

数学实验：第二次作业

计算机系 计 73 2017011620 李家昊

2020 年 3 月 17 日

1 实验目的

- 掌握用 MATLAB 软件求微分方程初值问题数值解的方法
- 通过实例学习用微分方程模型解决简化的实际问题
- 了解欧拉方法和龙格-库塔方法的基本思想和计算公式，及稳定性等概念
- 练习数值微分的计算

2 问题求解

2.1 Chap4-Ex5 放射性废物的处理（应用题）

2.1.1 问题分析

题目给定装有放射性废物圆桶的体积，重量，所受海水浮力，海水阻力，最大撞击速度等参数，求圆桶撞击海底时会不会破裂。

2.1.2 模型假设

考虑到实际情况，此模型基于以下假设，

1. 圆桶紧贴水面以零初速度自由下落。
2. 圆桶下落过程中仅受浮力，重力，以及海水阻力的作用。
3. 海水的密度不随深度变化，即海水对圆桶的浮力不变。

2.1.3 模型建立

设圆桶的质量为 m ，重力加速度为 g ，所受浮力固定为 F ，海水阻力系数为 k ，圆桶在 t 时刻下沉速度为 $v(t)$ ，根据牛顿第二定律得到下列微分方程，

$$mg - F - kv = m \frac{dv}{dt} \quad (1)$$

不妨假设物体在 $t = 0$ 时刻自由落体投入水中，即 $v(0) = 0$ ，作为微分方程的初值条件。

2.1.4 算法设计

解析解法 根据方程 (1) 得出通解为，

$$v(t) = \frac{1}{k} \left(mg - F - Ae^{-\frac{k}{m}t} \right) \quad (2)$$

带入初值条件 $v(0) = 0$ ，解得系数 $A = mg - F$ ，得到特解，

$$v(t) = \frac{mg - F}{k} \left(1 - e^{-\frac{k}{m}t} \right) \quad (3)$$

设圆桶在 t_m 时刻达到最大承受速度 v_m ，根据方程 (3) 解得，

$$t_m = \frac{m}{k} \ln \left(1 + \frac{kv_m}{mg - F - kv_m} \right) \quad (4)$$

设圆桶在 t 时刻下落距离为 $s(t)$ ，则有，

$$s(t) = \int_0^t v(\tau) d\tau = \frac{mg - F}{k} \left(t + \frac{m}{k} e^{-\frac{k}{m}t} - \frac{m}{k} \right) \quad (5)$$

设海水深度为 s_m ，若 $s(t_m) < s_m$ ，则说明圆桶到达海底之前，速度已经达到 v_m ，因此圆桶撞击海底时将破裂；否则，圆桶将不会破裂。

带入题目提供的数据， $g = 9.80665 \text{ m/s}^2 = 32.17405 \text{ ft/s}^2$ ， $mg = 527.436 \text{ lb}$ ， $F = 470.327 \text{ lb}$ ， $k = 0.08 \text{ lb} \cdot \text{s/ft}$ ， $v_m = 40 \text{ ft/s}$ ， $s_m = 300 \text{ ft}$ ，解得 $t_m = 11.8163 \text{ s}$ ， $s(t_m) = 238.5968 \text{ ft}$ ，因此有 $s(t_m) < s_m$ ，圆桶撞击海底时将破裂，工程师赢得了官司。

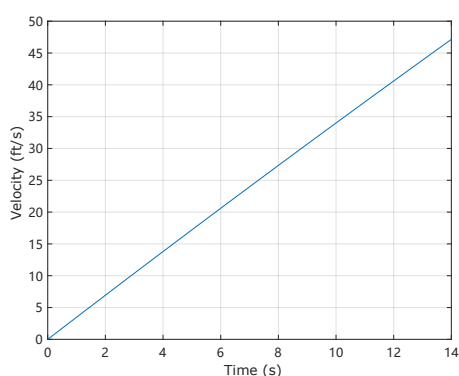
数值解法 模型需要求解方程 (1)，为非刚性方程，可利用 Matlab 内置的 5 级 4 阶 Runge-Kutta 方法 `ode45` 求解。

2.1.5 Matlab 程序

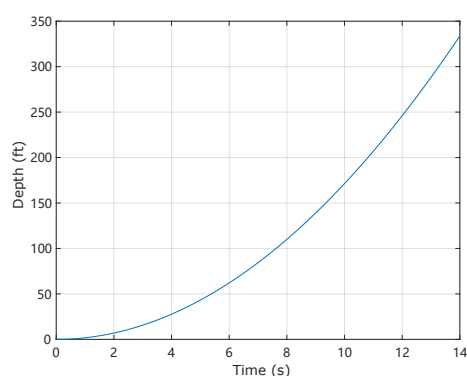
请参见 Section 4.1。

表 1: 圆桶临近海底时的时刻，速度和深度的数值结果。

Time (s)	Velocity (ft/s)	Depth (ft)
13.0000	43.8814	288.2447
13.1000	44.2083	292.6492
13.2000	44.5350	297.0864
13.3000	44.8616	301.5562
13.4000	45.1880	306.0587
13.5000	45.5142	310.5938



