数学实验第二次实验报告

计 76　张翔　2017011568

2020 年 3 月 17 日

1　实验目的

1) 掌握用 MATLAB 软件求微分方程初值问题数值解的方法;
2) 通过实例学习用微分方程模型解决简化的实际问题;
3) 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;
4) 练习数值微分的计算。

2　实验题目

2.1　Ch4-P5 放射性废物的处理

2.1.1　问题分析与模型建立

设圆桶质量为 m，重力加速度为 g，圆桶受浮力 F 和阻力 f 以及重力 $G = mg$ 共同作用下的运动方程如下

$$m\frac{\mathrm{d}v}{\mathrm{d}t} = mg - F - f = mg - F - kv \tag{1}$$

式子中 k 为阻力与下沉速度的比例系数。初始时，可以认为桶由静止状态释放，$v(0) = 0$，由此可以解出 $v(t)$，并进一步得到 $v(t) = 40$ 时的时刻 t，比较位移 $\int_0^t v(t)dt$ 与海的深度即可判断圆桶是否会破裂。

2.1.2　解析求解

方程 (1) 是一阶齐次 ODE，直接分离变量，代入初值，可解得

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

初始位移为 $x(0) = 0$，则 t 时刻位移为

$$x(t) = \int_0^t v(\tau)d\tau = \frac{mg - F}{k}t + \frac{m(mg - F)}{k^2}\left(e^{-\frac{k}{m}t} - 1\right) \tag{3}$$

代入题中所给数据, 利用 (2) 解出 $v = 40$ ft/s 时, $t = 11.4935$ s, 再利用 (3) 得到 $x(t) = 229.937$ ft $<$ 300 ft，可见圆桶尚未沉底速度就已经超过阈值，因此会破裂。

2.1.3　数值求解

算法设计　容易观察题给微分方程不是刚性方程，可以使用 Matlab 的 Runge-Kutta 方法 (`ode45`) 求解。为了计算桶的下落高度，还需要进行积分操作，这里可以使用 Matlab 的 `trapz` 函数求解。解完之后，得到的是位移与速度的离散点，可以使用三次样条插值 `spline` 获取插值函数，它是位移对速度函数 $x(v)$ 的近似，代入 $v = 40$ 得到此时的位移，与 300 比较即可。

Matlab 程序 代码如下:

```
%% MathExp Ch4.5
global m;
global g;
global F_m;
global k;
m = 527.436;
g = 32.17; %% gravitational constant in british unit ft/s^2
F_m = 470.327;
k = 0.08;

max_depth = 300;
speed_thres = 40;

t = 0:0.1:15;
v0 = 0; %% initial speed
%% solve ODE
[t, v] = ode45(@vel, t, v0);

%% solve depth
h = zeros([length(t), 1]);
for i = 2:length(t)
    h(i) = trapz(t(1:i), v(1:i));
end

%% get approx result
approx_h_v = spline(v, h);
approx_h = ppval(approx_h_v, speed_thres); %% approximate time to reach
    threshold speed
if(approx_h < max_depth)
    fprintf("The bucket filled with nuclear waste will break!\n");
else
    fprintf("The bucket will not break.\n")
end

%% plot
yyaxis left
plot(t, v)
ylabel('Velocity(ft/s)')
yyaxis right
plot(t, h)
ylabel('Depth(ft)')
xlabel('Time(s)');

```

```
%% function for velocity vs accel
function [dx] = vel(t, x)
    global g;
    global F_m;
    global k;
    global m;
    dx = g - (F_m * g + k * x) / m;
end
```

计算结果 计算结果如下:

