方法一：浦丰投针

```
function pi_estimate = buffons_needle_simulation(n)
% n: 模拟次数
a = 1; % 平行线之间的距离
l = 0.5; % 针的长度
m = 0; % 与平行线相交的次数

for i = 1:n
    % 随机位置和角度
    x = rand() * (a / 2); % 0 ~ a/2
    theta = rand() *(pi / 2); % 0 ~ pi/2

    % 判断是否与平行线相交
    if x <= l/2 * sin(theta)
        m = m + 1;
    end
end

% 估计 π
pi_estimate = (2 * l * n) / (a * m);
end

% 测试
pi_estimate = buffons_needle_simulation(100000000);
disp(pi_estimate);

% 输出:
>> hw1
    3.1415
```

方法二：蒙特卡罗方法

```
function pi_estimate = montecarlo_simulation(n)
% n: 模拟次数
inside = 0;

for i = 1:n
    % 在 [-1, 1]x[-1, 1] 的正方形内随机生成点
    x = 2 * rand() - 1;
    y = 2 * rand() - 1;

    % 判断是否在单位圆内
```

```
        if x^2 + y^2 <= 1
            inside = inside + 1;
        end
    end

    % 估计 π
    pi_estimate = 4 * inside / n;
    end

    % 测试
    pi_estimate = montecarlo_simulation(100000000);
    disp(pi_estimate);

    % 输出
    > hw1
        3.1418
```