(a) 速度随时间变化图像



(b) 深度随时间变化图像

图 1: 圆桶的速度和深度随时间变化的图像

2.1.6 计算结果

圆桶的速度和深度随时间变化的图像如图 1所示，截取圆桶临近海底时的数值结果如表 1所示，可以看出，圆桶撞击海底的时刻约为 $t = 13.3$ s，速度约为 $v = 44.8$ ft/s，大于圆桶的最大承受速度 $v_m = 40$ ft/s。

2.1.7 结果的数学分析

为了分析微分方程数值解法的误差，设圆桶速度的解析解为 $v(t)$ ，数值解为 $v^*(t)$ ，则在数值解法的离散点处作出 $v^*(t) - v(t)$ 的残差图像，如图 2所示。可见，在 $t \in [0, 14]$ 范围内，圆桶速度的绝对误差控制在 10^{-11} ft/s 以内；撞击海底时，圆桶速度的相对误差控制在 10^{-12} 以内，数值计算的结果十分精确，可作为最终的计算结果。

2.1.8 结果的实际意义

圆桶大约以 44.8 ft/s 的速度撞击海底，并发生破裂，工程师将赢得官司。

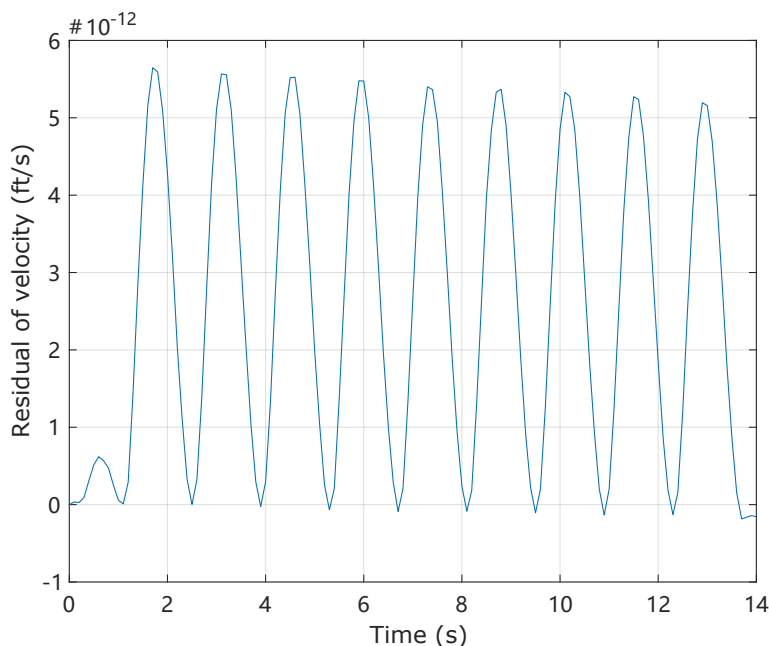


图 2: 圆桶速度的数值解与解析解之间的残差图像

2.1.9 结论

圆桶撞击海底时会发生破裂，工程师赢得了官司。

2.2 Chap4-Ex6 小船过河（计算题）

2.2.1 算法设计

数学模型 以 A 为原点，以 v_1 方向为 x 轴正方向， \overrightarrow{AB} 方向为 y 轴正方向，建立平面直角坐标系，则有坐标 $A(0,0)$, $B(0,d)$ ，设在 t 时刻，船的坐标为 $C(x,y)$ ，设 $\angle ABC = \theta$ ，则可列出微分方程组，

$$\frac{dx}{dt} = v_1 - v_2 \sin \theta, \quad \frac{dy}{dt} = v_2 \cos \theta \quad (6)$$

消掉 t ，带入题目条件 $k = v_1/v_2$ ，得到，

$$\frac{dx}{dy} = \frac{v_1 - v_2 \sin \theta}{v_2 \cos \theta} = \frac{k - \sin \theta}{\cos \theta} \quad (7)$$

其中 θ 满足，

$$\cos \theta = \frac{d - y}{\sqrt{x^2 + (d - y)^2}}, \quad \sin \theta = \frac{x}{\sqrt{x^2 + (d - y)^2}} \quad (8)$$