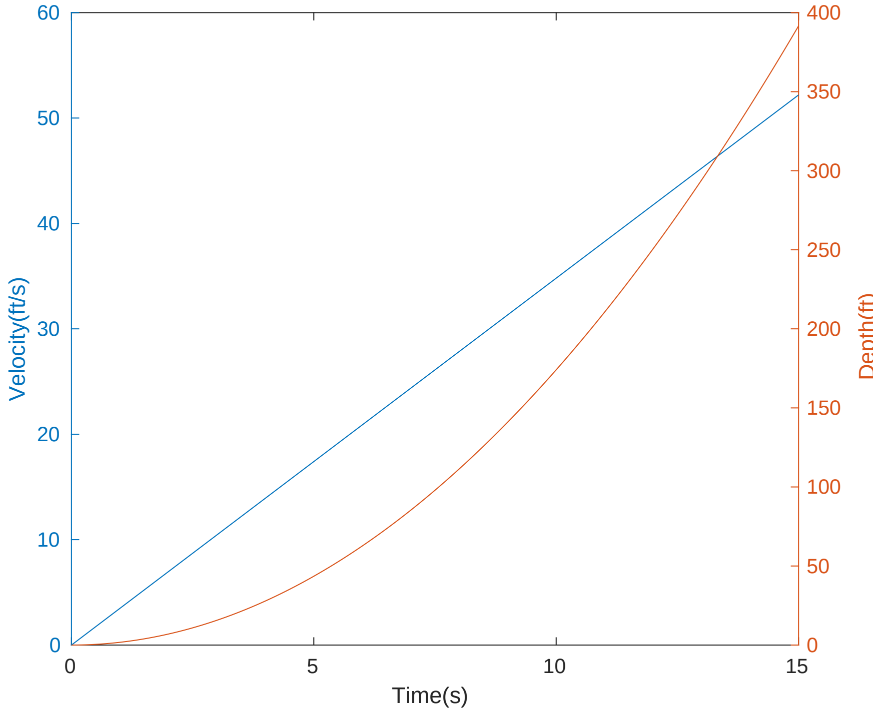


图 1: 速度/位移关于时间的图像

表 1: 在阈值速度附近的时间/速度/位移计算值

Time(s)	**Velocity(ft/s)**	**Depth(ft)**	**Time(s)**	**Velocity(ft/s)**	**Depth(ft)**
11.1	38.6317	214.466	11.5	40.0226	230.197
11.2	38.9794	218.346	11.6	40.3703	234.216
11.3	39.3271	222.262	11.7	40.7180	238.271
11.4	39.6748	226.212	11.8	41.0657	242.360

对得到的离散数据进行三次样条插值，代入 $v = 40$ 可直接得到位移 $x = 229.937$，与上面的解析解相同。

2.1.4 结果分析

从上述结果可以看出，在 Matlab 预设条件下，数值计算得到了和解析计算一样的结果，说明数值计算方法的可行性。由于速度到达阈值时，位移只有 229.937 ft，还未到达海底，因此最后圆桶会破裂。

2.1.5 结论

桶会因为与海底冲撞发生破裂，工程师赢得了官司。

2.2 Ch4-P6 小船过河

2.2.1 模型建立

起点 A 为原点建立平面直角坐标系，正右方向为 x 轴正方向，AB 方向为 y 轴正方向，则终点坐标为 $(0, d)$，小船坐标设为 (x, y)。将小船速度沿轴向分解，令坐标为 $(x, y)-(0, d)$ 的向量（就是小船船速的方向）与 x 轴负向夹角为 θ ($\theta \in [0, \frac{\pi}{2}]$)，水速、船速分别为 $v_1, v_2 (v_1 = kv_2)$，可以得到如下方程：

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = v_1 - v_2\cos\theta \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = v_2\sin\theta \end{cases} \tag{4}$$

利用几何关系可得

$$\tan\theta = \frac{d-y}{x} \tag{5}$$

2.2.2 解析求解

利用 (4)(5) 消去 t, θ 得到

$$\frac{dx}{dy} = \frac{v_1\sqrt{(d-y)^2+x^2} - v_2 x}{v_2(d-y)} \tag{6}$$

上式令 $p = \frac{x}{d-y}$，而

$$\frac{\mathrm{d}p}{\mathrm{d}y} = \frac{\frac{\mathrm{d}x}{\mathrm{d}y} + p}{d-y}$$

代入得

$$(d-y)\frac{\mathrm{d}p}{\mathrm{d}y} = k\sqrt{1+p^2} \tag{7}$$

分离变量之后两边积分，利用初值 $x = 0, y = 0, p = 0$ 可得

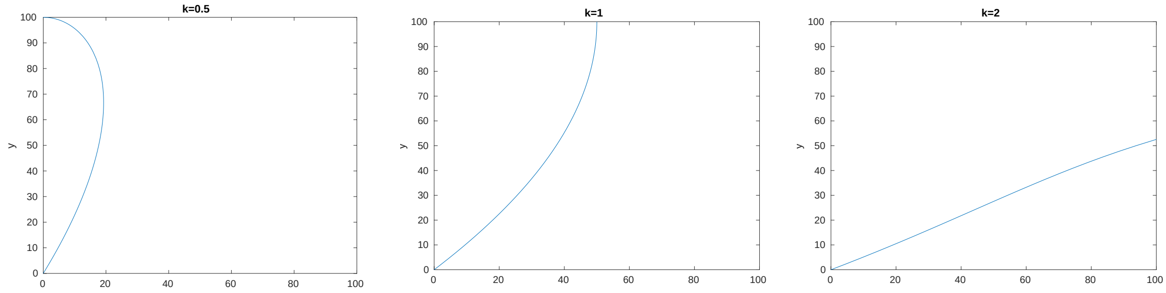
$$\ln\left(p + \sqrt{1+p^2}\right) = k\ln\left(\frac{d}{d-y}\right) \tag{8}$$

最后得到小船轨迹的解析式

$$x = \frac{d-y}{2}\left[\left(\frac{d}{d-y}\right)^k - \left(\frac{d}{d-y}\right)^{-k}\right] = \frac{1}{2}\left[d^k(d-y)^{-k+1} - d^{-k}(d-y)^{k+1}\right] \quad y \in [0, d) \tag{9}$$

注意到 $-k+1 < 0$ 即 $k > 1$ 时上式在 $y \to d$ 时的极限不存在，表示水速大于船速时，无论如何也不可能到达 B 点，这是与实际经验相符的；$k = 1$ 时，$\lim_{y\to d} x = \frac{d}{2}$，也到不了 B 点；$k < 1$ 时，$\lim_{y\to d} x = 0$，可以到达 B 点。

将上述解析式在 $k = 0.5, 1, 2$ 时的结果画图，如下：



(a) $k = 0.5$　　(b) $k = 1$　　(c) $k = 2$

图 2: 小船轨迹

2.2.3 数值求解

算法设计　利用式 (4)(5) 消去 θ，得到 ODE 方程组：

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = v_1 - v_2 \dfrac{x}{\sqrt{(d-y)^2 + x^2}} \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = v_2 \dfrac{d-y}{\sqrt{(d-y)^2 + x^2}} \end{cases} \tag{10}$$

