数学实验实验报告

ZhaohengLi 2017050025
cainetatum@foxmail.com
15801206130

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 CH4-T5 放射性废物处理

2.1 建立模型

通过对海中正在下降的圆筒作受力分析，可以发现圆筒受重力 G，浮力 F 和阻力 f 的影响，设圆筒质量为 m，重力加速度为 g，则 G = mg，阻力与下沉速度比例系数为 α。当桶的下降速度为 v 时，列出受力方程如下:

$$\frac{dv}{dt} = \frac{G - F - \alpha v}{m}$$

对于初值，假设桶从速度为 0 开始自由下落，即 $v(0) = 0$。

2.2 解析解法

对于上述一阶非齐次线性微分方程，直接代入通解公式即可得到:

$$v = \frac{G - F}{\alpha}(1 - e^{-\frac{\alpha}{m}t})$$

此时，桶已经下降的深度为 (假设 h(0) = 0)：

$$h = \int_0^t v(\tau)d\tau = \frac{G - F}{\alpha}(t + \frac{m}{a}e^{-\frac{\alpha}{m}t}) - \frac{m(G - F)}{\alpha^2}$$

首先,代入题中所给的数据,根据上面导出的 v(t) 解析式反解出当 v = 40ft/s 时的时间。得到 t = 11.8243s。再将这个数值代入 h(t) 中求解下落深度，得到 h = 238.76ft。可以看出，当速度已经达到 40ft/s 时，桶还没有到达海底，因此桶会发生破裂。此外，也可以画出桶的下降速度和下落深度随时间的变化图像。从图像中可以明显地看出，当深度未达到 300 英尺时，速度已经超过 40 英尺每秒。因此桶会与海底冲撞而发生破裂，工程师赢得了官司。

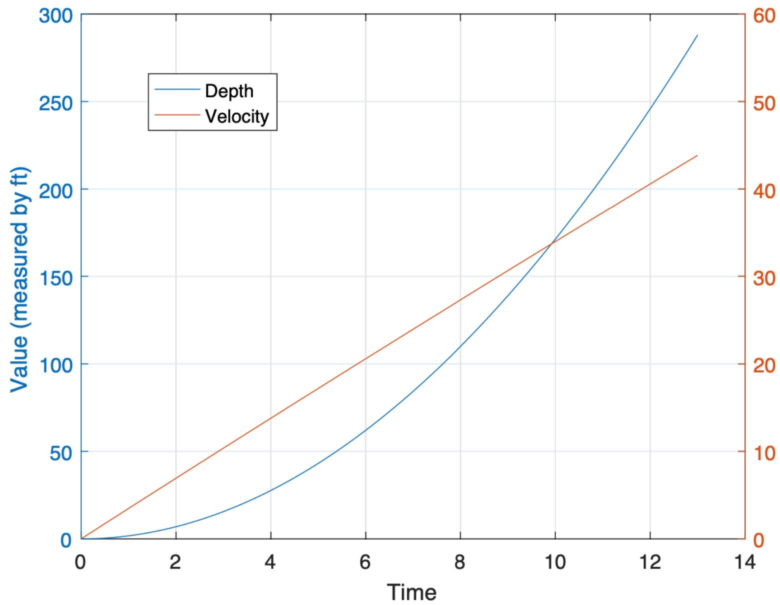

图 1: 桶的下降速度和下落深度随时间的变化图像 -解析解法

2.3 数值解法

根据模型可知，该微分方程可以使用龙格-库塔方法进行数值求解，且满足指定的精度需求。因此使用 MATLAB 中自带的 ode45 公式进行数值求解，并根据情况调整计算精度。此外，为了计算桶的下落高度，需要进行积分操作，这里使用的是上一个实验所学的 trap 函数进行计算。

MATLAB 程序如下：

```
%% Global variables
global m;
global G;
global F;
global alpha;
m = 527.436 * 0.4536;
g = 9.8;
G = m * g;
F = 470.327 * 0.4536 * 9.8;
alpha = 0.08 * 0.4536 * 9.8 / 0.3048;
depth_water = 300 * 0.3048;
speed_limit = 40 * 0.3048;
%% Runge Kutta Method for velocity estimation
ts = 0:0.1:13;
v0 = 0;
[t, v] = ode45(@vpartial, ts, v0);
%% Linear integral for depth integral
h = [0];
for i = 2:length(t)
h = [h; trapz(t(1:i), v(1:i))];
end
%% Plot the figure
[AX] = plotyy(t,h / 0.3048,t,v / 0.3048); grid on;
set(AX(1), 'yTick', 0:50:300);
set(AX(2), 'yTick', 0:10:60);
xlabel('Time');
ylabel('Value (measured by ft)');
legend('Depth', 'Velocity');

%% Fuction
function [ dx ] = vpartial( t, x )
% Global variables
global m;
global G;
global F;
global alpha;
% RHS of the equation
dx = (G - F - alpha * x) / m;
end
```

2.4 计算结果

程序执行之后，深度与速度随时间的变化如表中数据所示。画出的图像如图所示。可以看出，使用数值解法与解析解法所得到的图像基本相同。约 11.9 秒时，桶的下落速度达到 40ft/s，而此时下落高度约 240ft，没有到达海底，速度有效。

结论桶会与海底冲撞而发生破裂，工程师赢得了官司。

时间 (s)	速度 (ft/s)	下落高度 (ft)	时间 (s)	速度 (ft/s)	下落高度 (ft)
11.1	37.6155	210.6495	12.1	40.9054	249.9113
11.2	37.9452	214.4275	12.2	41.2335	254.0182
11.3	38.2748	218.2385	12.3	41.5615	258.158
11.4	38.6042	222.0825	12.4	41.8892	262.3305
11.5	38.9334	225.9594	12.5	42.2169	266.5358
11.6	39.2625	229.8691	12.6	42.5443	270.7739
11.7	39.5914	233.8118	12.7	42.8716	275.0447
11.8	39.9201	237.7874	12.8	43.1988	279.3482
11.9	40.2487	241.7959	12.9	43.5258	283.6844
12.0	40.5771	245.8372	13.0	43.8526	288.0533

