



组 号 Z08

实验题目 用精确法，数值法与仿真法求解追及问题

1. 宋昭琰

队员姓名 2. 徐意

3. 李宗琳

4. 向飞宇

目 录

1 符号说明	2
2 狗追兔子的微分方程模型	3
2.1 利用运动学方程建立模型	3
2.1.1 初始模型	3
2.1.2 旋转坐标系的改进模型	3
2.2 用微分方程的精确解, 数值解与仿真方法对模型求解	5
2.2.1 精确解求解模型	5
2.2.2 数值解求解模型	6
2.2.3 仿真方法求解模型	8
2.2.4 三种方法的比较	9
2.3 求解速度与时间相关的模型	10
2.3.1 对实验方法的选取	10
2.3.2 对速度的预处理	10
2.3.3 仿真方法求解模型	11
3 多人相互追及问题	12
3.1 利用仿真方法求解四人在正方形上的追及问题	12
3.2 利用仿真方法求解多人在正方形上的追及问题	14
3.3 利用精确解求解四人在正方形上的追及问题	15
3.4 利用精确解求解多人在正方形上的追及问题	17
3.5 仿真方法的误差分析	19
4 实验总结	19
附录 A 实验代码	20
A.1 狗追兔子精确求解代码	20
A.2 狗追兔子数值求解代码	20
A.3 狗追兔子仿真方法求解代码-1	21
A.4 狗追兔子仿真方法求解代码-2	22
A.5 多人追及问题求解代码	23
A.6 多人追及问题精确解作图代码	24

1 符号说明

符号	意义
x_d	原坐标系下狗的 x 轴坐标
y_d	原坐标系下狗的 y 轴坐标
x_r	原坐标系下兔子的 x 轴坐标
y_r	原坐标系下兔子的 y 轴坐标
x'_d	旋转后坐标系下狗的 x 轴坐标
y'_d	旋转后坐标系下狗的 y 轴坐标
x'_r	旋转后坐标系下兔子的 x 轴坐标
y'_r	旋转后坐标系下兔子的 y 轴坐标
v_d	狗的速度
t_d	狗追上兔子的时间
s_d	狗的运动路程
ε	判断狗与兔子是否相遇的临界值
x_i	第 i 个人的 x 轴坐标 ($i = 1, 2, \dots, n$)
y_i	第 i 个人的 y 轴坐标 ($i = 1, 2, \dots, n$)
t	各人相遇的时间
s	各人相遇时的运动路程
t_0	仿真方法中所选取的一个时刻
Δt	仿真方法中的时间步长

2 狗追兔子的微分方程模型

2.1 利用运动学方程建立模型

2.1.1 初始模型

首先, 根据运动学方程与追及方向的限制, 我们给出了初始模型需要满足的条件.

设时刻 t , 狗的坐标为 $D(x_d, y_d)$, 兔子的坐标为 $R(x_r, y_r)$. 由兔子的运动方向及运动速度, 将其分别向 x, y 轴投影, 可得:

$$\begin{aligned} x_r &= 8 \cos\left(-\frac{\pi}{4}\right)t = -4\sqrt{2}t \\ y_r &= 8 \sin\left(-\frac{\pi}{4}\right)t = 4\sqrt{2}t \end{aligned}$$

由于狗沿着兔子运动的方向追及兔子, 根据狗与兔子的位置可知狗的运动方向, 从而列出如下微分方程:

$$\frac{dy}{dx} = \frac{y - 4\sqrt{2}t}{x + 4\sqrt{2}t}$$

再假设狗的运动路程为 s_d , 狗的速度为 v_d , 则由弧长公式可知:

$$\frac{dt}{dx} = \frac{dt}{ds_d} \cdot \frac{ds_d}{dx} = -\frac{1}{v_d} \sqrt{1 + \left(\frac{dy}{dx}\right)^2}$$

由于狗的初始方向与旋转后 x 轴夹角为 45° , 初始坐标为 $(-100\sqrt{2}, -100\sqrt{2})$, 对于微分方程也存在初始条件:

$$y_d|_{x=-100\sqrt{2}} = -100\sqrt{2}, \quad \frac{dy_d}{dx_d}|_{x=-100\sqrt{2}} = \tan\left(\frac{\pi}{4}\right) = 1$$

狗追上兔子的时间 t_d 需要小于兔子跑进洞口的时间, 即 $t_d \leq \frac{120}{8} = 15(s)$.

综上所述, 我们总结出初始模型如下:

$$\begin{cases} \frac{dy_d}{dx_d} = \frac{y_d - 4\sqrt{2}t}{x_d + 4\sqrt{2}t} \\ \frac{dt}{dx_d} = -\frac{1}{v_d} \frac{ds_d}{dx_d} \\ \frac{ds_d}{dx_d} = \sqrt{1 + \left(\frac{dy_d}{dx_d}\right)^2} \\ y_d|_{x_d=-100\sqrt{2}} = -100\sqrt{2} \\ \frac{dy_d}{dx_d}|_{x_d=-100\sqrt{2}} = 1 \\ t_d = \frac{s_d}{v_d} \leq 15 \end{cases} \quad (1)$$

2.1.2 旋转坐标系的改进模型

方程 (1) 的降阶与求解较为复杂, 不适合使用精确求解和数值求解解决, 其主要原因在于 $\frac{dy}{dx} = \frac{y - 4\sqrt{2}t}{x + 4\sqrt{2}t}$ 中分母项存在与时间 t 相关的项, 且无法被代换方式消去. 我们观察

到兔子与狗的原始方向垂直, 而通过旋转坐标轴就可以将狗的初始坐标转换至 x 轴, 这样分母项中就不存在与时间 t 相关的项, 所以如下图所示, 可以将坐标系逆时针旋转 45° 度:

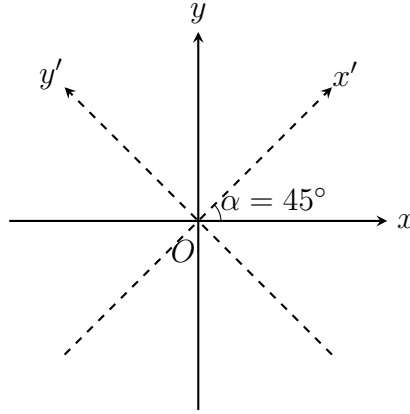


图 1 旋转坐标系示意图

根据这种方式得到一个改进的模型:

$$\frac{dy'_d}{dx'_d} = \frac{y'_d - 8t}{x'_d} \quad (2)$$

$$\frac{dt}{dx'_d} = \frac{1}{v_d} \frac{ds_d}{dx'_d} \quad (3)$$

$$\frac{ds_d}{dx'_d} = \sqrt{1 + \left(\frac{dy'_d}{dx'_d}\right)^2} \quad (4)$$

$$y'_d|_{x'_d=-200} = 0 \quad (5)$$

$$\frac{dy'_d}{dx'_d}|_{x'_d=-200} = 0 \quad (6)$$

$$t_d = \frac{s_d}{v_d} \leq 15 \quad (7)$$

这个方程与初始模型的区别在于:

1. 由于仅仅旋转了坐标系, 改进方程中的改变量只有初始坐标与投影在坐标轴上的速度分量, 而这些改变能够简化计算.

2. $\frac{dy'_d}{dx'_d} = \frac{y'_d - 8t}{x'_d}$ 的分母项中不包含与时间 t 相关的项, 所以能够求得其精确解, 并能对方程的结构进行分析.