带入方程 (7) 得到，

$$\frac{dx}{dy} = \frac{k\sqrt{x^2 + (d - y)^2} - x}{d - y} \quad (9)$$

即为模型对应的微分方程，对于初值，由于小船从 $A(0,0)$ 出发，因此有 $x(0) = 0$ 。

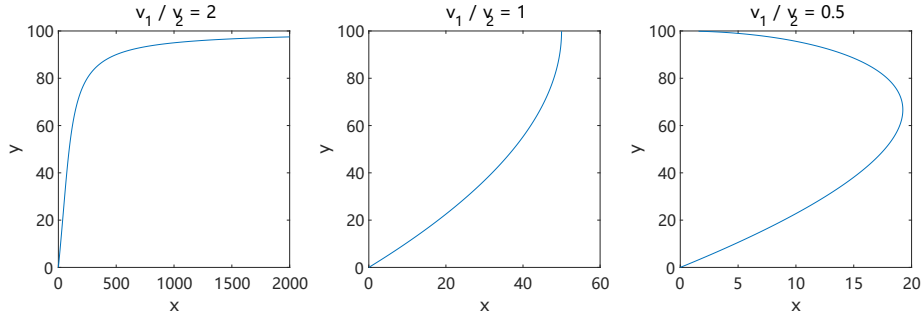


图 3: 当 $k = v_1/v_2$ 取不同值时, 小船航线的解析解。

解析解法 设 $p = x/(d - y)$, 则有,

$$\frac{dp}{dy} = \frac{1}{(d - y)^2} \left(\frac{dx}{dy}(d - y) + x \right) = \frac{1}{d - y} \left(\frac{dx}{dy} + p \right) \quad (10)$$

带入方程 (9) 得到,

$$(d - y) \frac{dp}{dy} - p = k \sqrt{p^2 + 1} - p \quad (11)$$

可分离变量得到,

$$\frac{1}{\sqrt{p^2 + 1}} dp = \frac{k}{d - y} dy \quad (12)$$

两边积分, 解得通解为,

$$p + \sqrt{p^2 + 1} = A(d - y)^{-k} \quad (13)$$

带入初值条件 $p(0) = x(0)/(d - 0) = 0$, 解得 $A = d^k$, 则特解为,

$$p + \sqrt{p^2 + 1} = \left(\frac{d}{d - y} \right)^k \quad (14)$$

将 p 移项, 两边平方, 化简得到,

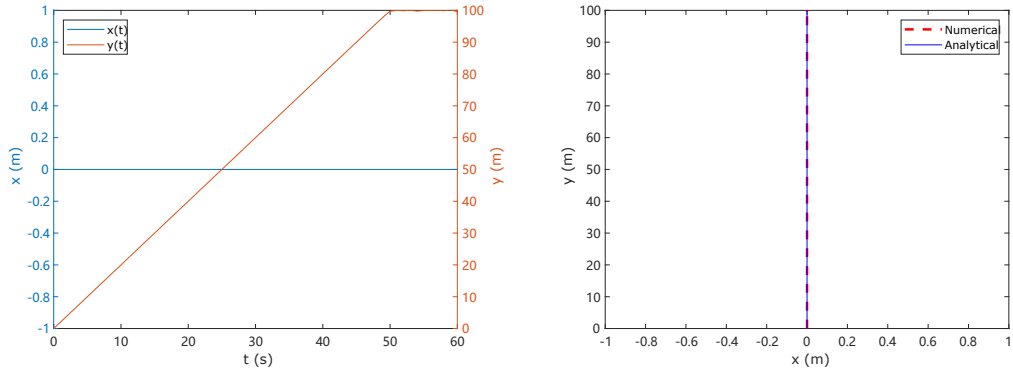
$$p = \frac{1}{2} \left[\left(\frac{d}{d - y} \right)^k - \left(\frac{d}{d - y} \right)^{-k} \right] \quad (15)$$

带入 $p = x/(d - y)$ 得到小船航线的解析解为,

$$x = \frac{d - y}{2} \left[\left(\frac{d}{d - y} \right)^k - \left(\frac{d}{d - y} \right)^{-k} \right], \quad 0 \leq y \leq d \quad (16)$$

当 v_1/v_2 取不同值时, 小船航线的解析解如图 3 所示, 可以看出, 当 $v_2 \leq v_1$ 时, 小船永远到不了终点 B , 当 $v_2 > v_1$ 时, 小船能到达终点 B 。

数值解法 可将方程 (6) 视为刚性方程, 利用 Matlab 内置的 ode23s 方法求解即可。



(a) $x(t)$ 和 $y(t)$ 图像

(b) $x(y)$ 图像

图 4: 流速 $v_1 = 0$ m/s 时, 每一时刻的小船坐标 (x, y) 以及航线的数值结果。

2.2.2 Matlab 程序

请参见 Section 4.2。

2.2.3 计算结果

流速 $v_1 = 0$ m/s 时 计算结果如图 4所示, 小船渡河时间为 50 s。

流速 $v_1 = 0.5$ m/s 时 计算结果如图 5所示, 小船渡河时间约为 54 s。

流速 $v_1 = 1$ m/s 时 计算结果如图 6所示, 小船渡河时间约为 67 s。

流速 $v_1 = 1.5$ m/s 时 计算结果如图 7所示, 小船渡河时间约为 114 s。

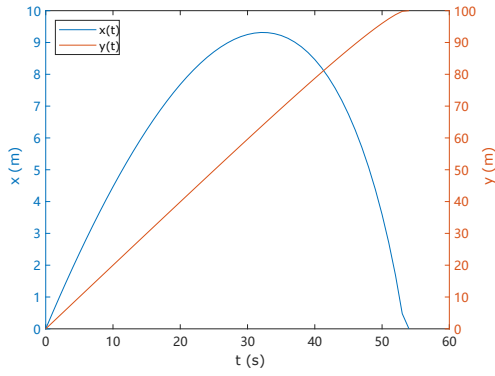
流速 $v_1 = 2$ m/s 时 计算结果如图 8所示, 小船无限逼近终点, 但始终不能到达, 渡河所需时间为无穷大。