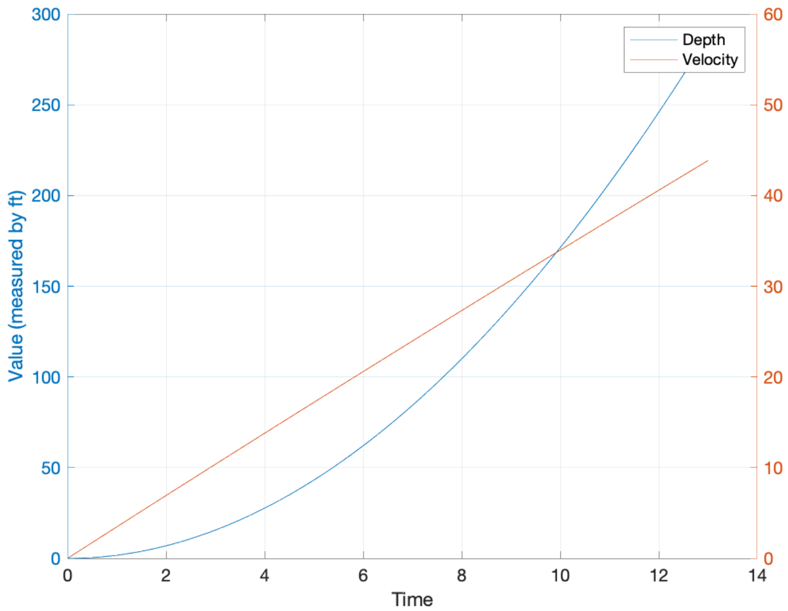

图 2: 桶的下降速度和下落深度随时间的变化图像-数值解法

3 CH4-T6 小船渡河

3.1 建立描述小船航线的数学模型，求其解析解；

首先建立平面直角坐标系，以 B 点为原点，正右方 (即河水流动方向) 为 x 轴正方向，正下方为 y 轴方向。在此坐标系下，A 点坐标为 (0, d)，B 点坐标为 (0, 0)，设小船坐标为 (x, y)。

将小船的航行速度 v 分解为河水流速 v1 和静水速度 v2，并设向量 (x, y) 与 x 轴夹角为 。则有下式成立：

$$\frac{dx}{dt} = v_1 - v_2 cos\theta$$

$$\frac{dy}{dt} = -v_2 sin\theta$$

消除 dt 得到：

$$\frac{dx}{dy} = \frac{k - cos\theta}{-sin\theta}$$

由 $sin\theta = \frac{y}{\sqrt{x^2+y^2}}$ 和 $cos\theta = \frac{x}{\sqrt{x^2+y^2}}$ 并设 $p = fracxy$，则有：

$$\frac{dx}{dy} = \frac{k - \frac{p}{\sqrt{1+p^2}}}{-\sqrt{1+p^2}} = p - k\sqrt{1+p^2}$$

$$\frac{dp}{dy} = \frac{\frac{dx}{dy}y - x}{y^2} = \frac{\frac{dx}{dy} - p}{y}$$

带入化简可得到：

$$y\frac{dp}{dy} + p = p - k\sqrt{1+p^2}$$

上述微分方程可以进行分离变量 y 和 p 进行求解：

$$-k\frac{dy}{y} = \frac{dp}{\sqrt{1+p^2}}$$

在两边同时积分：

$$-kln(Cy) = ln(p + \sqrt{p^2+1})$$

为了求出 p 和 y 的关系，上式两边同时求指数之后再正负相加，能导出 $\sqrt{1+p^2}$，p 和 y 的关系。最终化简结果为：

$$p = \frac{x}{y} = \frac{(Cy)^{-k} - (Cy)^k}{2}$$

代入初值 (p = 0, y = d)，能够解出 C = d/1 ，因此，小船航线的解析表达式为：

$$x = \frac{y}{2}[(\frac{y}{d})^{-k} - (\frac{y}{d})^k], \quad 0 < y \le d$$

当 y=0 时，小船到达了终点，这时有：

$$x = \lim_{y\to 0}\frac{1}{2}[(\frac{1}{d})^{-k}y^{-k+1} - (\frac{1}{d})^k y^{k+1}]$$

显然，当 k > 1 时，上式括号内的第一项在 y 趋于 0 时发散，最终小船的航线始终无法到达 (0, 0) 点，反而会越飘越远。这也和实际情况相符: 如果小船的速度不及静水的流速，那么无论航行多长时间，都无法到达终点。当 k = 1 时，x 最终收敛结果为 $d/2 \neq 0$，此时小船能够到达岸边，但是终点在 (d/2 , 0)。而当 k < 1 时，上式收敛，此时 x = 0，即小船能够到达终点。

MATLAB 代码及结果如下：

```
%% Analytical Solutions

% plot
figure;
subplot(1,3,1);
k=0.5;
x_ana = 0:0.1:d;
y_ana = (x_ana./2) .* ((x_ana./d).^(-k) - (x_ana./d).^k);
plot(y_ana,x_ana);
set(gca,'YDir','reverse')
title("k=0.5")

subplot(1,3,2);
k=1;
x_ana = 0:0.1:d;
y_ana = (x_ana./2) .* ((x_ana./d).^(-k) - (x_ana./d).^k);
plot(y_ana,x_ana);
set(gca,'YDir','reverse')
title("k=1")

subplot(1,3,3);
k=1.5;
x_ana = 0:0.1:d;
y_ana = (x_ana./2) .* ((x_ana./d).^(-k) - (x_ana./d).^k);
plot(y_ana,x_ana);
set(gca,'YDir','reverse')
title("k=1.5")
```

