disp(['龙头线段与龙身矩形相交，停止模拟 (t = ' num2str(j) 's)']);
        stop_signal = true;  % 设置标志
        break;
    end


    % %图像显示占用电脑资源影响运行速度，可以选择注释提高运行效率
    % % 使用 patch 函数绘制每段的宽线段
    % patch([x_left1, x_left2, x_right2, x_right1], ...
    % [y_left1, y_left2, y_right2, y_right1], 'b');


end

% %图像显示占用电脑资源影响运行速度，可以选择注释提高运行效率
% % 绘制延长后的宽线段
% patch([new_x_left1, new_x_left2, new_x_right2, new_x_right1], ...
%     [new_y_left1, new_y_left2, new_y_right2, new_y_right1], 'r');
% %画龙
% plot(positions(1, 1, t+1), positions(1, 2, t+1), 'ro-', ...
%     'MarkerSize', 0.3, 'LineWidth', 1,'MarkerFaceColor', 'r');
% plot(positions(2:end, 1, t+1), positions(2:end, 2, t+1), 'go-', ...
%     'MarkerSize', 3, 'LineWidth', 1,'MarkerFaceColor', 'r');
% hold off;
```

```
    disp(['当前运行' num2str(j) 's']);
end

%保存结果到Excel文件
writematrix(positions(:, 1,end),'result2.xlsx', 'Range', 'B2:B225');%x数据
writematrix(positions(:, 2,end),'result2.xlsx', 'Range', 'C2:C225');%y数据
writematrix(velocities(:,end),'result2.xlsx', 'Range', 'D2:D225');  %v数据
disp('数据已存入result2.xlsx');
```