2.2 用微分方程的精确解, 数值解与仿真方法对模型求解

2.2.1 精确解求解模型

首先我们来求解方程 (2) 的精确解, 先对其化简:

$$\begin{aligned}\frac{dy'_d}{dx'_d} &= \frac{y'_d - 8t}{x'_d} \\ x'_d \frac{dy'_d}{dx'_d} &= y'_d - 8t\end{aligned}\quad (8)$$

对于方程 (8), 两边关于 x_d 求导, 可得:

$$x'_d \frac{d^2 y'_d}{dx'^2_d} = -8 \frac{dt}{dx'_d} \quad (9)$$

将方程 (3), (4) 代入 (9), 可得:

$$x'_d \frac{d^2 y'_d}{dx'^2_d} = -\frac{8}{v_d} \sqrt{1 + \left(\frac{dy'_d}{dx'_d}\right)^2} \quad (10)$$

令 $p = \frac{dy'_d}{dx'_d}$, 则 $\frac{dp}{dx'_d} = \frac{d^2 y'_d}{dx'^2_d}$. 根据原方程的初始条件可得 $p|_{x=-200} = 0$. 通过这个代换可以将方程 (10) 转换为:

$$x'_d \frac{dp}{dx'_d} = -\frac{8}{v_d} \sqrt{1 + p^2} \quad (11)$$

通过将方程 (11) 移项并两边积分, 可得:

$$\begin{aligned}\int \frac{dp}{\sqrt{1 + p^2}} &= -\frac{8}{v_d} \int \frac{dx'_d}{x'_d} \\ p|_{x'_d=-200} &= 0\end{aligned}\quad (12)$$

解方程 (12) 可得:

$$\begin{aligned}p + \sqrt{1 + p^2} &= \left(-\frac{x'_d}{200}\right)^{-\frac{8}{v_d}} \\ p - \sqrt{1 + p^2} &= -\left(-\frac{200}{x'_d}\right)^{-\frac{8}{v_d}}\end{aligned}$$

由上式可以得到:

$$\frac{dy'_d}{dx'_d} = \frac{1}{2} \left[\left(-\frac{x'_d}{200}\right)^{-\frac{8}{v_d}} - \left(-\frac{200}{x'_d}\right)^{-\frac{8}{v_d}} \right] \quad (13)$$

方程 (13) 具有初始条件 $y'_d|_{x'_d=-200} = 0$, 结合初始条件对其积分, 可得其解为:

$$y'_d = \frac{x'_d}{2} \left[\frac{\left(-\frac{200}{x'_d}\right)^{-\frac{8}{v_d}}}{-\frac{8}{v_d} - 1} + \frac{\left(-\frac{x'_d}{200}\right)^{-\frac{8}{v_d}}}{-\frac{8}{v_d} + 1} \right] + 100 \cdot \frac{-\frac{16}{v_d}}{\left(\frac{8}{v_d}\right)^2 - 1} \quad (14)$$

如果不考虑兔子进入兔子洞, 在 $x = 0$ 时, 狗将追上兔子, 此时兔子奔跑的距离为:

$$s_r = y'_d|_{x=0} = 100 \cdot \frac{-\frac{16}{v_d}}{\left(\frac{8}{v_d}\right)^2 - 1} \quad (15)$$

根据兔子与狗运动时间的相等性, 可以得到狗追上兔子的时间:

$$t_d = \frac{s_r}{8} \quad (16)$$

由模型中的时间限制 (7) 可得以下不等式:

$$\frac{\frac{8}{v_d}}{1 - (\frac{8}{v_d})^2} \leq \frac{3}{5} \quad (17)$$

由于狗追上兔子至少需要满足狗的速度大于兔子的速度, 即 $v_d > 8$, 而在 $v_d > 8$ 时, 由于不等式左端关于 v_d 单调递减, 只需求出其临界值即可, 求解得:

$$\min v_d = \frac{4}{3}(\sqrt{61} + 5) \approx 17.0803 \quad (18)$$

狗跑过的路程可由以下积分式求得, 其中 $\frac{dy'_d}{dx'_d}$ 的值由方程 (13) 给出:

$$\begin{aligned} s_d &= \int_{-200}^0 \sqrt{1 + \left(\frac{dy'_d}{dx'_d}\right)^2} dx'_d \\ &= \int_{-200}^0 \sqrt{\left(\frac{1}{2}\left[-\frac{x'_d}{200}\right]^{-\frac{8}{v_d}} + \left(-\frac{200}{x'_d}\right)^{-\frac{8}{v_d}}\right)^2} dx'_d \\ &= \int_{-200}^0 \frac{1}{2}\left[\left(-\frac{x'_d}{200}\right)^{-\frac{8}{v_d}} + \left(-\frac{200}{x'_d}\right)^{-\frac{8}{v_d}}\right] dx'_d \\ &= \left\{\frac{x'_d}{2}\left[\frac{\left(-\frac{200}{x'_d}\right)^{-\frac{8}{v_d}}}{\frac{8}{v_d} + 1} + \frac{\left(-\frac{x'_d}{200}\right)^{-\frac{8}{v_d}}}{-\frac{8}{v_d} + 1}\right]\right\}\bigg|_{-200}^0 \approx 256.2050 \end{aligned}$$

综上所述, 狗的速度至少为 17.0803m/s , 此时狗跑过的路程为 256.2050m . 根据所求的数据可以做出以下的奔跑轨迹图:

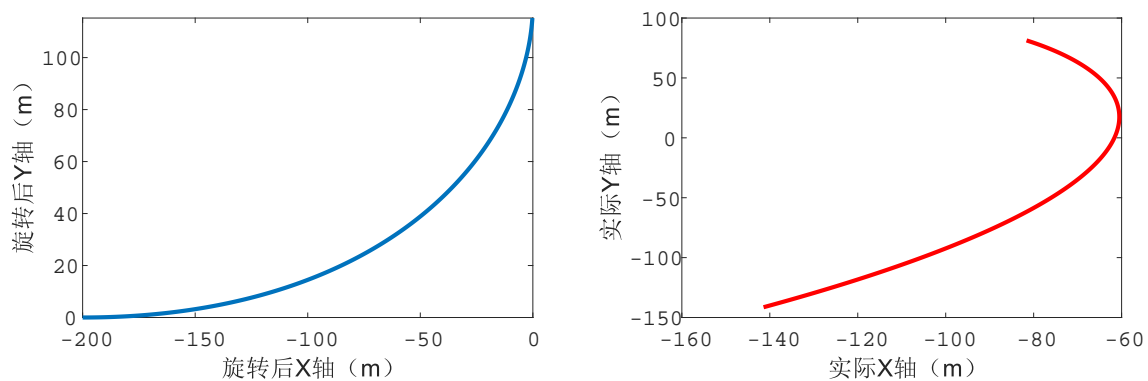


图2 狗的奔跑轨迹图

2.2.2 数值解求解模型

虽然通过精确求解方程可以得出方程最为精确的解, 但是这种方式需要花费较多人工计算时间, 而且旋转坐标系的方法只是针对这一问题而言的特殊方法。所以在这一部

分中, 我们采用数值求解的手段, 不旋转坐标系, 直接重新求解初始模型, 从而为更普遍的追及情况打下基础.

对初始模型进行转换:

观察方程 (1), 考虑到 4-5 阶 Rugge-Kutta 方法更适合于求解一阶常微分方程组, 不妨将原方程作如下转换:

$$\begin{aligned}\frac{dx_d}{dt} &= v_d / \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \\ \frac{dy_d}{dt} &= \frac{dy_d}{dx_d} \cdot \frac{dx_d}{dt} \\ \frac{dy_d}{dx_d} &= \frac{y - 4\sqrt{2}t}{x + 4\sqrt{2}t}\end{aligned}\tag{19}$$

通过这样的转换, 就避免了求解高阶微分方程, 从而能够调用 MatLab 中的 ode45 函数求解方程.