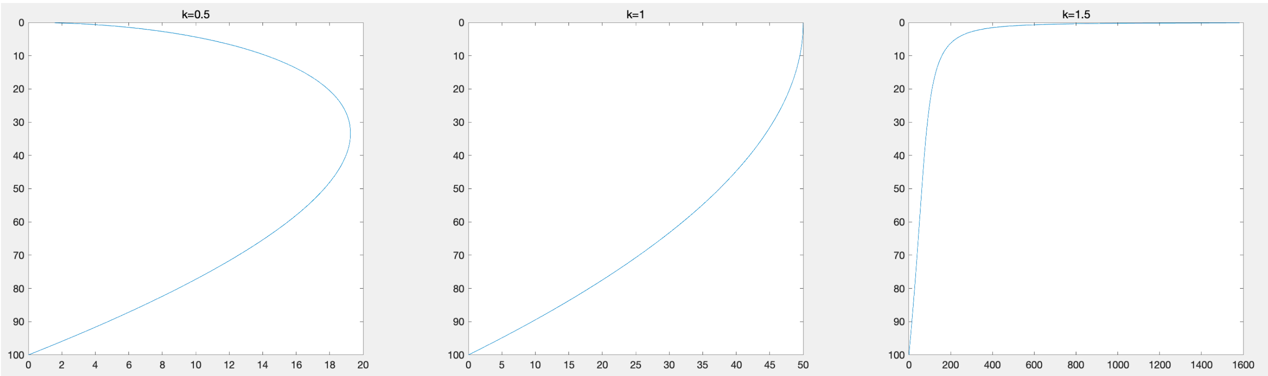

图 3: 不同 k 值下小船航行线路解析解图像

3.2 设 d=100 m，v1 =1 m/s，v2 = 2 m/s。用数值解法求渡河所需时间、任意时刻小船的位置及航行曲线，作图，并与解析解比较；

在小船航线和航行时间求解时，依然使用公式：

$$\frac{dx}{dt} = v_1 - v_2 cos\theta$$

$$\frac{dy}{dt} = -v_2 sin\theta$$

且 $sin\theta = \frac{y}{\sqrt{x^2+y^2}}$，$cos\theta = \frac{x}{\sqrt{x^2+y^2}}$。

给定时间精度之后，即可使用 MATLAB 中自带的 ode45 函数进行求解任意时刻的小船位置，将这些时刻的小船位置进行连接即可得到小船的航行曲线。当小船位置 $y = 0$ 时，即认为小船已经到达岸边，迭代结束。

MATLAB 代码如下：

```matlab
%% Global Variables
global v1;
global v2;
v1 = 1;
v2 = 2;
k = v1 / v2;
d = 100;

%% Analyical Method
x_ana = 0:0.1:d;
y_ana = (x_ana ./ 2) .* ((d ./ x_ana) .^ k- (x_ana ./ d) .^ k);

%% Runge Kutta Method
ts = 0:1:100;
[t, xy] = ode45(@xypartial, ts, [0, d]);
x = xy(:,1);
y = xy(:,2);
% Trim x,y for better plot
x = max(x, 0);
y = max(y, 0);

figure;
set (gcf,'Position',[100,100,760,320])
subplot(1,2,1);
[AX] = plotyy(t,x,t,y);
grid on;
xlabel('Time');
legend('x(t)', 'y(t)');
subplot(1,2,2);
hold on;
plot(x, y);
plot(y_ana, x_ana, 'b--', 'LineWidth', 2); xlabel('x');
ylabel('y');
set(gca,'ydir','reverse')
legend('y(x)数值解', 'y(x)解析解');

%% Function
function [ dx ] = xypartial( t, x )
global v1;
global v2;
norm = sqrt(x(1) ^ 2 + x(2) ^ 2);
dx = [v1 - v2 * x(1) / norm; - v2 * x(2) / norm];
end
```

时间 (s)	y(m)	数值 x(m)	解析 x(m)	时间 (s)	y(m)	数值 x(m)	解析 x(m)
0.0	100.0	0.0	0.0	48.0	12.78	15.649	15.59
4.0	92.002	3.8368	3.8358	52.0	8.12	13.152	13.091
8.0	84.018	7.3279	7.3245	56.0	4.3946	10.079	10.021
12.0	76.067	10.444	10.437	59.0	2.3	7.459	7.4085
16.0	68.172	13.151	13.14	60.0	1.7454	6.5375	6.4903
20.0	60.363	15.414	15.398	61.0	1.2651	5.596	5.5528
24.0	52.676	17.196	17.173	62.0	0.86066	4.6374	4.5987
28.0	45.16	18.455	18.427	63.0	0.53305	3.6645	3.6311
32.0	37.874	19.151	19.117	64.0	0.28315	2.6803	2.653
36.0	30.892	19.245	19.205	65.0	0.11154	1.6878	1.668
40.0	24.305	18.705	18.659	66.0	0.0185	0.69011	0.67995
44.0	18.225	17.508	17.455	67.0	0.0	0.0	0.0

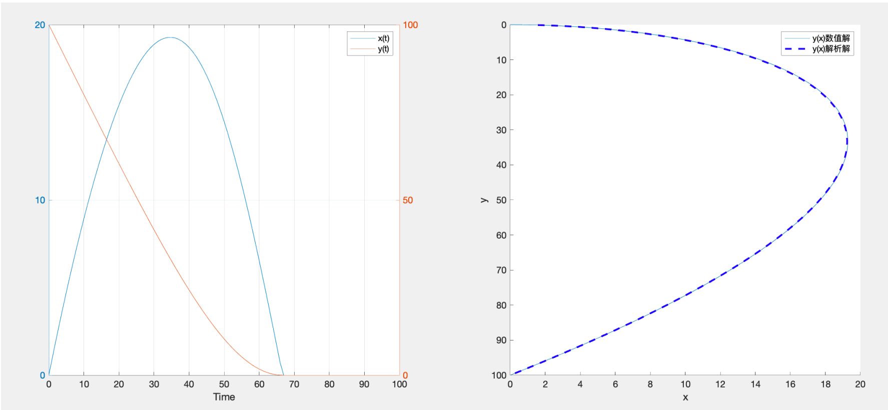

图 4: $v_1 = 1.0m/s$ 时小船航行线路数据

从图表中可以清晰看出，数值解和解析解的结果比较相近，最终求得小船渡河所需的时间为 67 秒。

3.3 若流速为 0, 0.5, 1.5, 2 (m/s)，结果将如何。

- $v_1 = 0.0m/s$ 此时静水速度为 0，小船理论上应该径直前进，最终求得小船渡河所需的时间为 50 秒。
- $v_1 = 0.5m/s$ 数值解和解析解的结果由于累计误差稍有差别但大体相近，最终求得小船渡河所需的时间为 53.5 秒。
- $v_1 = 1.5m/s$ 数值解和解析解的结果比较相近，最终求得小船渡河所需的时间为 114 秒。
- $v_1 = 2.0m/s$ 极限情况下小船会在离终点 50m 处的位置附近，这与解析解的分析是一致的。但是在数值求解的过程中，虽然 t 增大，但是小船始终与岸边有微小的距离，可以这样分析: 小船越靠近岸边，径向速度越慢，以至于最后所有的速度分量全部用于平衡水流速度。但无论如何，小船在这种情况下始终无法到达 B 点。