初始条件 $x(0) = y(0) = 0$，上述方程组可以使用 Matlab 自带的函数进行求解，其中 t 的范围可以根据通过试验得出，这里取 $[0, 80]$。在实际实验过程中，使用 `ode45` 的速度非常慢，说明上述方程适合使用刚性方程的求解函数 `ode23s`，换用该函数后求解速度正常。

Matlab 程序　如下

```
global v1;
global v2;
global d;
v1 = 1;
v2 = 2;
d = 100;
k = v1/v2;

%% solve ode
t = 0:1:80;
[t, xy] = ode23s(@boat_traj, t, [0 0]);

%% analytical results
y_a = 0:0.1:d;
x_a = 1 / 2 * (d.^k .* (d-y_a).^(-k+1) - d.^(-k) .* (d-y_a).^(k+1));

xy = max(xy, 0); %% remove negative vals on edge

%% plot x(t) y(t)
figure(1);
yyaxis left;
plot(t, xy(:, 1))
```

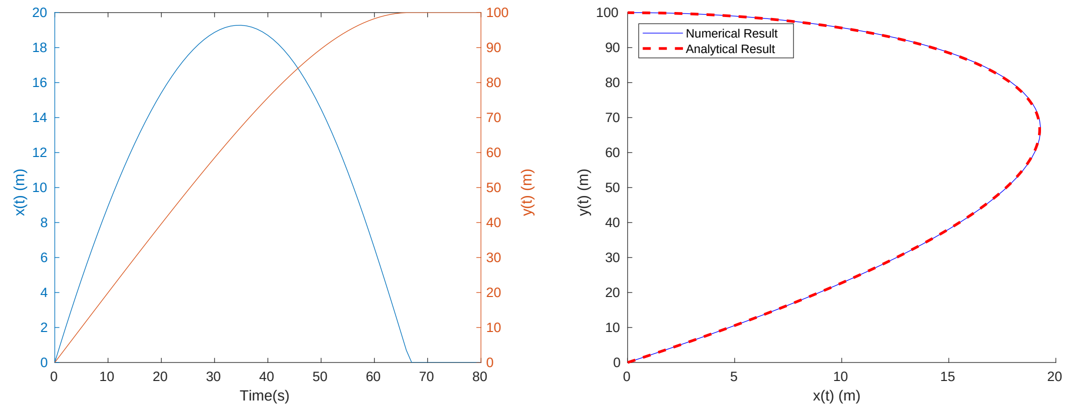
```
ylabel('x(t) (m)');
yyaxis right;
plot(t, xy(:, 2))
ylabel('y(t) (m)');
xlabel('Time(s)');

figure(2);
hold on;
plot(xy(:, 1), xy(:, 2), 'b-', 'DisplayName', 'Numerical Result');
plot(x_a, y_a, 'r--', 'DisplayName', 'Analytical Result', 'LineWidth', 2);
xlabel('x(t) (m)');
ylabel('y(t) (m)');
legend('show','Location','NorthWest');

%% trajectory ode for the boat
function dx = boat_traj(t, x)
    global d;
    global v1;
    global v2;
    denom = sqrt((d-x(2))^2+x(1)^2);
    dx = [v1-v2*x(1)/denom; v2*(d-x(2))/denom];
end
```

计算结果与分析

- $v_1 = 1$ m/s 的情况，由 Matlab 程序内可以读取出 $t = 67$ s 时到达对岸

表 2: 部分计算结果示例 ($x'(t)$ 表示 x 的数值解，$x(t)$ 表示解析解)

Time(s)	y(t) (m)	x'(t) (m)	x(t) (m)	Time(s)	y(t) (m)	x'(t) (m)	x(t) (m)
0.0000	0.0000	0.0000	0.0000	40.0000	75.6947	18.7052	18.6589
5.0000	9.9958	4.7429	4.7415	45.0000	83.2007	17.1051	17.0507
10.0000	19.9628	8.9349	8.9297	50.0000	89.6598	14.4792	14.4156
15.0000	29.8610	12.5147	12.5042	55.0000	94.7828	10.9016	10.8248
20.0000	39.6374	15.4144	15.3978	60.0000	98.2803	6.5451	6.4440
25.0000	49.2205	17.5606	17.5372	65.0000	99.9143	1.6947	1.4622
30.0000	58.5149	18.8749	18.8444	70.0000	100.0000	0.0000	0.0000