根据实际物理背景对转化后的方程进行调整:

方程 (19) 方便了求解, 但是存在一个问题: 其中的 x_d , y_d 是不存在方向的, 但是很有可能出现狗在某一个坐标轴上的速度投影由正值变化为负值, 而方程中并不能体现出这一点, 这样就会使求解时的某些速度分量与实际上的速度分量反向. 对于这个问题, 我们进行了如下分析:

首先分析兔子的运动, 兔子向正西北方向运动, 其 x 轴速度分量为负, y 轴速度分量为正, 而且大小不随时间改变. 然后分析狗的运动, 初始状态其 x 轴速度分量为正, y 轴速度分量为正. 如果狗要追上兔子, 由于其速度是连续变化的, 其 x 轴速度分量必然经过了从正到负的过程, 所以需要调整狗在 x 轴的速度分量. 因为 y 轴速度分量一直为正, 所以不需要对狗在 y 轴的速度分量进行调整. 我们的调整基于 (20):

$$\frac{dy_d}{dx_d} = \frac{y_d - 4\sqrt{2}t}{x_d + 4\sqrt{2}t}\tag{20}$$

由于 Rugge-Kutta 方法数值求解方程使用了迭代的思想, 可以假定每一步迭代中 $\Delta y_d > 0$, 这样 (20) 的正负性就与 Δx_d 的正负性等价, 从而可将 (19) 中第一个式子写成分段函数的形式:

$$\frac{dx_d}{dt} = \begin{cases} v_d / \sqrt{1 + \left(\frac{dy}{dx}\right)^2} & \text{若 } \frac{y - 4\sqrt{2}t}{x + 4\sqrt{2}t} \geq 0 \\ -v_d / \sqrt{1 + \left(\frac{dy}{dx}\right)^2} & \text{若 } \frac{y - 4\sqrt{2}t}{x + 4\sqrt{2}t} < 0 \end{cases}$$

用数值求解方法得出模型的解

我们所求解的方程是狗的坐标与时间的关系, 并没有涉及到“狗追上兔子”这一行为, 所以需要使用它们之间的距离给出这个行为的数学定义:

$$\sqrt{(x_d - x_r)^2 + (y_d - y_r)^2} < \varepsilon\tag{21}$$

其中 ε 是一个固定的小值, 在本次实验中设 $\varepsilon = 10^{-6}(m)$.

在模型的求解中, 基于狗追兔子的最长容许时间, 给定时间区间 $t \in [0, 15]$, 在这个区间内使用 ode45 求解问题, 并采取了以下的停止计算策略:

1. 在 Runge-Kutta 方法的每一步迭代中通过 (21) 进行相对距离检测, 如果相对距离符合条件, 则停止计算.
2. 判定追及成功: 如果在时间区间内停止计算, 则追及成功, 否则追及失败.
3. 逐步搜索速度: 首先以 1 秒为步长, 递增试探最小速度区间; 然后以 0.1 毫秒为步长, 精细搜索得出最小速度.

根据实验得出最小速度区间为 $[17, 18]m/s$, 精细搜索得出的最小速度为 $17.1489m/s$. 将每一步迭代(共 2121 步)所得的坐标用直线相连, 求出这一段折线的长度为 $257.3020m$, 由于迭代步数足够大, 可以将追及的曲线长度看作折线长度求解, 所以狗跑过的路程为 $257.3020m$. 此时狗的运动轨迹如图3所示:

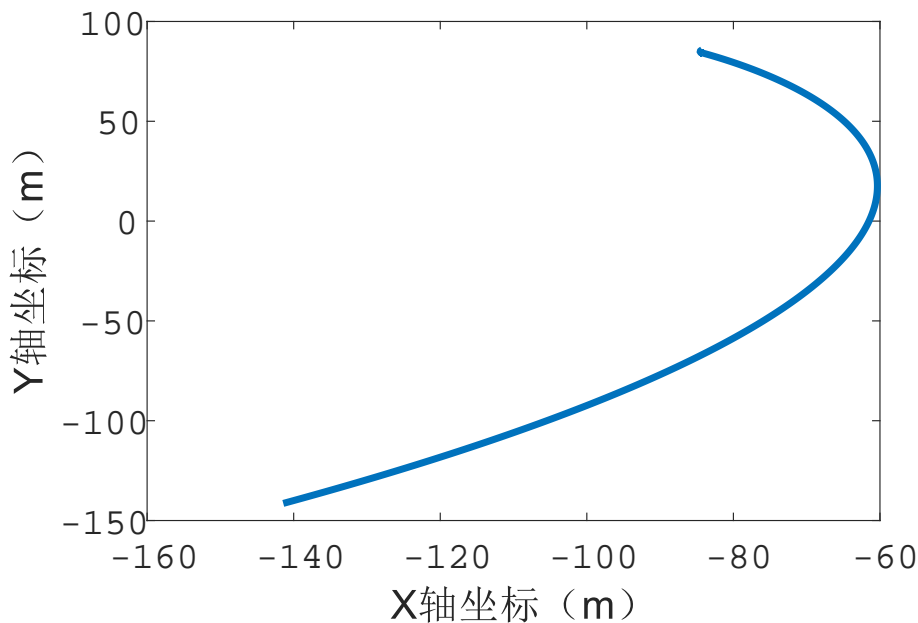


图3 数值计算下狗的运动轨迹

2.2.3 仿真方法求解模型

仿真方法是编程最为简单的一种方法, 只需要狗与兔子前一时刻的运动状态就能够得出它们后一时刻的运动状态, 重复这一过程直至它们相遇即可. 对于仿真方法不需要求解微分方程, 只需要对方程进行一些修改以适合于使用仿真方法即可. 首先, 设时间步长为 Δt , 初始状态为 $x_d|_{t=0} = -100\sqrt{2}, y_d|_{t=0} = -100\sqrt{2}$. 将狗的运动方向分别向 x, y

轴作投影可得, 在时刻 t_0 与一个时间步长 Δt 内, 有以下方程:

$$\begin{aligned} x_d|_{t=t_0+\Delta t} &= x_d|_{t=t_0} + v_d \Delta t \frac{x_r|_{t=t_0} - x_d|_{t=t_0}}{\sqrt{(x_r|_{t=t_0} - x_d|_{t=t_0})^2 + (y_r|_{t=t_0} - y_d|_{t=t_0})^2}} \\ y_d|_{t=t_0+\Delta t} &= y_d|_{t=t_0} + v_d \Delta t \frac{y_r|_{t=t_0} - y_d|_{t=t_0}}{\sqrt{(x_r|_{t=t_0} - x_d|_{t=t_0})^2 + (y_r|_{t=t_0} - y_d|_{t=t_0})^2}} \end{aligned} \quad (22)$$

与此同时, 在 t_0 时刻兔子的坐标为:

$$\begin{aligned} x_r|_{t=t_0} &= -4\sqrt{2}t_0 \\ y_r|_{t=t_0} &= 4\sqrt{2}t_0 \end{aligned} \quad (23)$$

与数值计算类似, 在用仿真方法求解模型时, 也需要 (21) 来检查狗是否追上了兔子. 一旦满足了这个条件, 则停止计算, 并记录经过的时间步长的总数 T , 若 $T \leq 15$, 则狗能够以该速度追上兔子. 在对速度的搜索中, 仍先以 1 秒为步长, 递增试探最小速度区间, 然后以 0.1 毫秒为步长, 精细搜索得出最小速度.