时间 (s)	y(m)	数值 x(m)	解析 x(m)	时间 (s)	y(m)	数值 x(m)	解析 x(m)
0.0	100.0	0.0	0.0	26.0	48.0	0.0	0.0
2.0	96.0	0.0	0.0	28.0	44.0	0.0	0.0
4.0	92.0	0.0	0.0	30.0	40.0	0.0	0.0
6.0	88.0	0.0	0.0	32.0	36.0	0.0	0.0
8.0	84.0	0.0	0.0	34.0	32.0	0.0	0.0
10.0	80.0	0.0	0.0	36.0	28.0	0.0	0.0
12.0	76.0	0.0	0.0	38.0	24.0	0.0	0.0
14.0	72.0	0.0	0.0	40.0	20.0	0.0	0.0
16.0	68.0	0.0	0.0	42.0	16.0	0.0	0.0
18.0	64.0	0.0	0.0	44.0	12.0	0.0	0.0
20.0	60.0	0.0	0.0	46.0	8.0	0.0	0.0
22.0	56.0	0.0	0.0	48.0	4.0	0.0	0.0
24.0	52.0	0.0	0.0	50.0	0.0	0.0	0.0

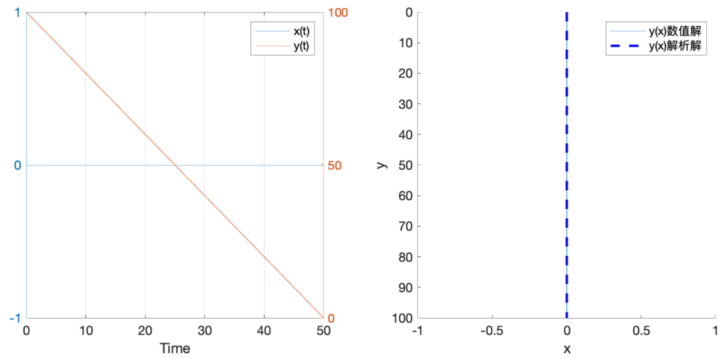

图 5: $v_1 = 0.0m/s$ 时小船航行线路数据

时间 (s)	y(m)	数值 x(m)	解析 x(m)	时间 (s)	y(m)	数值 x(m)	解析 x(m)
0.0	100.0	0.0	0.0	30.0	40.396	9.2506	9.2327
1.5	97.0	0.73874	0.73864	31.5	37.477	9.3079	9.2879
3.0	94.0	1.4543	1.4541	33.0	34.572	9.3092	9.2882
4.5	91.001	2.1463	2.1456	34.5	31.682	9.2519	9.2298
6.0	88.002	2.8134	2.8124	36.0	28.811	9.1322	9.1085
7.5	85.004	3.4555	3.4537	37.5	25.961	8.9437	8.9193
9.0	82.007	4.0709	4.0686	39.0	23.136	8.6828	8.6567
10.5	79.011	4.6591	4.6561	40.5	20.342	8.3409	8.3142
12.0	76.017	5.2194	5.2153	42.0	17.583	7.9114	7.8836
13.5	73.025	5.7498	5.7451	43.5	14.868	7.3846	7.3556
15.0	70.035	6.2503	6.2444	44.5	13.088	6.9725	6.9439
16.5	67.048	6.7189	6.7119	45.5	11.335	6.5086	6.4791
18.0	64.065	7.154	7.1463	46.5	9.6166	5.9857	5.9569
19.5	61.085	7.5557	7.5463	47.5	7.9367	5.3995	5.3702
21.0	58.11	7.9207	7.9103	48.5	6.3068	4.7405	4.7123
22.5	55.14	8.2477	8.2365	49.5	4.7383	3.9997	3.9726
24.0	52.175	8.5366	8.5233	50.5	3.2515	3.1621	3.1382
25.5	49.217	8.7827	8.7686	51.5	1.8789	2.2082	2.1896
27.0	46.268	8.9854	8.9702	52.5	0.68658	1.1046	1.0938
28.5	43.326	9.1431	9.1258	53.5	0.0	1.0786e-08	0.0