2.2.4 结果分析

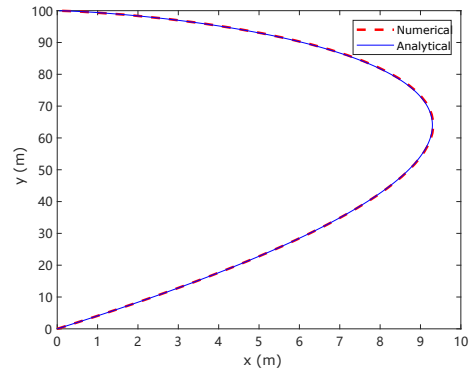
数值结果的精度较高, 与解析结果十分接近。从以上结果可以看出, 当 $v_1 < v_2$ 时, 小船能够到达终点, 且 v_1 越小, 则渡河时间越短; 当 $v_1 = v_2$ 时, 随着时间的流逝, 小船无限逼近终点, 但永远不能到达; 当 $v_1 > v_2$ 时, 小船将随波逐流, 离终点越来越远。

2.2.5 结论

当流速小于船速时, 小船能够到达终点, 且流速越小, 小船渡河时间越短; 当流速等于船速时, 小船无限逼近终点, 但永远不能到达; 当流速大于船速时,

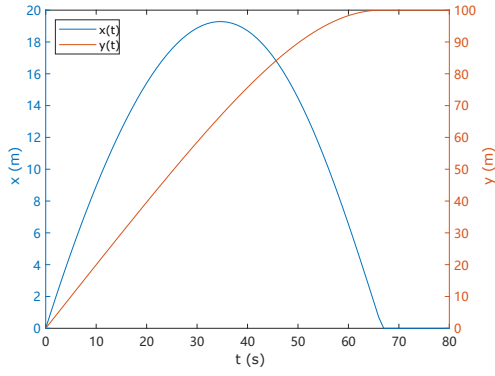


(a) $x(t)$ 和 $y(t)$ 图像

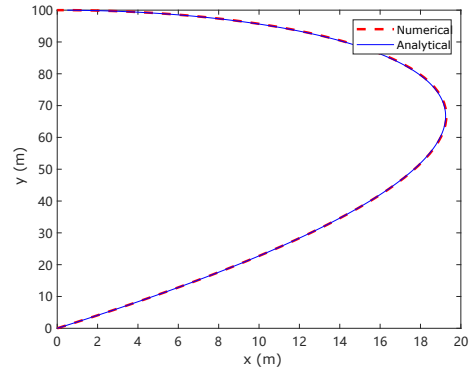


(b) $x(y)$ 图像

图 5: 流速 $v_1 = 0.5$ m/s 时, 每一时刻的小船坐标 (x, y) 以及航线的数值结果。

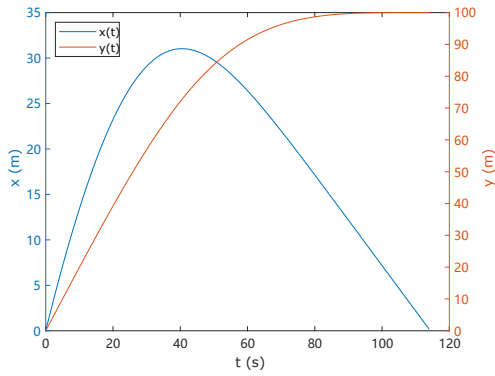


(a) $x(t)$ 和 $y(t)$ 图像

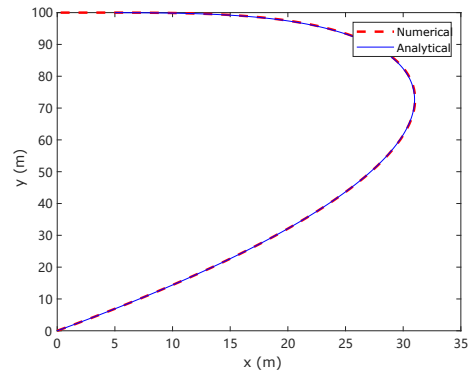


(b) $x(y)$ 图像

图 6: 流速 $v_1 = 1$ m/s 时, 每一时刻的小船坐标 (x, y) 以及航线的数值结果。



(a) $x(t)$ 和 $y(t)$ 图像



(b) $x(y)$ 图像

图 7: 流速 $v_1 = 1.5$ m/s 时, 每一时刻的小船坐标 (x, y) 以及航线的数值结果。

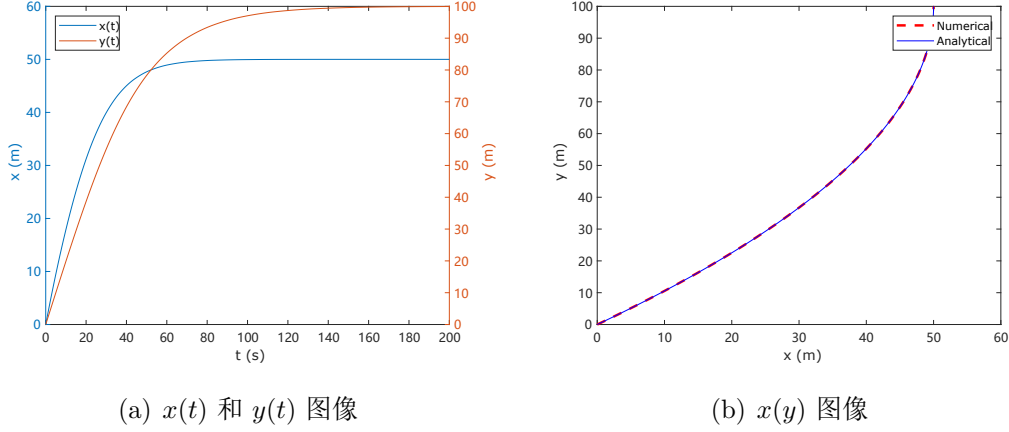


图 8: 流速 $v_1 = 2 \text{ m/s}$ 时, 每一时刻的小船坐标 (x, y) 以及航线的数值结果。

小船将离终点越来越远。当流速分别为 $0, 0.5, 1, 1.5, 2 \text{ m/s}$ 时, 小船渡河时间分别为 $50, 54, 67, 114, \infty \text{ s}$ 。