根据实验得出最小速度区间为 $[17, 18]m/s$, 精细搜索得出的最小速度为 $17.0952m/s$. 将每一步迭代所得的坐标用直线相连, 求出这一段折线的长度为 $256.4280m$, 由于迭代步数足够大, 可以将追及的曲线长度看作折线长度求解, 所以狗跑过的路程为 $256.4280m$. 此时狗的运动轨迹如下图所示:

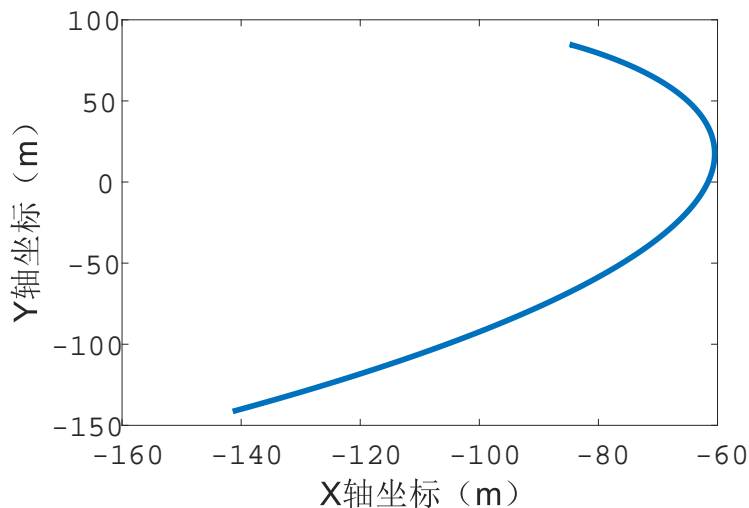


图 4 仿真方法求解狗的运动轨迹

2.2.4 三种方法的比较

我们分别对求解难度, 运算时间, 结果的准确性三个方面对着三种方法进行了一个比较, 以下是比较的结果:

项目	精确解法	数值解法	仿真解法
求解难度	较难	中等	较容易
运算时间	不需要计算机运算	较快	精度与时间称反比
结果的准确性	准确	较高	精度与时间称反比
方法瓶颈	方法不适用于所有情况	难以运算高阶方程	时间与运算精度的平衡

表 1 三种求解方法的比较

综上所述, 精确解法仅对特定问题有效, 对于夹角不为直角的问题, 旋转坐标系也不能解决, 而且对于微分方程的求解也较为复杂; 数值解法对大部分的追及问题都有效, 但是由于时间的限制, 在寻找最优解的问题上较为复杂; 仿真解法对所有的追及问题都有效, 因为它模拟了物体的运动过程, 很容易就能够求出一个解, 但是取不同的步长将会得到不同的解, 步长越小, 解越精确, 但消耗的时间也越长, 在本次实验中, 数值解法与仿真解法消耗的时间相近, 但仿真解法的效果更好.

2.3 求解速度与时间相关的模型

2.3.1 对实验方法的选取

本题与之前的模型最大的区别在于, 兔子的速度不再是线性的, 这意味着它的坐标与时间不线性相关. 所以不能通过兔子的速度推出它进洞的时间, 这对精确求解造成了困难. 同时本题中, 由于速度与时间相关, 对于原模型的求解变为:

$$\frac{dy_d}{dx_d} = \frac{x_d - \frac{\sqrt{2}}{2}v_r(t) \cdot t}{y_d + \frac{\sqrt{2}}{2}v_r(t) \cdot t} \quad (24)$$

此时该方程将无法求出精确解, 所以精确解法将不适用于下面的实验.

对于数值解法而言, 方程 (24) 能够被数值求解, 但是对于兔子进洞的时间判定, 狗与兔子的距离对它们速度的影响并不好用一个独立的公式表达, 所以采用数值求解方法也不太合理.

本题使用仿真方法较为合理, 通过位置随时间步长的更新, 仿真方法既能判断狗与兔子的距离, 也能处理兔子逃跑的距离.

2.3.2 对速度的预处理

根据题目, 在狗与兔子的距离小于 30 米时, 狗的速度每秒增加 0.1 倍, 兔子的速度每秒减半. 但如果以秒作为时间区间, 使用仿真方法无法得到很好的结果, 所以我们对

时间进行了预处理. 最适合每秒加倍与减半的函数是指数函数, 根据题意, 假定在 t_1 时刻狗与兔子的距离小于 30 米, 那么它们的速度均可以表示成以下形式:

$$v = \begin{cases} v_0 & t < t_1 \\ v_0 b^{t-t_1} & t \geq t_1 \end{cases} \quad (25)$$

对于狗, $v_0 = v_d, b = 1.1$; 对于兔子, $v_0 = 8, b = 0.5$. 这样就可以得到狗与兔子任意时间的速度, 从而方便以下的计算过程.

2.3.3 仿真方法求解模型

此处的求解方法与之前类似, 由于精确性的要求, 在这一部分中将时间步长取为 $10^{-4}s$, 由于计算时间大大增加, 需要使用效率更高的算法. 考虑到狗的速度越快, 它追上兔子的时间越短, 兔子逃离的距离就越短, 这满足**单调性**, 所以在精细搜索最小速度时可以采用**二分法**提升效率. 与此同时, 粗略搜索的时间区间仍为 $1s$.

通过实验得出最小速度区间为 $[15, 16]m/s$, 以下是二分法的计算过程 (共进行了 10 次计算, 部分过程省略):

当前速度区间 (m/s)	当前计算速度 (m/s)	结果	下一速度区间 (m/s)
$[15, 16]$	15.5	成功	$[15, 15.5]$
$[15, 15.5]$	15.25	失败	$[15.25, 15.5]$
$[15.25, 15.5]$	15.375	失败	$[15.4375, 15.5]$
$[15.4375, 15.5]$	15.4688	成功	$[15.4375, 15.4688]$
...
$[15.4414, 15.4415]$	15.4415	成功, 终止计算	

表 2 二分法的求解过程

综上所述, 狗能够追上兔子的最小速度为 $15.4415m/s$, 按照相同的方法求得此时狗跑过的路程为 $252.1541m$, 此时狗的运动轨迹如下图所示:

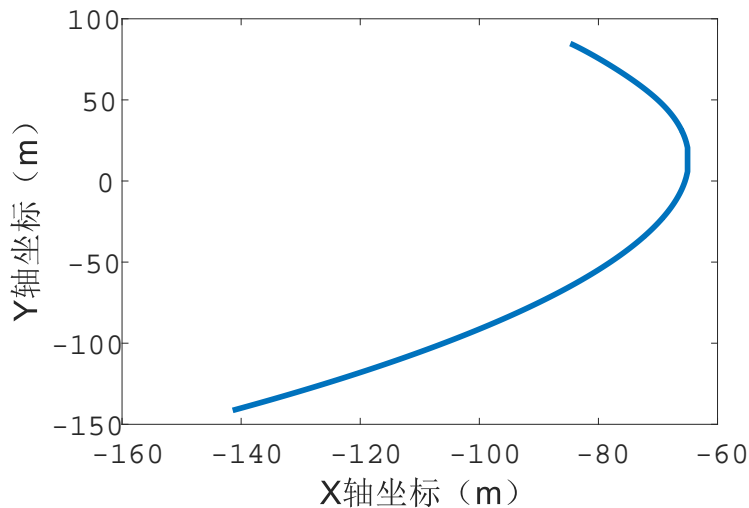


图5 速度随时间变化情况下狗的运动轨迹

3 多人相互追及问题

3.1 利用仿真方法求解四人在正方形上的追及问题

首先, 为了方便下一节中的推广, 我们将正方形抽象成为圆内接多边形的问题, 在这种假定下, 正方形的四个点在坐标系中的表示按顺时针分别为 $(r, 0)$, $(0, -r)$, $(-r, 0)$, $(0, r)$, 显然满足 $a = \sqrt{2}r$, 其中 a 是原正方形的边长, 如下图所示:

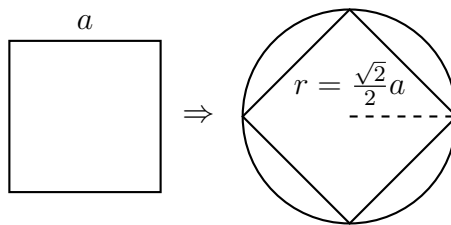


图6 圆内接正方形转换示意图

对于四人的追及问题, 初始状态对应正方形的四个顶点. 设时间步长为 Δt , 在时刻 t_0 与一个时间步长 Δt 内, 假设四个人在 $t = t_0$ 时的坐标分别为 (x_i, y_i) , $i = 1, 2, 3, 4$, 根据他们的运动方向有下方程:

$$\begin{aligned} x_1|_{t=t_0+\Delta t} &= x_1|_{t=t_0} + v_1 \Delta t \frac{x_2|_{t=t_0} - x_1|_{t=t_0}}{\sqrt{(x_1|_{t=t_0} - x_2|_{t=t_0})^2 + (y_1|_{t=t_0} - y_2|_{t=t_0})^2}} \\ y_1|_{t=t_0+\Delta t} &= y_1|_{t=t_0} + v_1 \Delta t \frac{y_2|_{t=t_0} - y_1|_{t=t_0}}{\sqrt{(x_1|_{t=t_0} - x_2|_{t=t_0})^2 + (y_1|_{t=t_0} - y_2|_{t=t_0})^2}} \\ x_2|_{t=t_0+\Delta t} &= x_2|_{t=t_0} + v_2 \Delta t \frac{x_3|_{t=t_0} - x_2|_{t=t_0}}{\sqrt{(x_2|_{t=t_0} - x_3|_{t=t_0})^2 + (y_2|_{t=t_0} - y_3|_{t=t_0})^2}} \\ y_2|_{t=t_0+\Delta t} &= y_2|_{t=t_0} + v_2 \Delta t \frac{y_3|_{t=t_0} - y_2|_{t=t_0}}{\sqrt{(x_2|_{t=t_0} - x_3|_{t=t_0})^2 + (y_2|_{t=t_0} - y_3|_{t=t_0})^2}} \end{aligned}$$

$$\begin{aligned}
 x_3|_{t=t_0+\Delta t} &= x_3|_{t=t_0} + v_3 \Delta t \frac{x_4|_{t=t_0} - x_3|_{t=t_0}}{\sqrt{(x_3|_{t=t_0} - x_4|_{t=t_0})^2 + (y_3|_{t=t_0} - y_4|_{t=t_0})^2}} \\
 y_3|_{t=t_0+\Delta t} &= y_3|_{t=t_0} + v_3 \Delta t \frac{y_4|_{t=t_0} - y_3|_{t=t_0}}{\sqrt{(x_3|_{t=t_0} - x_4|_{t=t_0})^2 + (y_3|_{t=t_0} - y_4|_{t=t_0})^2}} \\
 x_4|_{t=t_0+\Delta t} &= x_4|_{t=t_0} + v_4 \Delta t \frac{x_1|_{t=t_0} - x_4|_{t=t_0}}{\sqrt{(x_4|_{t=t_0} - x_1|_{t=t_0})^2 + (y_4|_{t=t_0} - y_1|_{t=t_0})^2}} \\
 y_4|_{t=t_0+\Delta t} &= y_4|_{t=t_0} + v_4 \Delta t \frac{y_1|_{t=t_0} - y_4|_{t=t_0}}{\sqrt{(x_4|_{t=t_0} - x_1|_{t=t_0})^2 + (y_4|_{t=t_0} - y_1|_{t=t_0})^2}}
 \end{aligned}$$

这些方程数目巨大, n 个人就会有 $2n$ 个方程, 在本节中我们先对其求解, 下一节中再给出它们的统一形式. 为了研究更多的运动形式, 我们对可能出现的情况进行了分类并依次处理.

针对四人速度大小分类:

在四人速度相等的条件下, 通过调整速度与半径的比值进行试验. 为了让结果更加明显, 我们选取了 $\frac{r}{v} = 0.01, 1, 100$ 的三种情况作出下图, 下图中从左至右分别对应比值为 $0.01, 1, 100$:

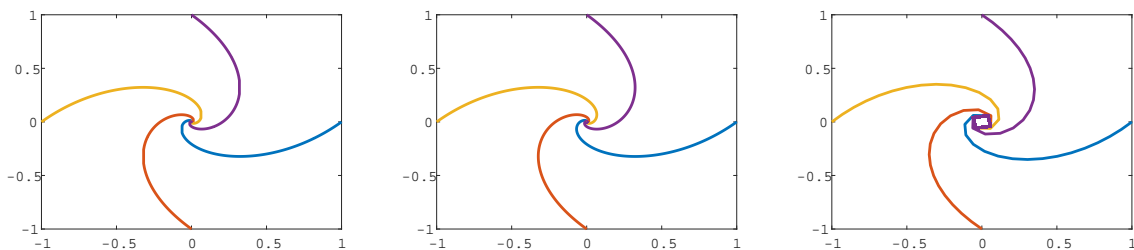


图 7 不同速度与半径比值下四人的运动轨迹

从以上的实验中, 我们可以看出, 在速度与半径比值较小的情况下, 四人的轨迹基本不会改变 (只是四人相遇的时间发生了改变), 但是在速度与半径的比值较大的情况下, 四人最终可能不会相遇, 只是在一个半径较小的正方形内运动. 因为速度较大时, 即使在一个很小的时间步长下, 运动的距离也会较大, 所以可能会造成四人“偏离”了相遇的轨道, 进而在一个半径较小的区域内运动. 这一点将在对仿真方法的误差分析中再次被提到.

针对四人相对速度大小分类:

研究四人速度成不同比例时的各人的运动轨迹, 我们选取了四种不同的比例, $1:1:1:1$ (左上), $1:3:5:7$ (右上), $1:3:1:3$ (左下), $1:1:1:10$ (右下), 作出四人的运动轨迹图.

根据不同速度比例得出四人各自的轨迹虽然各有不同, 但是它们出现了“提前追及”的情况: 即速度最快的人追上了速度最慢的人. 在此之后, 速度快的人与速度慢的人的运动轨迹几乎重合, 只是速度快的人可能在这条轨迹上来回奔跑, 因为对这两人的系统而言, 追及问题此时转变为相遇问题.

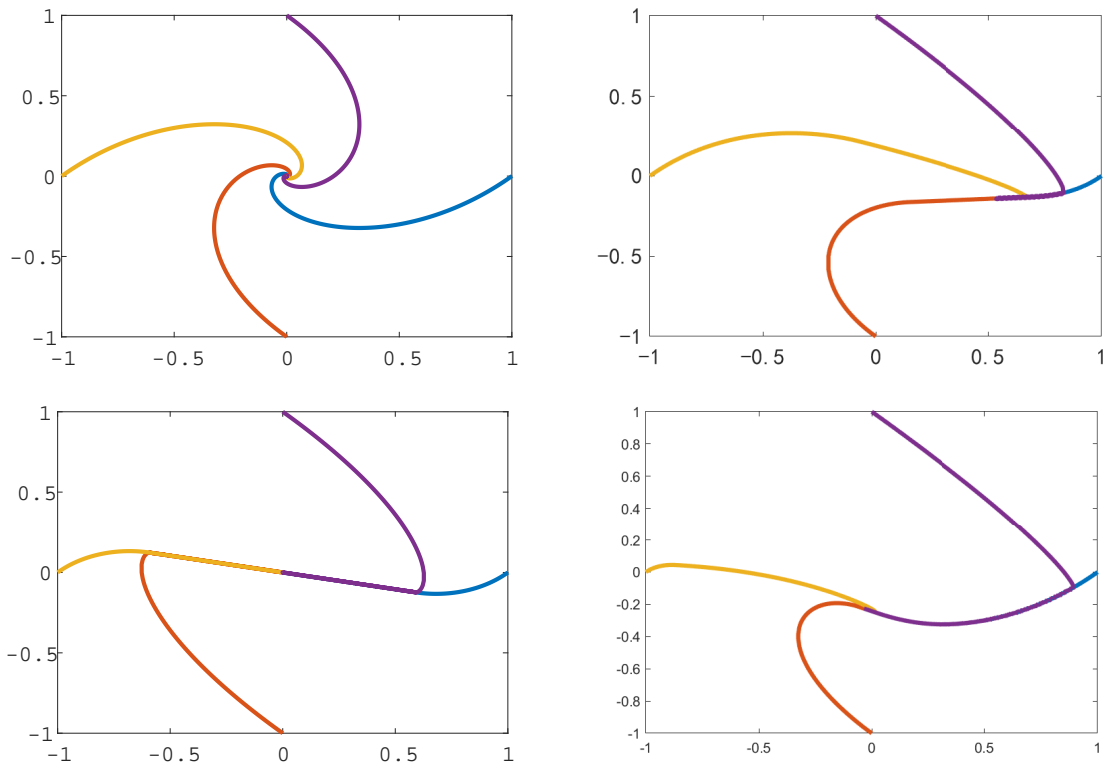


图 8 不同速度下四人的运动轨迹

同时, 只有在四人速度完全相同的时候, 它们的轨迹经过旋转可以完全重合; 在速度比例对称的时候 (即 $x:y:x:y$), 相对的两人轨迹中心对称, 其余情况轨迹均不会对称.