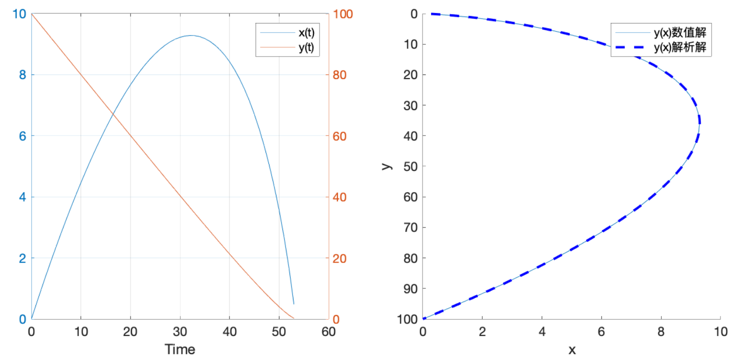

图 6: $v_1 = 0.5m/s$ 时小船航行线路数据

时间 (s)	y(m)	数值 x(m)	解析 x(m)	时间 (s)	y(m)	数值 x(m)	解析 x(m)
0.0	100.0	0.0	0.0	90.0	0.34391	12.161	12.106
5.0	90.009	7.1147	7.1129	95.0	0.13664	9.6637	9.6126
10.0	80.083	13.408	13.402	99.0	0.053902	7.6641	7.6184
15.0	70.307	18.805	18.794	100.0	0.04112	7.1642	7.12
20.0	60.788	23.243	23.225	101.0	0.03076	6.6642	6.6216
25.0	51.652	26.677	26.653	102.0	0.022495	6.1642	6.1233
30.0	43.04	29.094	29.063	103.0	0.016023	5.6642	5.6254
35.0	35.096	30.52	30.483	104.0	0.011058	5.1642	5.1273
40.0	27.948	31.026	30.983	105.0	0.0073499	4.6642	4.6296
45.0	21.695	30.725	30.676	106.0	0.004663	4.1642	4.1318
50.0	16.383	29.755	29.701	107.0	0.0027912	3.6642	3.6343
55.0	12.008	28.266	28.208	108.0	0.0015496	3.1642	3.1371
60.0	8.5147	26.399	26.338	109.0	0.00077713	2.6642	2.6399
65.0	5.8162	24.271	24.21	110.0	0.00033735	2.1642	2.1429
70.0	3.8021	21.978	21.915	111.0	0.00011713	1.6642	1.6449
75.0	2.3573	19.583	19.521	112.0	2.7407e-05	1.1642	1.144
80.0	1.3677	17.133	17.072	113.0	2.5694e-06	0.66424	0.63304
85.0	0.72778	14.654	14.595	114.0	0.0	0.16424	0

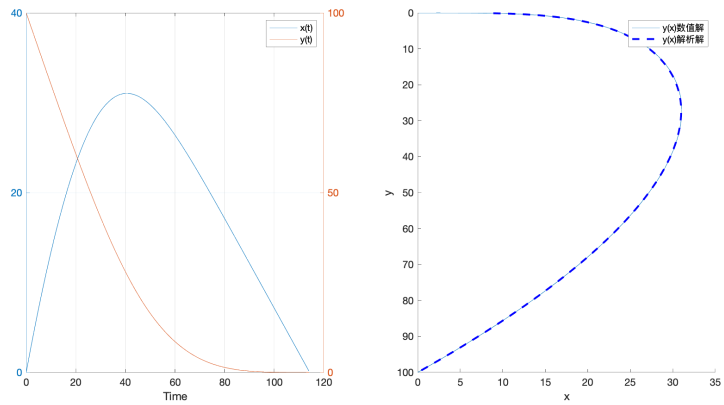

图 7: $v_1 = 1.5m/s$ 时小船航行线路数据

时间 (s)	y(m)	数值 x(m)	解析 x(m)	时间 (s)	y(m)	数值 x(m)	解析 x(m)
0.0	100.0	0.0	0.0	136.0	0.70705	50.005	49.998
8.0	84.073	14.663	14.659	144.0	0.5131	50.006	49.999
16.0	68.657	26.44	26.431	152.0	0.37224	50.007	49.999
24.0	54.401	35.216	35.203	160.0	0.27006	50.007	50.0
32.0	41.915	41.224	41.216	168.0	0.196	50.007	50.0
40.0	31.587	45.022	45.011	176.0	0.14221	50.007	50.0
48.0	23.438	47.265	47.253	184.0	0.10316	50.008	50.0
56.0	17.225	48.525	48.516	192.0	0.074856	50.008	50.0
64.0	12.585	49.216	49.208	200.0	0.054325	50.008	50.0
72.0	9.1642	49.588	49.58	208.0	0.039413	50.008	50.0
80.0	6.6636	49.786	49.778	216.0	0.028593	50.008	50.0
88.0	4.8405	49.89	49.883	224.0	0.020751	50.008	50.0
96.0	3.5135	49.946	49.938	232.0	0.015057	50.008	50.0
104.0	2.5499	49.975	49.967	240.0	0.010923	50.008	50.0
112.0	1.8509	49.99	49.983	248.0	0.0079251	50.008	50.0
120.0	1.343	49.999	49.991	256.0	0.0057517	50.008	50.0
128.0	0.97432	50.003	49.995	264.0	0.0041731	50.008	50.0

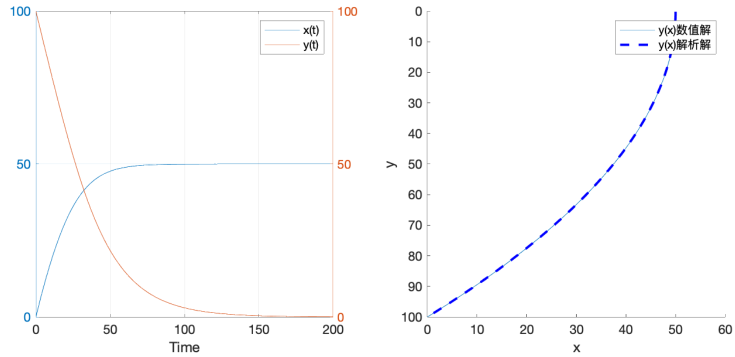

图 8: $v_1 = 2.0m/s$ 时小船航行线路数据

4 CH4-T9 种群竞争

4.1 模型建立

题目中已经建立模型。

4.2 算法设计与代码实现

ODE 已由题目中给出，使用 5 级 4 阶龙格—库塔公式计算，并控制相对误差 1e-6，绝对误差 1e-9。MATLAB 代码如下：

```
%% Gloabal Variables
%% Gloabal Variables
global r1;
global r2;
global n1;
global n2;
global s1;
global s2;
r1 = 1;
r2 = 1;
n1 = 100;
n2 = 100;
s1 = 0.5;
s2 = 2;

%% Solve
ts=[0:0.1:30];%时间区间
x0=[10,10];%初始条件
opt=odeset('reltol',1e-6,'abstol',1e-9);%相对误差1e-6，绝对误差1e-9
[t,x]=ode45(@odeFunc,ts,x0,opt);

figure;
grid on;

subplot(1,2,1);
hold on;
plot(t,x(:,1),'LineWidth',2);
gtext('x(t)');
plot(t,x(:,2),'LineWidth',2);
gtext('y(t)');
hold off;

subplot(1,2,2);
plot(x(:,1),x(:,2),'LineWidth',2);
xlabel('x');
ylabel('y');


%% Function
function dx = odeFunc(t, x)
global r1;
global r2;
global n1;
```

```
global n2;
global s1;
global s2;
dx = [r1*x(1)*(1-x(1)/n1-s1*x(2)/n2); r2*x(2)*(1-s2*x(1)/n1-x(2)/n2)];
end
```