2.3 Chap4-Ex9 种群竞争（计算题）

2.3.1 算法设计

根据题目给出的种群竞争微分方程组模型, 以及各小题给出的初值条件, 利用 Matlab 内置的 `ode45` 求解即可。

$$\frac{dx}{dt} = r_1 x \left(1 - \frac{x}{n_1} - s_1 \frac{y}{n_2} \right), \quad \frac{dy}{dt} = r_2 y \left(1 - s_2 \frac{x}{n_1} - \frac{y}{n_2} \right) \quad (17)$$

其中 $x(t), y(t)$ 分别为甲乙两个种群的数量, r_1, r_2 为它们的固有增长率, n_1, n_2 为它们的最大容量, s_1 表示, 对于供养甲的资源而言, 单位数量乙的消耗是单位数量甲消耗的 s_1 倍, s_2 同理。

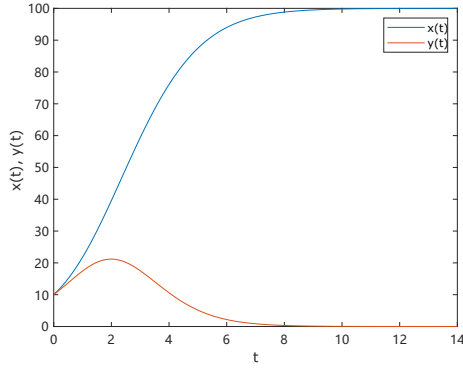
2.3.2 Matlab 程序

请参见 Section 4.3。

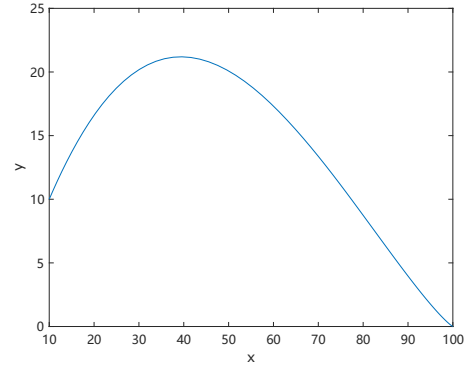
2.3.3 计算结果

第 1 小题 当 $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$ 时, 计算结果如图 9 所示, 时间 t 充分大后, $x(t)$ 趋向于 100, $y(t)$ 趋向于 0。

第 2 小题 当 $s_1 < 1$ 且 $s_2 > 1$ 时, 将参数重置为 $r_1 = 1, r_2 = 2, n_1 = 200, n_2 = 300, s_1 = 0.5, s_2 = 2, x_0 = 20, y_0 = 100$, 计算结果如图 10 所示。可以看出, 当 $s_1 < 1$ 且 $s_2 > 1$ 时, 时间充分大时, 甲种群数量 x 趋向于 n_1 , 乙种群数量 y 趋向于 0, 甲种群取代了乙种群。