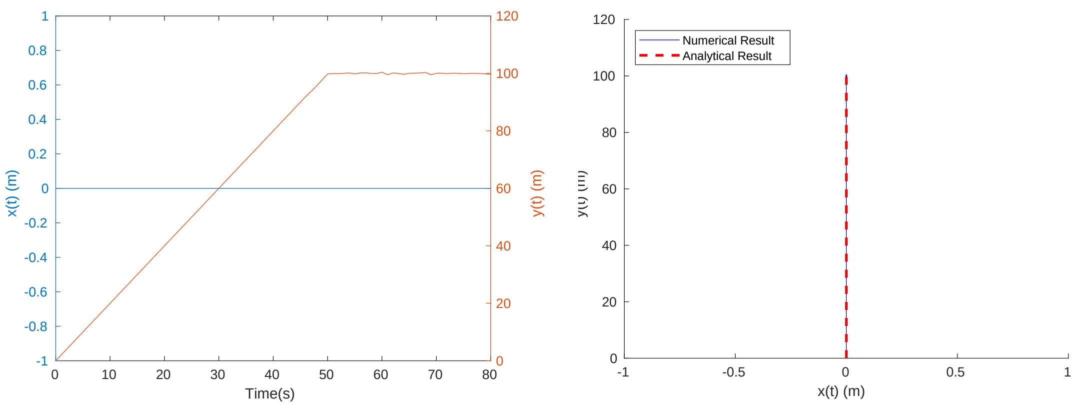


(a) $x(t)$, $y(t)$ 图线　　(b) $y(x)$ 图线解析与数值解比较

图 3: $v_1 = 1$ m/s 时的图线

（注：下面几种情形的计算结果表格与 $v_1 = 1$ m/s 的情况类似，故略去）

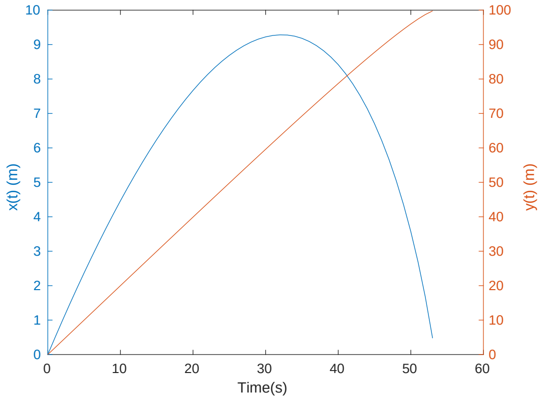
- $v_1 = 0$ 的情况，此时小船沿直线轨迹直接到达对岸，从 Matlab 读取出数值解的结果是 $t = 50$ s 到达对岸



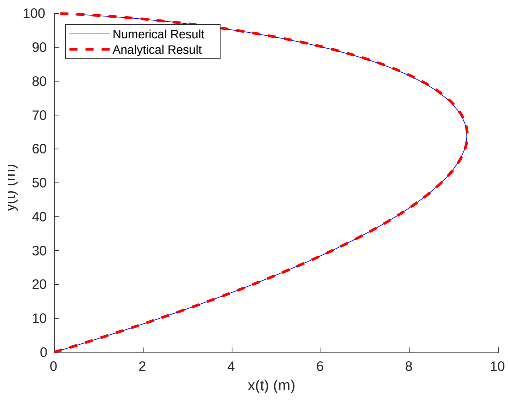
(a) $x(t)$, $y(t)$ 图线　　(b) $y(x)$ 图线解析与数值解比较