4.3 (1)

在 $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$ 的情况下，x(t) 图形，y(t) 图形及相图 (x, y) 如下图所示：

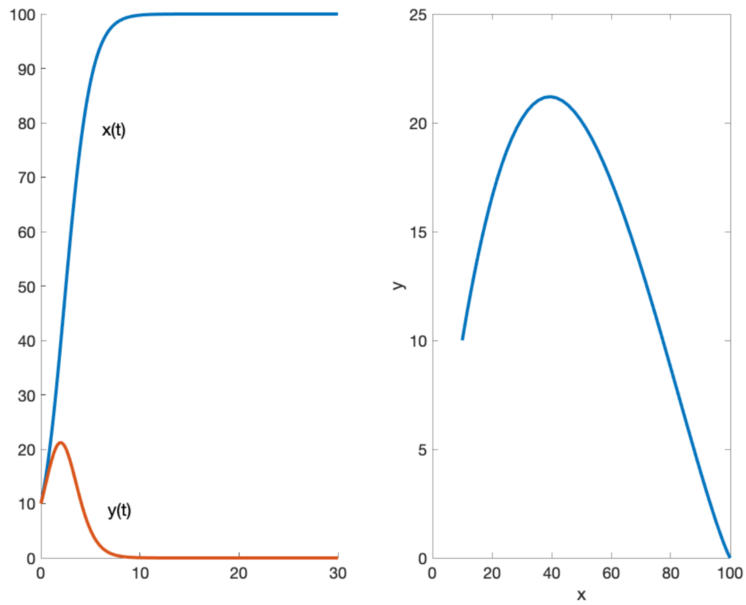

图 9: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$

图中可以看出最后数值稳定在 x=100,y=0 上，即物种甲达到最大值，物种乙灭绝。

4.4 (2)

以第一小题为基准，设置 $r_1 = r_2 = 0.3$ 得到如下图形：

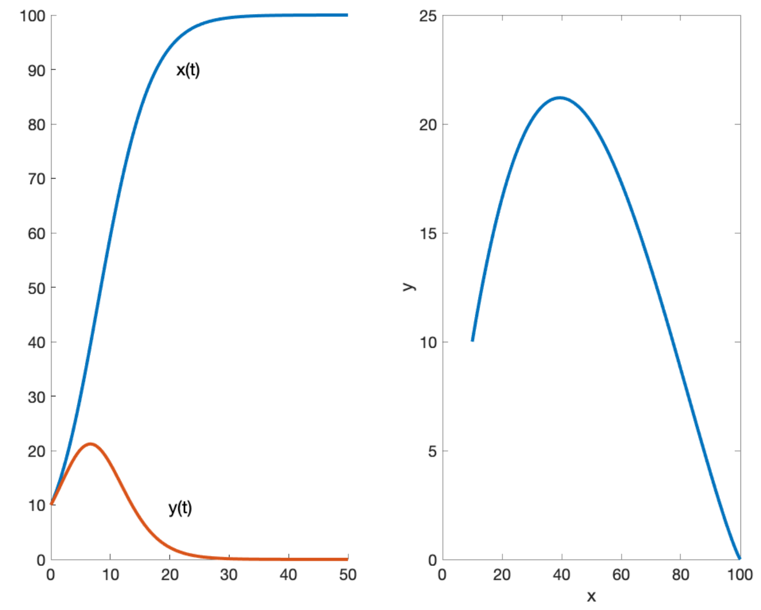

图 10: $r_1 = r_2 = 0.3, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$

我们可以看到甲乙两物种最终结果仍然是甲达到数量极限而乙灭绝，但与原先不同的是变化速度减缓了，这是由于自然增长率 r1,r2 变小的缘故（相当于变化率减小）。

以第一小题为基准，设置 $n_1 = 10000, n_2 = 100$ 得到如下图形：

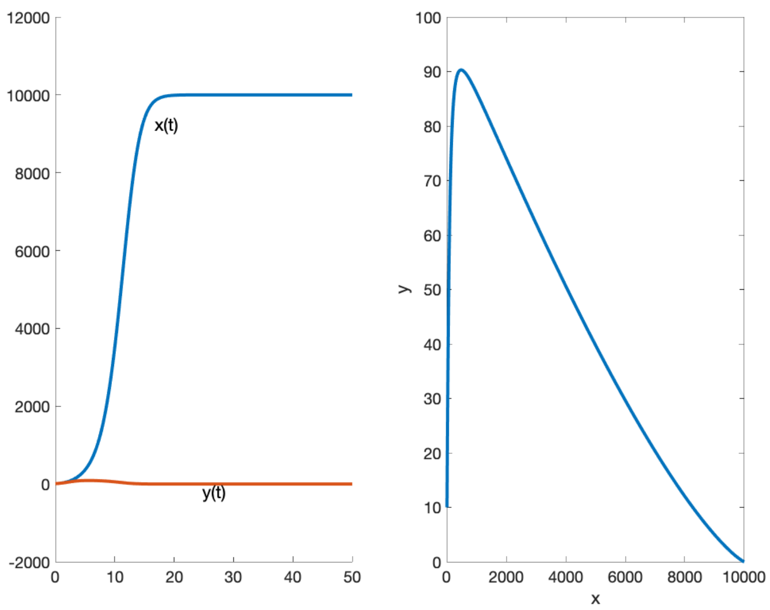

图 11: $r_1 = r_2 = 1, n_1 = 10000, n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$

由于一开始甲物种的数量相对较少，所以乙物种得以快速增长，数量一度达到 90 以上，但最终仍然灭绝。物种容量的改变并不能影响最终谁会灭绝.