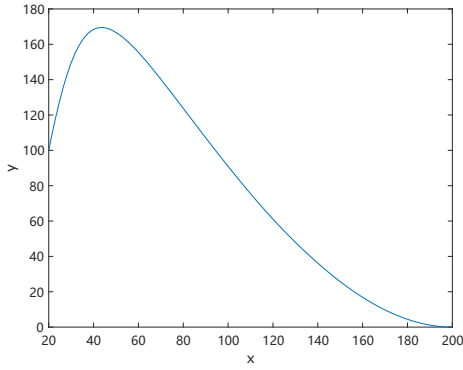


(a) 甲乙种群数量随时间变化的图像

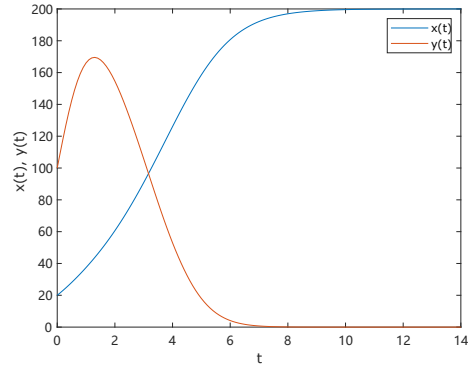


(b) 甲乙种群数量的相图

图 9: 当 $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$ 时, 甲乙种群数量的图像和相图。



(a) 甲乙种群数量随时间变化的图像



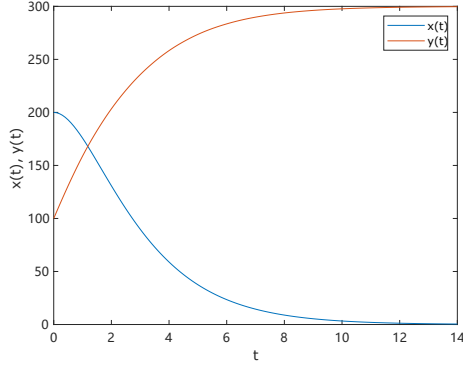
(b) 甲乙种群数量的相图

图 10: 当 $r_1 = 1, r_2 = 2, n_1 = 200, n_2 = 300, s_1 = 0.5, s_2 = 2, x_0 = 20, y_0 = 100$ 时, 甲乙种群数量的图像和相图。

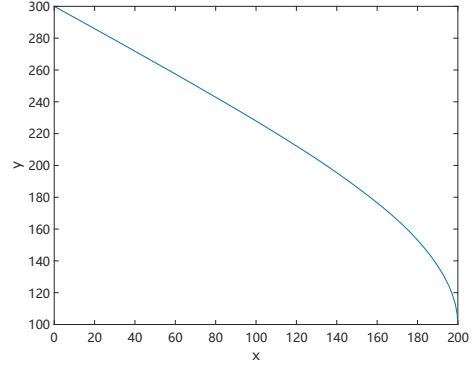
当 $s_1 > 1$ 且 $s_2 < 1$ 时, 将参数重置为 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 1.5, s_2 = 0.7, x_0 = 200, y_0 = 100$, 计算结果如图 11 所示。可以看出, 当 $s_1 > 1$ 且 $s_2 < 1$ 时, 时间充分大时, 甲种群数量 x 趋向于 0, 乙种群数量 y 趋向于 n_2 , 乙种群取代了甲种群。

第 3 小题 当 $s_1 < 1, s_2 < 1$ 时, 将参数重置为 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 0.8, s_2 = 0.7, x_0 = 200, y_0 = 100$, 计算结果如图 12 所示。可以看出, 当 $s_1 < 1, s_2 < 1$ 时, 时间充分大时, 甲乙种群数量稳定在固定值, 它们共同存在, 不能完全取代对方。

当 $s_1 > 1, s_2 > 1$ 时, 将参数重置为 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 1.5, s_2 = 1.7, x_0 = 200, y_0 = 100$, 计算结果如图 13 所示。可以看出, 时间充分大时, 甲种群取代了乙种群; 然而, 当我将初值调整为 $x_0 = 100, y_0 = 100$ 时, 情

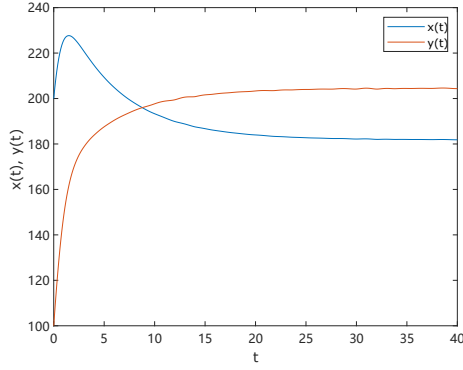


(a) 甲乙种群数量随时间变化的图像

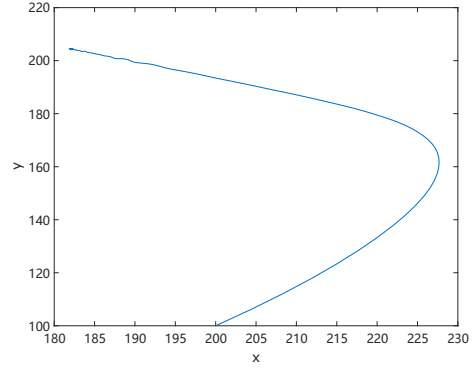


(b) 甲乙种群数量的相图

图 11: 当 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 1.5, s_2 = 0.7, x_0 = 200, y_0 = 100$ 时, 甲乙种群数量的图像和相图。