图 4: $v_1 = 0$ 时的图线

- $v_1 = 0.5$ m/s 的情况，此时小船沿直线轨迹直接到达对岸，从 Matlab 读取出数值解的结果是 $t = 54$ s 到达对岸



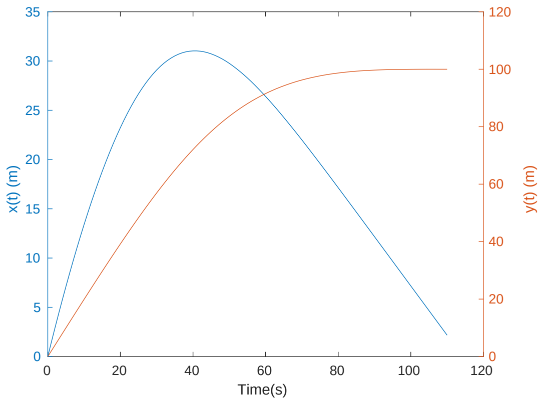
(a) $x(t)$, $y(t)$ 图线



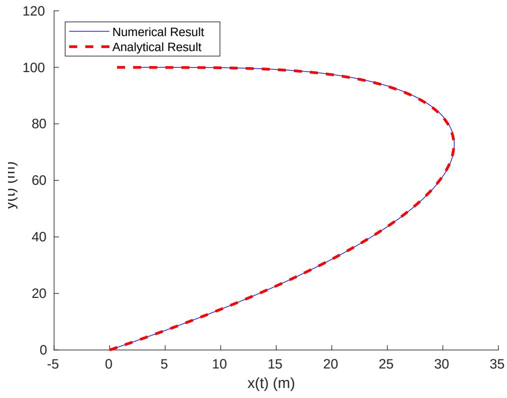
(b) $y(x)$ 图线解析与数值解比较

图 5: $v_1 = 0.5$ m/s 时的图线

- $v_1 = 1.5$ m/s 的情况，从 Matlab 读取出数值解的结果是 $t = 103$ m 到达对岸



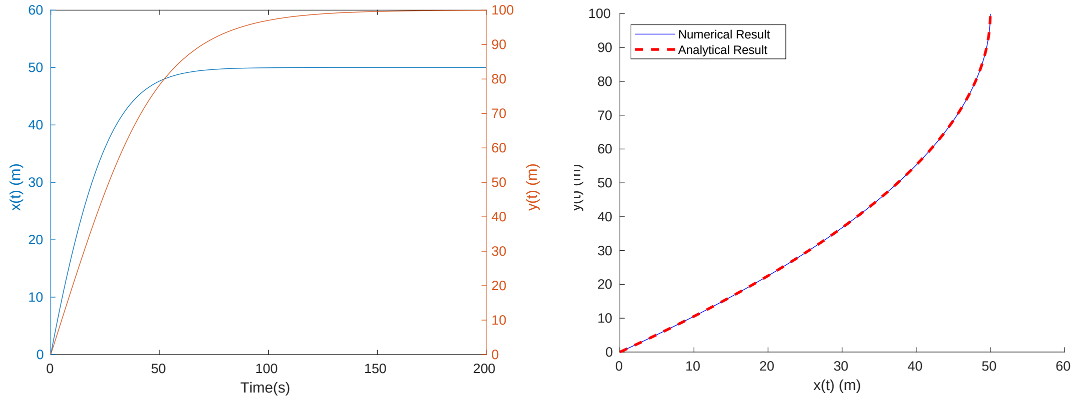
(a) $x(t)$, $y(t)$ 图线



(b) $y(x)$ 图线解析与数值解比较

图 6: $v_1 = 1.5$ m/s 时的图线

- $v_1 = 2$ m/s 的情况，此时小船无论如何也不能达到对岸 B 点，极限位置只能到达距离 B 50m 处，这与理论计算结果是一致的。接近岸边时，小船的速度几乎全部被用来平衡水流速度，只剩下微小的 y 分量，因此呈现出图中的轨迹。



(a) $x(t)$, $y(t)$ 图线　　(b) $y(x)$ 图线解析与数值解比较

图 7: $v_1 = 2$ m/s 时的图线

从上述图表可以看出，数值解和解析解的结果比较接近，也符合客观规律。当船速 > 水速时，小船可以到达 B 点，且水速越小，x 方向位移越小；其他情况下小船则无法到达对岸 B 点。

2.2.4 结论

使用 Lunge-Kutta 方法能对小船过河问题的微分方程进行数值求解。具体结果请参考上面。

2.3 Ch4-P9 种群竞争

2.3.1 模型建立

题目已经建立了两种群的相互竞争模型，方程如下

$$\begin{cases} \dot{x}(t) = r_1 x \left(1 - \dfrac{x}{n_1} - s_1 \dfrac{y}{n_2}\right) \\ \dot{y}(t) = r_2 y \left(1 - s_2 \dfrac{x}{n_1} - \dfrac{y}{n_2}\right) \end{cases} \tag{11}$$

容易发现，这个模型其实是 Logistic 模型加上了物种之间的影响因子，如 $s_1 \frac{y}{n_2}$ 项，它通过系数 s_1 将物种乙对甲的生存资源的消耗等效转化为甲种群对自己生存资源的消耗。

2.3.2 算法设计

上述方程可以利用 Lunge-Kutta 方法进行数值求解，时间 t 的范围可以通过试验给出。

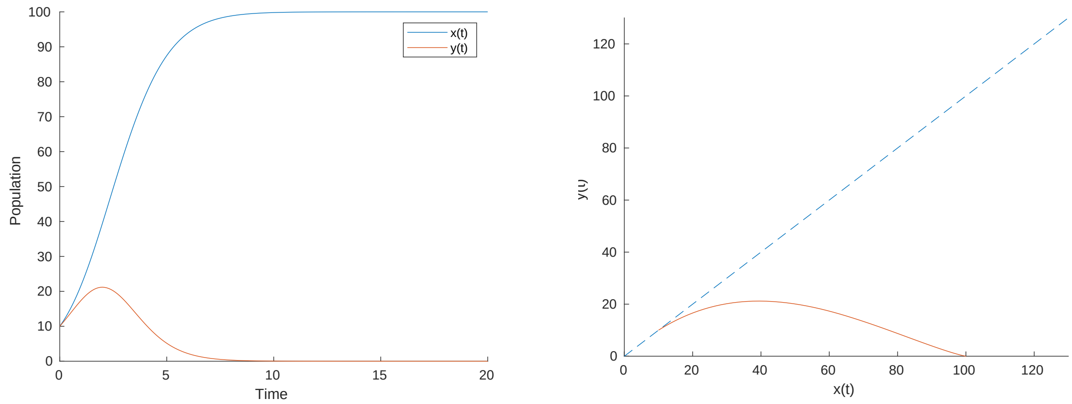
2.3.3 Matlab 程序

```
%% params
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;
x0 = 10;
y0 = 10;

%% solve ode
end_t = 20;
t = 1:0.1:end_t;
[t, xy] = ode45(@pop_ode, t, [x0 y0]);

%% plot
figure(1);
hold on;
plot(t, xy(:, 1), 'DisplayName', 'x(t)');
plot(t, xy(:, 2), 'DisplayName', 'y(t)');
xlabel('Time');
ylabel('Population');
legend('show');

figure(2);
hold on;
range = max(max(xy)) + 30;
fplot(@(x)x, [0 range], '--');
plot(xy(:, 1), xy(:, 2));
xlabel('x(t)');
ylabel('y(t)');
xlim([0 range]);
ylim([0 range]);

%% ode to solve
function dx = pop_ode(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
```