以第一小题为基准，设置 $x_0 = 10, y_0 = 100$ 得到如下图形：

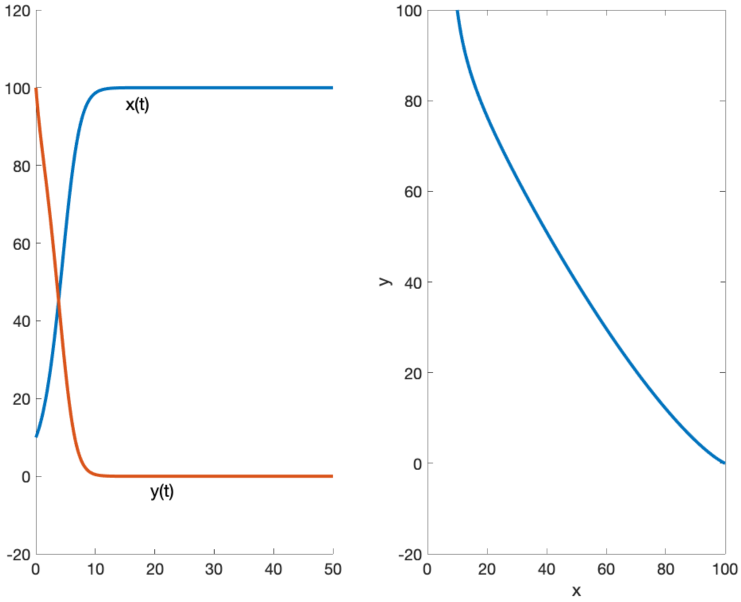

图 12: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = 10, y_0 = 100$

乙物种的初始数量大使其灭绝时间稍稍延后，但它灭绝的趋势不变。

综上，无论怎样改变 $r_1, r_2, n_1, n_2, x_0, y_0$**，都改变不了最后甲物种存活并达到数量最大且乙物种灭绝的结果**。

以第一小题为基准，设置 $s_1 = 1.5, s_2 = 0.7$ 得到如下图形：

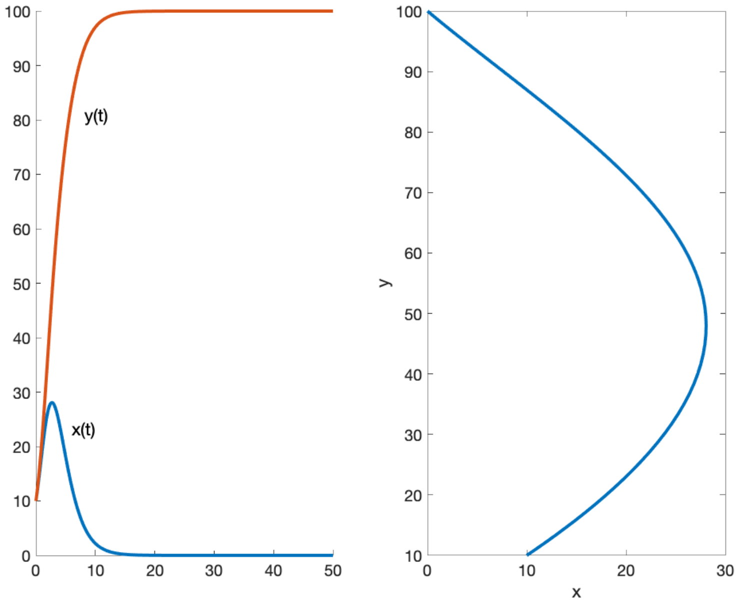

图 13: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 1.5, s_2 = 0.7, x_0 = y_0 = 10$

最后甲物种灭绝，乙物种存活并达到数量极限。

所以 s_1, s_2 对两物种在自然界长期演变的结局起决定作用。下面在第三小题中进一步讨论 s_1, s_2 对物种的影响。

4.5 (3)

设置 $s_1 = 0.8(< 1), s_2 = 0.7(< 1)$ 得到如下图形：

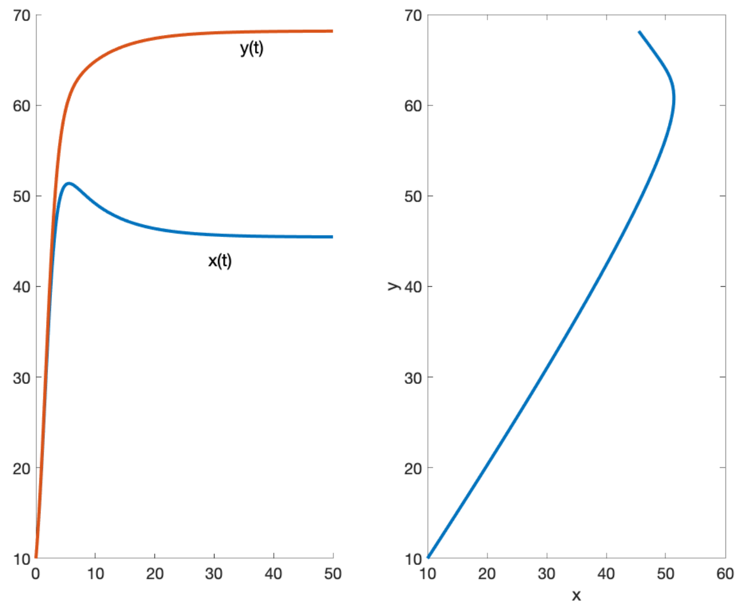

图 14: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.8, s_2 = 0.7, x_0 = y_0 = 10$