关于相遇点, 当四人速度完全相同的时候, 理论上的相遇点在坐标轴原点, 在四人速度不完全相同的时候, 每个可能的相遇点均靠近速度较慢的一侧.

3.2 利用仿真方法求解多人在正方形上的追及问题

本节仅研究多人速度相同情况下各人的运动轨迹. 首先将上节的公式推广至一般情况, 为表示方便, 先记 $x_{n+1} = x_1$, $y_{n+1} = y_1$. 则这 n 个人的运动方程如下:

$$\begin{aligned}
 x_i|_{t=t_0+\Delta t} &= x_i|_{t=t_0} + v_i \Delta t \frac{x_{i+1}|_{t=t_0} - x_i|_{t=t_0}}{\sqrt{(x_i|_{t=t_0} - x_{i+1}|_{t=t_0})^2 + (y_i|_{t=t_0} - y_{i+1}|_{t=t_0})^2}} \\
 y_i|_{t=t_0+\Delta t} &= y_i|_{t=t_0} + v_i \Delta t \frac{y_{i+1}|_{t=t_0} - y_i|_{t=t_0}}{\sqrt{(x_i|_{t=t_0} - x_{i+1}|_{t=t_0})^2 + (y_i|_{t=t_0} - y_{i+1}|_{t=t_0})^2}} \\
 i &= 1, 2, \dots, n
 \end{aligned}$$

显然, n 个人运动的初始状态有所改变, 将它们看作一个圆上顺时针遍历的 n 等分

点, 则由几何关系可得:

$$\begin{aligned} x_i|_{t=0} &= r \cos -\frac{2\pi}{n} * (i-1) \\ y_i|_{t=0} &= r \sin -\frac{2\pi}{n} * (i-1) \\ i &= 1, 2, \dots, n \end{aligned}$$

根据上述的运动方程与初始状态便可以计算 n 个人的运动轨迹, 在下面的实验中, 我们在这 n 个人速度相等且速度与半径比值为 $\frac{r}{v} = 1$ 的情况下, 改变人数, 研究它们的运动轨迹. 以下是 $n = 10, 50, 250$ 时的实验结果:

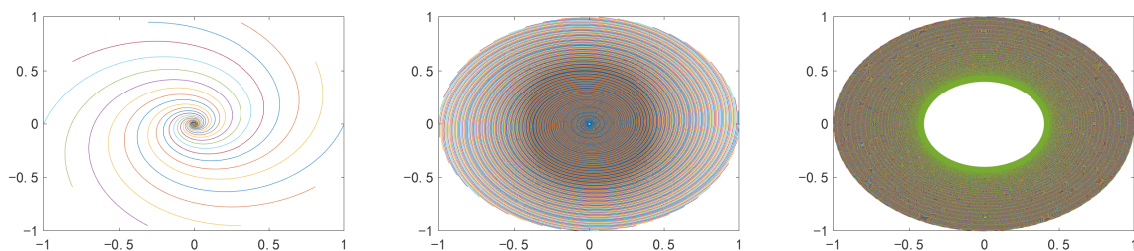


图 9 不同人数下的运动轨迹

根据实验结果可以发现, 随着人数的增多, 中心区域趋向于一个圆形, 而且虽然人数增大, 它的半径也同步变大, 所以可以预测, 在人数达到一定数目后, 它们将近似在初始状态下半径为 r 的圆上运动.

在下面的两小节中, 我们将通过方程的对称性, 使用精确解法对方程进行求解.

3.3 利用精确解求解四人在正方形上的追及问题

本节仅研究四人速度相同情况下各人的运动轨迹.

由于四人运动状态完全相同, 而四个人的初始状态正好可以通过旋转的方式得到 (即构成一个旋转群), 所以在任意时刻, 四个人的状态仍然可以通过旋转的方式得到, 这样只用考虑两个人的追及问题, 然后把所求轨迹分别旋转 $90, 180, 270$ 度, 就可以得到四个人的运动轨迹, 设第一个人的坐标为 (x, y) , 则第二个人的坐标为第一个人顺时针旋转 90 度后的坐标:

$$\begin{aligned} x' &= \cos\left(-\frac{\pi}{2}\right)x - \sin\left(-\frac{\pi}{2}\right)y = y \\ y' &= \sin\left(-\frac{\pi}{2}\right)x + \cos\left(-\frac{\pi}{2}\right)y = -x \end{aligned} \quad (26)$$

所以最终得到的微分方程为:

$$\frac{dy}{dx} = \frac{-x-y}{y-x} = \frac{x+y}{x-y} \quad (27)$$

$$y|_{x=r} = 0 \quad (28)$$

方程 (27) 可以转化为:

$$(x - y) dy = (x + y) dx \quad (29)$$

将方程 (29) 等价变化为:

$$\frac{x dx + y dy}{x^2 + y^2} + \frac{y dx - x dy}{x^2 + y^2} = 0 \quad (30)$$

利用积分因子法, 对方程 (30) 变换后可得:

$$\frac{1}{2} \frac{d(x^2 + y^2)}{x^2 + y^2} - \frac{d(\frac{y}{x})}{1 + (\frac{y}{x})^2} = 0 \quad (31)$$

对上式积分可得:

$$\log \sqrt{x^2 + y^2} - \arctan \frac{y}{x} = C \quad (32)$$

结合初始条件, 可得方程的解为:

$$\log \sqrt{x^2 + y^2} - \arctan \frac{y}{x} = \log r \quad (33)$$

该方程的解能够通过极坐标继续简化, 将 (x, y) 转化为 (R, θ) , 可得:

$$\begin{aligned} \log R - \theta &= \log r \\ \log \frac{R}{r} &= \theta \\ R(r, \theta) &= r \exp \theta, \quad \theta \leq 0 \end{aligned}$$

根据方程的解的形式, 可以判断出这条曲线是阿基米德螺线. 当时间 $t \rightarrow \infty$ 的时候, $\theta \rightarrow \infty$, 此时 $R \rightarrow 0$, 符合四人最终将相遇于坐标原点的情况.

根据极坐标下曲线的求长公式, 对这条曲线求长:

$$\begin{aligned} s &= \int_0^{-\infty} \sqrt{r^2(\theta) + r'^2(\theta)} d\theta \\ &= \int_0^{-\infty} \sqrt{2} r \exp \theta d\theta \\ &= \sqrt{2} r \exp \theta \Big|_{\theta=0}^{-\infty} \\ &= \sqrt{2} r \end{aligned}$$

这说明, 从精确解的角度上看, 四人最终在有限时间内相交于原点, 相遇时间 $t = \sqrt{2} \frac{r}{v}$. 以下是它们的运动轨迹:

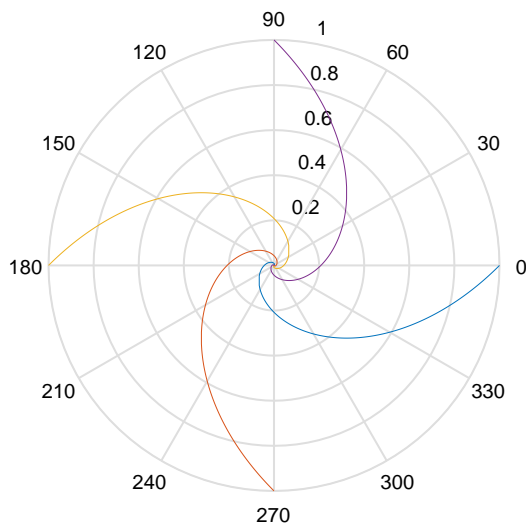


图 10 精确解下四人的运动轨迹

这与使用仿真方法求解出的结果类似.