2.3.4 计算结果与分析

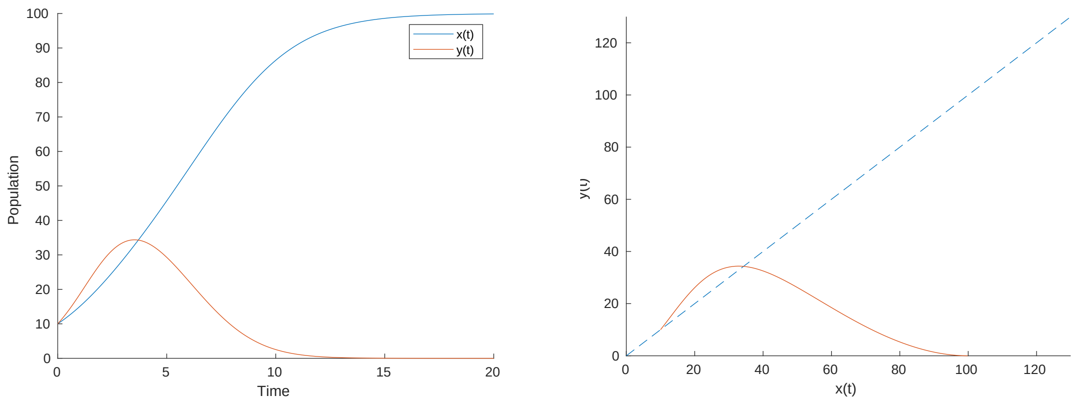
情形 1：$\mathbf{r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2}$，初值 $\mathbf{x_0 = y_0 = 10}$ 该情况除了 s 参数外其他均相同，可以看做“同一起跑线上”的竞争。由时间图可以看出，物种甲数量稳定上升，而物种乙到达一个较小峰值 (约为 20) 后下降，到达稳态时物种甲已经到达最大承载量 100，物种乙数量为 0；由相图可以看出，图线分布在 $y = x$ 下方且距离越来越远，说明物种乙在与甲竞争中一直处于劣势，随着时间增加，二者差距越来越大，最终时间足够长时到达稳态，乙消失，甲占据所有资源。



(a) $x(t)$, $y(t)$ 图线 (b) 相图

图 8: 情形 1 的结果

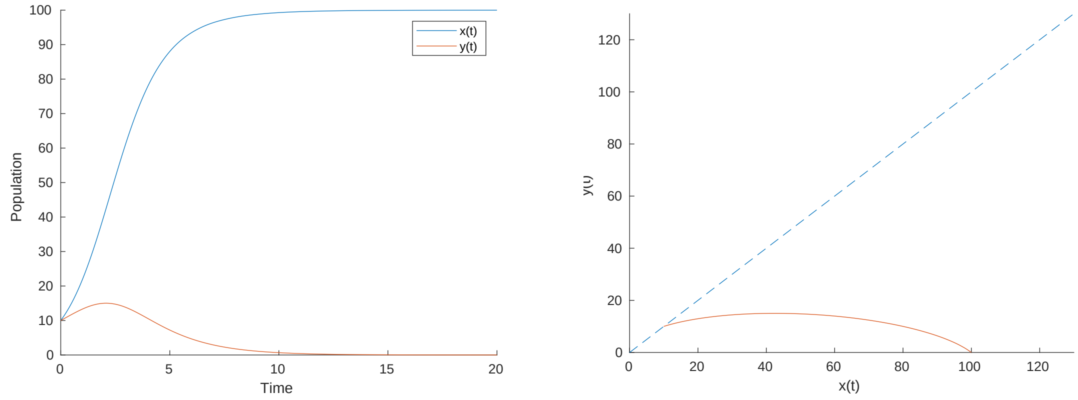
情形 2：将情形 1 改为 $\mathbf{r_1 = 0.5}$，其他参数不变 可以看出，初期由于甲增长率比乙低，使得乙稍微占据优势，但后期甲仍然能竞争过乙，并让乙消失。相比情形 1，竞争的时间也变长了。



(a) $x(t)$, $y(t)$ 图线 (b) 相图

图 9: 情形 2 的结果

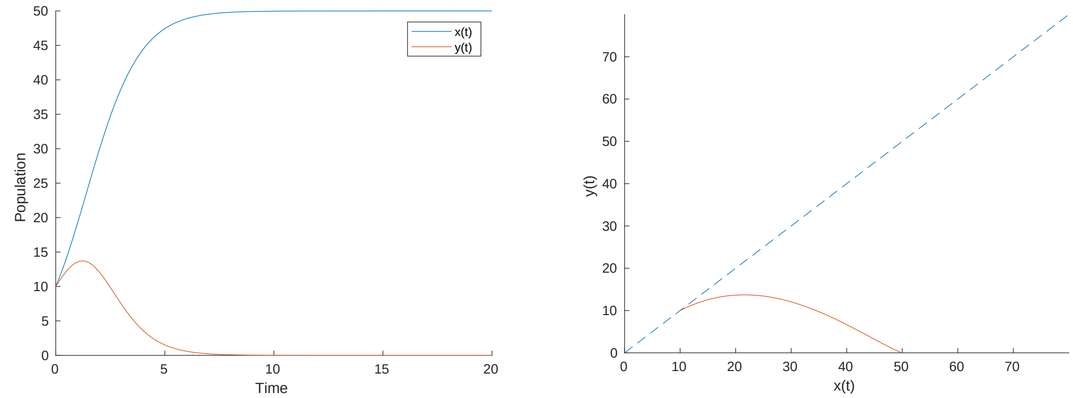
情形 3：将情形 1 改为 $\mathbf{r_2 = 0.5}$，其他参数不变 显然，本身处于劣势的物种乙在 $r_2 = 0.5$ 时处于更大的劣势，使得峰值进一步降低 (低于 15)，且更快被甲淘汰。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 10: 情形 3 的结果