(a) 甲乙种群数量随时间变化的图像



(b) 甲乙种群数量的相图

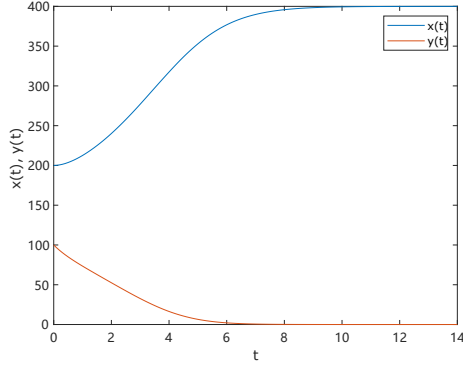
图 12: 当 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 0.8, s_2 = 0.7, x_0 = 200, y_0 = 100$ 时, 甲乙种群数量的图像和相图。

况则完全相反, 计算结果如图 14 所示。可以看出, 时间充分大时, 乙种群取代了甲种群。因此, 当 $s_1 > 1, s_2 > 1$ 时, 模型是不稳定的, 最终稳定状态的种群数量受初值条件的影响。

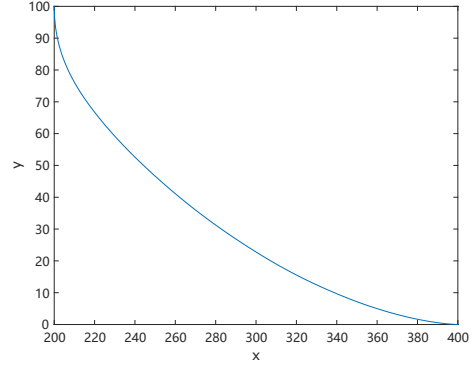
2.3.4 结果分析

由 x 和 y 的对称性, 不妨假设 $s_1 < s_2$, 此时单位数量乙对供养甲的资源的消耗量比单位数量甲对供养乙的资源的消耗量更少, 即甲的竞争能力相对更强。

当 $s_1 < s_2 < 1$ 时, 单位数量甲消耗供养乙的资源量为 $s_2 < 1$, 不及单位数量乙消耗供养乙的资源量, 因此甲不能完全取代乙, 随着时间的流逝, 甲乙种群数量将趋于一个固定值, 这个值可通过方程 (17) 确定, 令 $dx/dt = 0, dy/dt = 0$, 注意到此时 $x \neq 0, y \neq 0$, 可解得稳态时甲乙种群数量如下式, 与数值计算结果

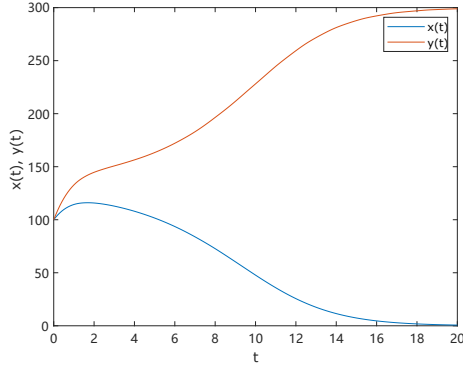


(a) 甲乙种群数量随时间变化的图像

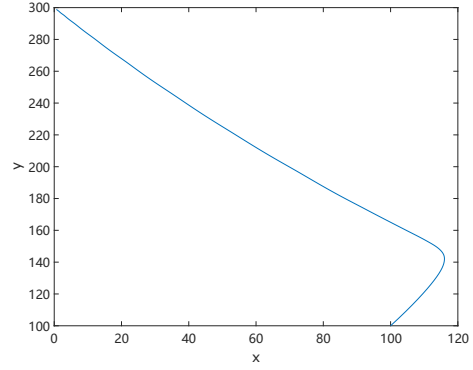


(b) 甲乙种群数量的相图

图 13: 当 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 1.5, s_2 = 1.7, x_0 = 200, y_0 = 100$ 时, 甲乙种群数量的图像和相图。



(a) 甲乙种群数量随时间变化的图像



(b) 甲乙种群数量的相图

图 14: 当 $r_1 = 1, r_2 = 2, n_1 = 400, n_2 = 300, s_1 = 1.5, s_2 = 1.7, x_0 = 100, y_0 = 100$ 时, 甲乙种群数量的图像和相图。

相符。

$$\lim_{t \rightarrow +\infty} x(t) = n_1 \frac{1 - s_1}{1 - s_1 s_2}, \quad \lim_{t \rightarrow +\infty} y(t) = n_2 \frac{1 - s_2}{1 - s_1 s_2} \quad (18)$$

当 $s_1 < 1 < s_2$ 时, 在时间足够长时, 由于甲的竞争力太强, 而乙的竞争力太弱, 因此甲最终会取代乙。

当 $1 < s_1 < s_2$ 时, 模型不稳定, 最终稳定状态的种群数量受初值条件的影响。

2.3.5 结论

当 $s_1 < s_2 < 1$ 时, 最终甲乙种群共存; 当 $s_1 < 1 < s_2$ 时, 最终甲会取代乙; 当 $1 < s_1 < s_2$ 时, 模型不稳定, 最终状态受初值条件的影响。