最后稳定在了 $x = 45, y = 68$。

设置 $s_1 = 1.5(> 1), s_2 = 1.7(> 1)$ 得到如下图形：

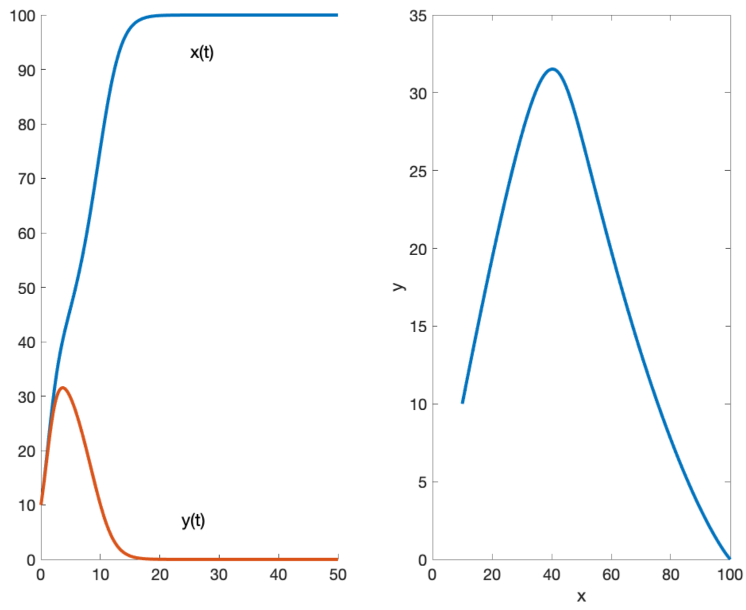

图 15: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 1.5, s_2 = 1.7, x_0 = y_0 = 10$

虽然 s_1, s_2 都大于 1，但是 s_2 更大，严重消耗了乙物种的生存资源，使乙物种在竞争中灭绝。设置 $s_1 = s_2 = 1.6(> 1)$ 得到如下图形：

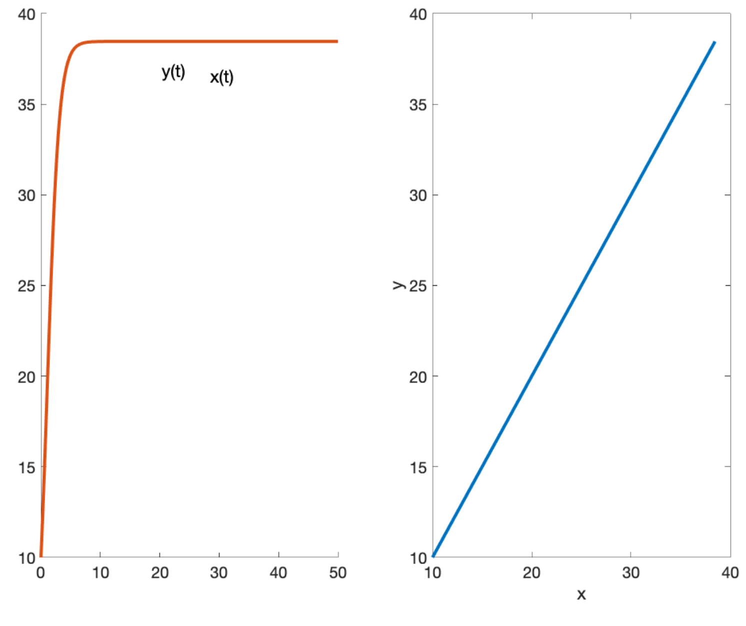

图 16: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = s_2 = 1.6, x_0 = y_0 = 10$

设置 $s_1 = s_2 = 0.6(< 1)$ 得到如下图形：

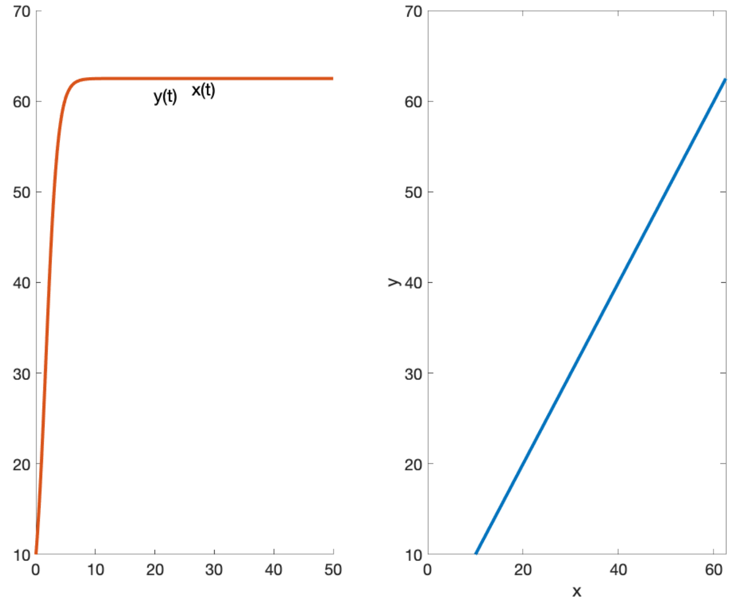

图 17: $r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = s_2 = 0.6, x_0 = y_0 = 10$

当 s_1, s_2 相等的时候，两者可以共生。当 $s_1, s_2 > 1$ 时，代表两物种消耗资源比较大，最终稳定的物种数量要小于 $s_1, s_2 < 1$ 的时候。

结论

当 $s_1, s_2 < 1$ 时，消耗生存资源的严重程度较轻，所以甲乙物种可以共存，但两者都达不到最大值；

当 $s_1, s_2 > 1$ 时，两物种竞争激烈，最后 s1,s2 中更大者对应作用的物种灭绝。所谓物尽天择，自然资源是有限的，需要更少资源就能生存的物种在竞争中将占有优势。

当其中之一大于 1 时，对应作用的物种就会由于生存资源的过度消耗而灭绝；

当 $s_1 = s_2$ 时，都大于 1 但相等时，由于方程的对称性，甲乙两物种都能生存下来，但都不能达到最大值。

5 收获和建议

通过这次的实验，我对 MATLAB 中提供微分方程求解函数理解更加深刻，通过实际编程、画图的方式观察了方程求解的结果，这是书本上无法学到的知识。同时，在做上机实验的过程中，我对 MATLAB 这款软件的使用也更加熟练了。希望在之后的课堂上老师能够当堂进行相关的技巧演示并给出题目的分步解答。