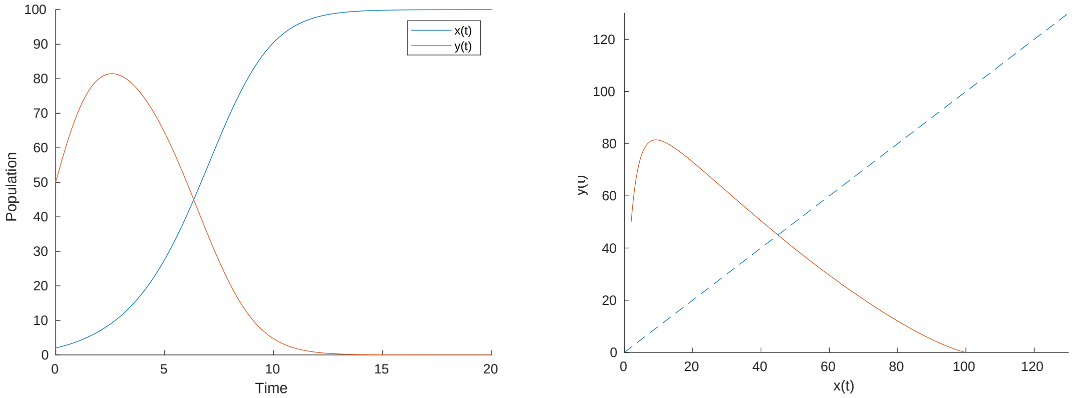
情形 4：将情形 1 改为 $\mathbf{n_1 = 50}$，其他参数不变　减小物种甲的最大承载量，也使得物种乙能够获取的属于物种甲的资源变少，使得峰值比情形 1 低，不过没有改变最终的稳态趋势（物种甲达到最大承载量，乙消失）。同样，如果置 $n_2 = 50$，其他参数不变，对于物种乙会带来更大的劣势，同样无法改变最终的稳态趋势。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 11: 情形 4 的结果

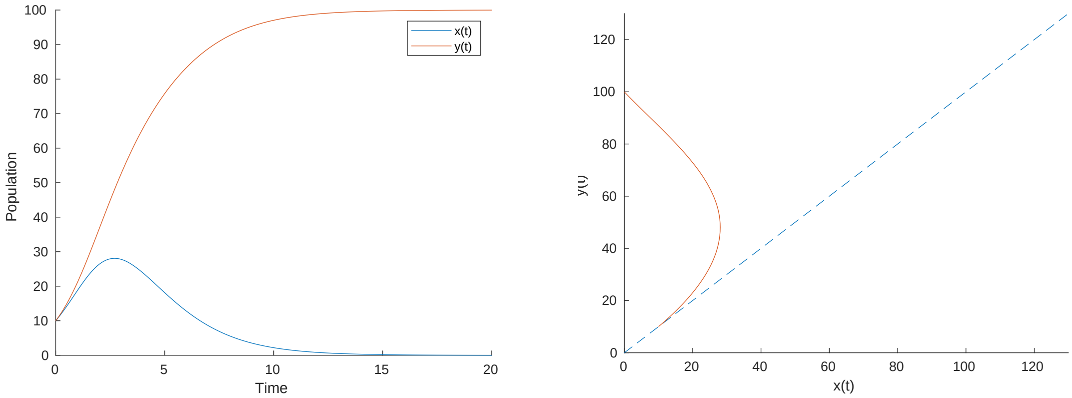
情形 5：将情形 1 改为 $\mathbf{x_0 = 2}$, $\mathbf{y_0 = 50}$，其他参数不变　可以看出，即使物种乙在初期占有优势，但后期竞争过程中仍被甲所取代，稳态解仍不变。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 12: 情形 5 的结果

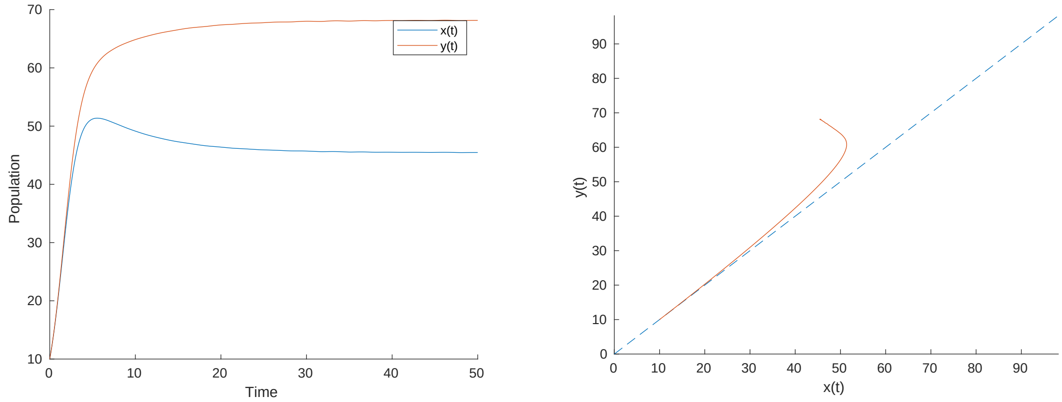
情形 6：将情形 1 改为 $\mathbf{s_1 = 1.5}$, $\mathbf{s_2 = 0.7}$，其他参数不变　此时两个物种数量的发展趋势逆转，稳态时乙占有所有资源，而甲消失。其实，根据方程 (11) 的对称性，只要将情况 1 的甲、乙两物种对调，就可以推出情形 6 下的结果。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 13: 情形 6 的结果