3 收获与建议

在本次实验中，我通过使用 Matlab，掌握了求微分方程初值问题数值解的方法，用微分方程模型解决了简化的实际问题，了解了欧拉方法和龙格-库塔方法的基本思想和计算公式，在解决实际问题的过程中，我对数学方法的原理和应用有了更深刻的理解。

希望助教能对每次的实验进行详细的解答，希望老师在未来的课堂上介绍更多数学应用的前沿知识。

4 附录：Matlab 程序代码

4.1 Chap4-Ex5

```
1 G = 527.436;
2 g = 32.17405;
3 m = G / g;
4 k = 0.08;
5 v0 = 40;
6 F = 470.327;
7 dv_dt = @(t, v) (G - F - k * v) / m;
8
9 [t, v] = ode45(dv_dt, 0:0.1:14, 0);
10
11 s = zeros(length(t), 1);
12 for i = 2:length(t)
13     s(i) = trapz(t(1:i), v(1:i));
14 end
15
16 figure;
17 plot(t, v);
18 grid on;
19 xlabel('Time (s)');
20 ylabel('Velocity (ft/s)');
21
22 figure;
23 plot(t, s);
24 grid on;
25 xlabel('Time (s)');
26 ylabel('Depth (ft)');
27
28 figure;
29 gt_v = @(t) (G - F) / k * (1 - exp(-k / m * t));
30 plot(t, v - gt_v(t));
```

```

31 grid on;
32 xlabel('Time (s)');
33 ylabel('Residual of velocity (ft/s)');

```

4.2 Chap4-Ex6

```

1  global d;
2  global v1;
3  global v2;
4
5  d = 100;
6  v2 = 2;
7  y = 0:0.1:d;
8  v1_choices = [1 2 4];
9
10 gt_x = @(y, d, v1, v2) (d-y)./2 .* ((d./(d-y)).^(v1 / v2) - (d./(d-y)
    ).^(-v1 / v2));
11
12 figure;
13 for i = 1:3
14     subplot(1, 3, i);
15     v1 = v1_choices(i);
16     x = gt_x(y, d, v1, v2);
17     plot(x, y);
18     xlabel('x');
19     ylabel('y');
20     if (max(x) > 2000)
21         xlim([0 2000]);
22     end
23     title(['v_1 / v_2 = ' num2str(v1/v2)]);
24 end
25
26 v1_choices = [0.5 1.5 2];
27
28 v1 = 2;
29 [t, xy] = ode23s(@dxy_dt, 0:200, [0 0]);
30 x = xy(:, 1);
31 y = xy(:, 2);
32 figure;
33 yyaxis left;
34 plot(t, x);
35 ylabel('x (m)');
36 yyaxis right;
37 plot(t, y);

```

```

38 ylabel('y (m)');
39 ylim([0 100]);
40 xlabel('t (s)');
41 legend('x(t)', 'y(t)', 'Location', 'northwest');
42 figure;
43 plot(x, y, 'r—', 'lineWidth', 2);
44 ylim([0 100]);
45 hold on;
46 gt_y = 0:0.01:100;
47 plot(gt_x(gt_y, d, v1, v2), gt_y, 'b');
48 xlabel('x (m)');
49 ylabel('y (m)');
50 legend('Numerical', 'Analytical');
51
52 function result = dxy_dt(t, xy)
53 global d;
54 global v1;
55 global v2;
56 x = xy(1);
57 y = xy(2);
58 len = sqrt(x.^2 + (d-y).^2);
59 cos_theta = (d-y) ./ len;
60 sin_theta = x ./ len;
61 result = [v1 - v2 * sin_theta; v2 * cos_theta];
62 end

```

4.3 Chap4-Ex9

```

1 global r1;
2 global r2;
3 global n1;
4 global n2;
5 global s1;
6 global s2;
7
8 r1=1;r2=2;n1=400;n2=300;s1=1.5;s2=1.7;x0=100;y0=100;
9
10 [t, xy] = ode45(@dxy_dt, 0:0.1:20, [x0 y0]);
11 x = xy(:, 1);
12 y = xy(:, 2);
13
14 figure;
15 plot(t, x);
16 hold on;

```

```

17 plot(t, y);
18 legend('x(t)', 'y(t)');
19 xlabel('t');
20 ylabel('x(t), y(t)');
21
22 figure;
23 plot(x, y)
24 xlabel('x');
25 ylabel('y');
26
27 function result = dxy_dt(t, xy)
28     global r1;
29     global r2;
30     global n1;
31     global n2;
32     global s1;
33     global s2;
34     x = xy(1);
35     y = xy(2);
36     result = [r1*x*(1-x/n1-s1*y/n2); r2*y*(1-s2*x/n1-y/n2)];
37 end

```