3.4 利用精确解求解多人在正方形上的追及问题

本节仅研究多人速度相同情况下各人的运动轨迹, 作为上一节的推广, 对于 n 个人的情形, 仍然可以作为 n 阶旋转群处理, 设第一个人的坐标为 (x, y) , 则第二个人的坐标为第一个人顺时针旋转 $\frac{360}{n}$ 度后的坐标:

$$\begin{aligned} x' &= \cos\left(-\frac{2\pi}{n}\right)x - \sin\left(-\frac{2\pi}{n}\right)y \\ y' &= \sin\left(-\frac{2\pi}{n}\right)x + \cos\left(-\frac{2\pi}{n}\right)y \end{aligned} \quad (34)$$

所以最终得到的微分方程为:

$$\frac{dy}{dx} = \frac{\sin\left(-\frac{2\pi}{n}\right)x + \cos\left(-\frac{2\pi}{n}\right)y - y}{\cos\left(-\frac{2\pi}{n}\right)x - \sin\left(-\frac{2\pi}{n}\right)y - x} \quad (35)$$

$$y|_{x=r} = 0 \quad (36)$$

方程 (35) 可以转化为:

$$\left((1 - \cos\left(-\frac{2\pi}{n}\right))x + \sin\left(-\frac{2\pi}{n}\right)y\right) dy = \left((1 - \cos\left(-\frac{2\pi}{n}\right))y - \sin\left(-\frac{2\pi}{n}\right)x\right) dx \quad (37)$$

将方程 (37) 等价变化为:

$$\sin\left(\frac{2\pi}{n}\right) \frac{x dx + y dy}{x^2 + y^2} + (1 - \cos\left(\frac{2\pi}{n}\right)) \frac{y dx - x dy}{x^2 + y^2} = 0 \quad (38)$$

此时发现, 除了系数有所不同以外, 该问题与上一部分的问题完全一致, 根据相同方法求出方程的解为:

$$\sin\left(\frac{2\pi}{n}\right) \log \sqrt{x^2 + y^2} - (1 - \cos\left(\frac{2\pi}{n}\right)) \arctan \frac{y}{x} = C \quad (39)$$

结合初始条件, 可得方程的解为:

$$\sin\left(\frac{2\pi}{n}\right) \log \sqrt{x^2 + y^2} - (1 - \cos\left(\frac{2\pi}{n}\right)) \arctan \frac{y}{x} = \sin\left(\frac{2\pi}{n}\right) \log r \quad (40)$$

该方程的解能够通过极坐标继续简化, 将 (x, y) 转化为 (R, θ) , 可得:

$$\begin{aligned} \sin\left(\frac{2\pi}{n}\right) \log R - (1 - \cos\left(\frac{2\pi}{n}\right))\theta &= \sin\left(\frac{2\pi}{n}\right) \log r \\ \sin\left(\frac{2\pi}{n}\right) \log \frac{R}{r} &= (1 - \cos\left(\frac{2\pi}{n}\right))\theta \\ R(r, \theta) &= r \exp\left(\frac{1 - \cos\left(\frac{2\pi}{n}\right)}{\sin\left(\frac{2\pi}{n}\right)}\theta\right), \quad \theta \leq 0 \end{aligned}$$

这个方程的解仅与四人追及方程的解在 θ 项上差一个系数, 因此曲线在 n 为有限值时仍然可求长, 且

$$\begin{aligned} s &= \frac{\sin\left(\frac{2\pi}{n}\right)}{1 - \cos\left(\frac{2\pi}{n}\right)} \cdot \sqrt{1 + \left(\frac{1 - \cos\left(\frac{2\pi}{n}\right)}{\sin\left(\frac{2\pi}{n}\right)}\right)^2} \cdot r \\ &= \sqrt{1 + \left(\frac{\sin\left(\frac{2\pi}{n}\right)}{1 - \cos\left(\frac{2\pi}{n}\right)}\right)^2} \cdot r \end{aligned} \quad (41)$$

故 n 人追及所需时间为:

$$t = \sqrt{1 + \left(\frac{\sin\left(\frac{2\pi}{n}\right)}{1 - \cos\left(\frac{2\pi}{n}\right)}\right)^2} \cdot \frac{r}{v} \quad (42)$$

以下是 $n = 20, 60$ 时, 各人的运动轨迹:

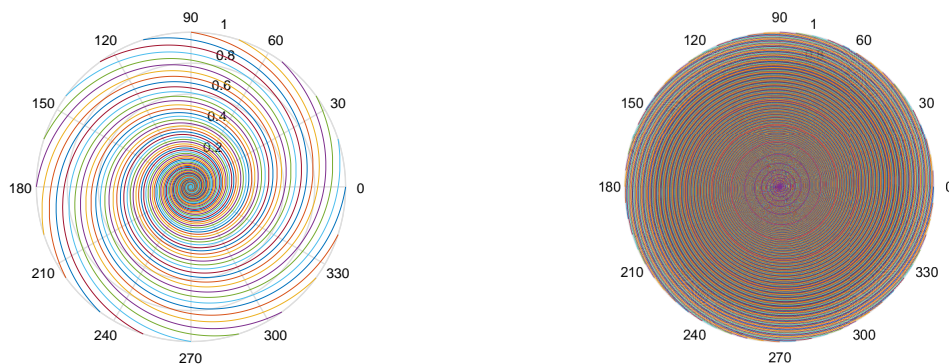


图 11 精确解下多人的运动轨迹

根据求解结果, s, t 关于 n 单调递增, 即人数越多, 运动路程越长, 所需相遇时间越长. 同时当 $n \rightarrow \infty$ 时, 有 $s \rightarrow \infty$, $t \rightarrow \infty$, $R \rightarrow r$, 反映到实际情况则是他们将在半径为 r 的圆上运动. 这符合仿真方法求解的结果.

3.5 仿真方法的误差分析

以上的方程表明, 最终轨迹与速度无关, 速度只能决定人们最终相遇的时间, 但是在仿真方法的求解中, 速度与半径的比值较大时, 也会产生圆形区域. 我们认为, 这种情况产生的原因与时间步长 Δt 的选取有关, 如果 $v\Delta t \approx r'$, 那么在以 r' 为半径的区域内, 任何人在 Δt 时间内运动的距离约等于 r' , 即由于精度问题他们不能向半径小于 r' 的区域运动, 这就在中心产生了一块空白的区域.

综上所述, 仿真解法对时间步长的选取较为严格, 这也是造成计算误差的来源.

4 实验总结

本次实验主要采用了精确解法, 数值解法与仿真方法求解关于追及问题的微分方程模型, 精确解法能够得出模型的最精确的解, 但是难以推广至一般情况, 它计算上可以完全不需要计算机的帮助, 对于追及两方初始速度夹角为 90 度的问题, 可以考虑这种解法处理; 数值解法依托迭代的方式得出微分方程的数值解, 在区间足够合理, 精度足够高的情况下, 能够达到与精确解法非常接近的解, 而且它能够处理的问题更加广泛, 需要的预处理较少, 对于追及两方初始速度夹角不为 90 度的问题以及部分与速度与时间相关的问题, 可以考虑这种方法处理; 仿真方法可以运用在最广泛的问题中, 只要时间步长选取得当, 往往编写程序较为容易, 但是选取较大的时间步长将损失计算精度, 选取较小的时间步长会消耗计算时间, 处理时间与效率的平衡是这一方法需要解决的问题.