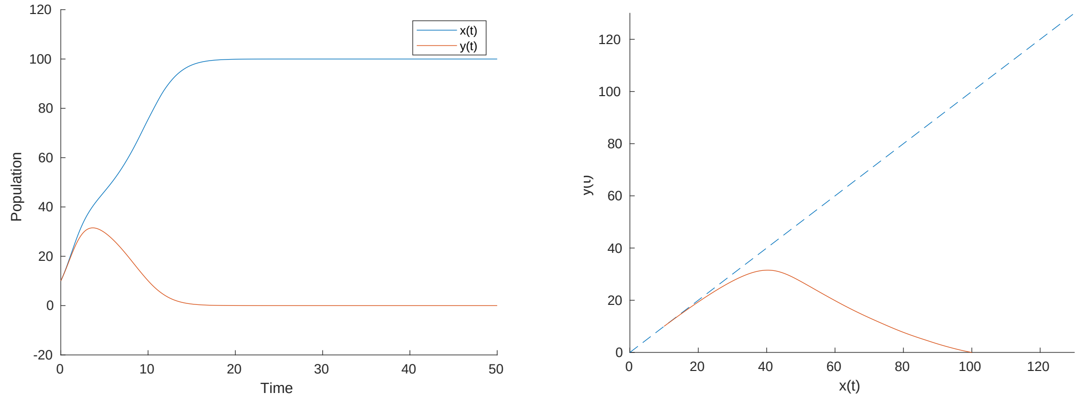
情形 7：将情形 1 改为 $\mathbf{s_1 = 0.8}$, $\mathbf{s_2 = 0.7}$　此时物种乙略占优势，但并没有将物种甲消灭，到达稳态时，甲数量约 45.5，乙数量约 68.2。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 14: 情形 7 的结果

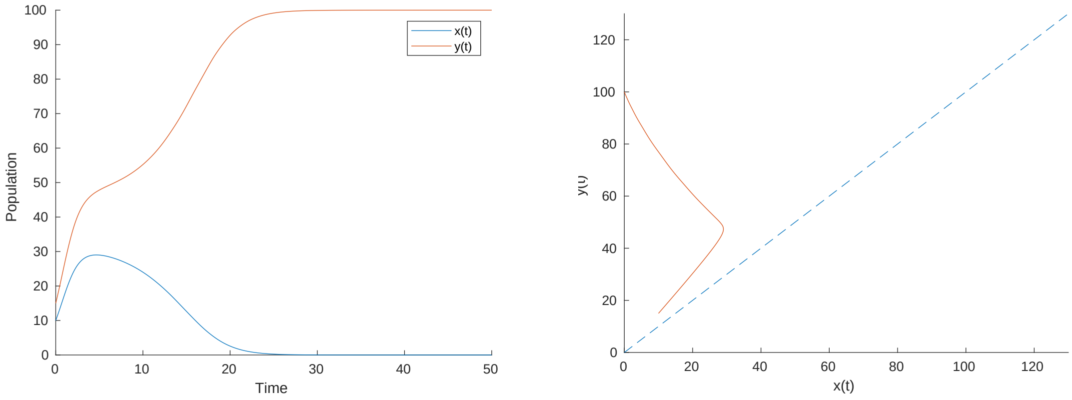
情形 8：将情形 1 改为 $\mathbf{s_1 = 1.5, s_2 = 1.7}$ 此时物种甲占优势，且最终取代了物种乙。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 15: 情形 7 的结果

情形 8：扰动情形 7，使得物种乙的初始数量比甲略多 ($y_0 = 15, x_0 = 10$) 此时物种乙占优势，且最终取代了物种乙。相比情形 7，只是初始值得到了小扰动，但最终结果相差较大。



(a) $x(t)$, $y(t)$ 图线　　(b) 相图

图 16: 情形 7 的结果

2.3.5 结论

由上述计算结果可得出结论，在题给的种群增长模型中，最大承载量、自然增长率、种群初始数量均不是影响稳态解的决定性因素，参数 s 决定了最终的结果，在微分方程中它反映了单位竞争物种对单位本物种的生存资源的抢夺能力。若两种群的 s 参数分别记为 s_1, s_2，在 $s_1 < 1, s_2 < 1$ 时可以达到稳定平衡，此时 s 小的种群占相对优势，稳态时二者可以共存；在 $s_1 < 1, s_2 > 1$ 或 $s_1 > 1, s_2 < 1$ 时可以达到稳定平衡，s 小的种群占绝对优势，它会消灭另一个种群；在 $s_1 > 1, s_2 > 1$ 时平衡不稳定，其他自然条件的改变可能带来最终结果的变化，二者在竞争中均可能获胜，且此时优势种也能消灭劣势种。

3 收获与建议

通过这次实验，我熟练掌握了用 Matlab 对微分方程进行数值求解的方法，并在解决实际问题中加深了对这些方法的理解。建议：希望课本上需要物理知识建模应用题都能使用国际单位制（而不是本实验第一题的英制单位），这样解题计算的时候能够更加方便。