在狗追兔子的模型求解中, 我们分别使用了三种不同解法对问题进行了求解, 它们的结果相差不大, 说明对于两个物体的追及问题, 这三种方法均能处理; 但是在速度与时间相关的问题中, 只有仿真算法能够处理. 在多人相互追及的模型求解中, 在使用仿真方法解决问题的过程中, 我们发现在速度与半径的比值较大时或人数较多时, 最终奔跑的结果将会趋向一个圆形, 这可能与微分方程的稳定性密切相关. 同时我们使用了精确求解的方法, 证明了人数较多时, 最终奔跑的结果将会趋向一个圆形. 而速度与半径的比值较大时产生圆形区域的原因在于时间步长较大, 达不到得出奔跑轨迹的精确性要求.

附录 A 实验代码

A.1 狗追兔子精确求解代码

```

r = (-5 + sqrt(61)) / 6;
y = @(x) x ./ 2 * (((-200 ./ x)^(- r)) / (- r - 1) + ((-x ./ 200)^(- r))
    ↪ / (- r + 1)) + 200 * -r / (r^2 - 1);
x = linspace(-200, 0, 1000);
for i = 1:1000
    s(i) = y(x(i));
end
plot(x, s)

xr = cos(-pi / 4) * x + sin(-pi / 4) * s;
yr = -sin(-pi / 4) * x + cos(-pi / 4) * s;
plot(xr, yr)

```

A.2 狗追兔子数值求解代码

```

% for v_d = 17:0.0001:18
%     tspan = [0 15];
%     initial = [-100 * sqrt(2) -100 * sqrt(2)];
%     options = odeset('RelTol',1e-12,'AbsTol',[1e-10 1e-10]);
%     [T, Y] = ode45(@(t, y)odefun(t, y, v_d), tspan, initial,
    ↪ options); % Solve the function
% end

% get distance
v_d = 17.1488;
initial = [-100 * sqrt(2) -100 * sqrt(2)];
options = odeset('RelTol',1e-12,'AbsTol',[1e-10 1e-10]);
[T, Y] = ode45(@(t, y)odefun(t, y, v_d), tspan, initial, options); %
    ↪ Solve the function
dist = 0;
for i = 1: size(Y, 1) - 1
    dist = dist + sqrt((Y(i + 1, 1) - Y(i, 1))^2 + (Y(i + 1, 2) - Y(i,
    ↪ 2))^2);
end

```

```

end
dist
plot(Y(:, 1), Y(:, 2))

function dydt = odefun(t, y, v_d)
    dydt = zeros(2, 1);
    dydx = (y(2) - 4 * sqrt(2) * t) / (y(1) + 4 * sqrt(2) * t);
    dydt(1) = v_d / sqrt(1 + (dydx)^2) * (dydx >= 0) - v_d / sqrt(1 +
        ↪ (dydx)^2) * (dydx < 0);
    dydt(2) = abs(dydx * dydt(1));
    if (norm(y(2) - 4 * sqrt(2) * t, y(1) + 4 * sqrt(2) * t) < eps)
        % distance ok
        error("Successfully chased within 15 seconds. Current velocity:
            ↪ " + v_d);
    end
end
end

```

A.3 狗追兔子仿真方法求解代码-1

```

v_d = 17.0952;
epsilon = 1e-6;
step = 1e-3;
max_time = 15;
xd = zeros(1, ceil(max_time / step) + 1);
yd = zeros(1, ceil(max_time / step) + 1);
xd(1) = -100 * sqrt(2);
yd(1) = -100 * sqrt(2);
for i = 1: ceil(max_time / step)
    time = (i - 1) * step;
    xr = -4 * sqrt(2) * time;
    yr = 4 * sqrt(2) * time;
    distance = sqrt((xd(i) - xr)^2 + (yd(i) - yr)^2);
    if (distance < epsilon)
        break;
    end
    xdir = xr - xd(i);

```

```
ydir = yr - yd(i);
xd(i + 1) = xd(i) + v_d * step * xdir / distance;
yd(i + 1) = yd(i) + v_d * step * ydir / distance;
end
dist = 0;
for i = 1: 15000
    dist = dist + sqrt((xd(i + 1) - xd(i))^2 + (yd(i + 1) - yd(i))^2);
end
dist
plot(xd(1: i), yd(1: i))
```

A.4 狗追兔子仿真方法求解代码-2

```
v_d = 15.4415;
epsilon = 1e-6;
step = 1e-4;
max_time = 20;
xr = zeros(1, ceil(max_time / step) + 1);
yr = zeros(1, ceil(max_time / step) + 1);
xd = zeros(1, ceil(max_time / step) + 1);
yd = zeros(1, ceil(max_time / step) + 1);
xd(1) = -100 * sqrt(2);
yd(1) = -100 * sqrt(2);
xr(1) = 0;
yr(1) = 0;
distance = inf;
key = inf;
r_dist = 0;
d_dist = 0;

for i = 1: ceil(max_time / step)
    if (distance < 30)
        key = min(key, i);
    end
    time = (i - 1) * step;
    if ~isinf(key)
```

```

        v_rt = @(t) 8 * (0.5)^(t - (key - 1) * step);
        v_dt = @(t) v_d * (1.1)^(t - (key - 1) * step);
    else
        v_rt = @(t)8;
        v_dt = @(t) v_d;
    end
    xr(i + 1) = xr(i) - v_rt(time) * cos(pi / 4) * step;
    yr(i + 1) = yr(i) + v_rt(time) * sin(pi / 4) * step;
    distance = sqrt((xd(i) - xr(i))^2 + (yd(i) - yr(i))^2);
    if (distance < epsilon)
        break;
    end
    xdir = xr(i) - xd(i);
    ydir = yr(i) - yd(i);
    xd(i + 1) = xd(i) + v_dt(time) * step * xdir / distance;
    yd(i + 1) = yd(i) + v_dt(time) * step * ydir / distance;
    d_dist = d_dist + sqrt((xd(i + 1) - xd(i))^2 + (yd(i + 1) -
        ↪ yd(i))^2);
    r_dist = r_dist + sqrt((xr(i + 1) - xr(i))^2 + (yr(i + 1) -
        ↪ yr(i))^2);
end

if (i < ceil(max_time / step)) && (r_dist <= 120)
    v_d
    time
    d_dist
end

plot(xd(1: i), yd(1: i))

```

A.5 多人追及问题求解代码

```

n = 250;
r = 1;
v = ones(1, n);
step = 1e-2;

```



```

max_time = 200;
x = zeros(n, ceil(max_time / step) + 1);
y = zeros(n, ceil(max_time / step) + 1);
for j = 1: n
    x(j, 1) = r * cos(- 2 * pi / n * (j - 1));
    y(j, 1) = r * sin(- 2 * pi / n * (j - 1));
end
per_dist = zeros(n, 1);

for i = 1: ceil(max_time / step)
    for j = 1: n
        distance = sqrt((x(mod(j, n) + 1, i) - x(j, i))^2 + ...
                        (y(mod(j, n) + 1, i) - y(j, i))^2);
        xdir = x(mod(j, n) + 1, i) - x(j, i);
        ydir = y(mod(j, n) + 1, i) - y(j, i);
        x(j, i + 1) = x(j, i) + v(j) * step * xdir / distance;
        y(j, i + 1) = y(j, i) + v(j) * step * ydir / distance;
        per_dist(j) = per_dist(j) + sqrt((x(j, i + 1) - x(j, i))^2 +
        ↪ (y(j, i + 1) - y(j, i))^2);
    end
end

for j = 1: n
    plot(x(j, :), y(j, :));
    hold on;
end

```

A.6 多人追及问题精确解作图代码

```

n = 1000;
theta = linspace(0, -100 * pi, 100000);
for i = 1: n
    r = 1 * exp((1 - cos(2 * pi / n)) / (sin(2 * pi / n)) * theta);
    polar(theta - 2 * pi / n * (i - 1), r);
    hold on;
end

```