数学实验 · 实验报告2

计算机系　计01　容逸朗　2020010869

1　4-5 放射性废物的处理

1.1　问题分析

首先需要通过物理定律构造圆桶下沉的方程式，然后利用题干给出的数据计算达到临界速度的位置，将这个结果与海的深度作比较即可得到结论。

1.2　模型假设与建立

由题意知，圆桶受到重力、浮力和阻力的影响。以向地心方向为正可列出下式：

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

其中 μ 是阻力系数，f 代表浮力、mg 是桶的重力，$v(t)$ 代表 t 时刻的速度，由于桶是由静止状态开始的，因此当 $t=0$ 时有 $v(0)=0$ 。通过桶的速度可以计算出对应的深度，其方程如下：

$$h(t) = \int_0^t v(t)\,\mathrm{d}t \tag{2}$$

1.3　解析求解

(1) 式是一階线性ODE，需要先去掉 $\frac{dv}{dt}$ 的系数，得到：

$$\frac{\mathrm{d}v}{\mathrm{d}t} = g - \frac{f}{m} - \frac{\mu}{m}v \tag{3}$$

这时可以取积分因子 $I(t) = e^{\int \frac{\mu}{m}\,\mathrm{d}t} = e^{\frac{\mu}{m}t}$ ，那么方程 (3) 的解为：

$$v(t) = \frac{\int (g - \frac{f}{m})e^{\frac{\mu}{m}t}\,\mathrm{d}t + C}{e^{\frac{\mu}{m}t}} = \frac{mg-f}{\mu} + Ce^{-\frac{\mu}{m}t} \tag{4}$$

又因为 $v(0)=0$ ，由此可知 $C = -\frac{mg-f}{\mu}$ ，故 (4) 式可化为：

$$v(t) = \frac{mg-f}{\mu}\left(1 - e^{-\frac{\mu}{m}t}\right) \tag{5}$$

整理得：

$$t = -\frac{m}{\mu}\ln\left(1 - \frac{\mu\cdot v(t)}{mg-f}\right) \tag{6}$$

对应的位移距离为：

$$h(t) = \int_0^t v(t)\,\mathrm{d}t = \frac{mg-f}{\mu}t + \frac{m(mg-f)}{\mu^2}(e^{-\frac{\mu}{m}t} - 1) \tag{7}$$

先将题干给出的数据变为标准单位：

$$mg = 527.436lbf = 527.436 \times 0.4536 \times 9.8N = 2344.6N$$
$$f = 470.327lbf = 470.327 \times 0.4536 \times 9.8N = 2090.7N$$
$$\mu = 0.08lbf \cdot s/ft = 0.08 \times 0.4536 \times 9.8 \div 3.281N \cdot s/m = 0.1083N \cdot s/m$$
$$g = 9.8m/s^2$$
$$m = \frac{mg}{g} = \frac{2344.6}{9.8} = 239.245kg$$
$$v(t) = 40ft/s = 40 \div 3.281 = 12.192m/s$$

再将数据代入 (6) 式得:

$$t = -\frac{2344.6 - 2090.7}{0.1083} \times ln(1 - \frac{0.3556 \times 12.192}{2344.6 - 2090.7}) = 11.588s$$

将此值代入 (7) 式:

$$h(11.588) = \frac{2344.6 - 2090.7}{0.1083} \times 11.588 + \frac{239.245 \times (2344.6 - 2090.7)}{0.1083^2} \times (e^{-\frac{0.1083}{239.245} \times 11.588} - 1) = 70.28m$$

因为 $70.28 * 3.281 = 230.569ft < 300ft$ ，这说明了圆桶尚未到海底时已经超过安全的速度。

1.4 算法设计

利用 Matlab 自带的 `ode45` 方法求解微分方程 (3)，得到时间-速度的对应关系，然后利用梯形公式（Matlab 中的 `cumtrapz` ）计算每个时间取样点的累积深度，画出对应的图象便可得到结论。如果要知道到达临界速度时的具体深度，则可以将上面得到的速度-高度用 `spline` 求出插值。

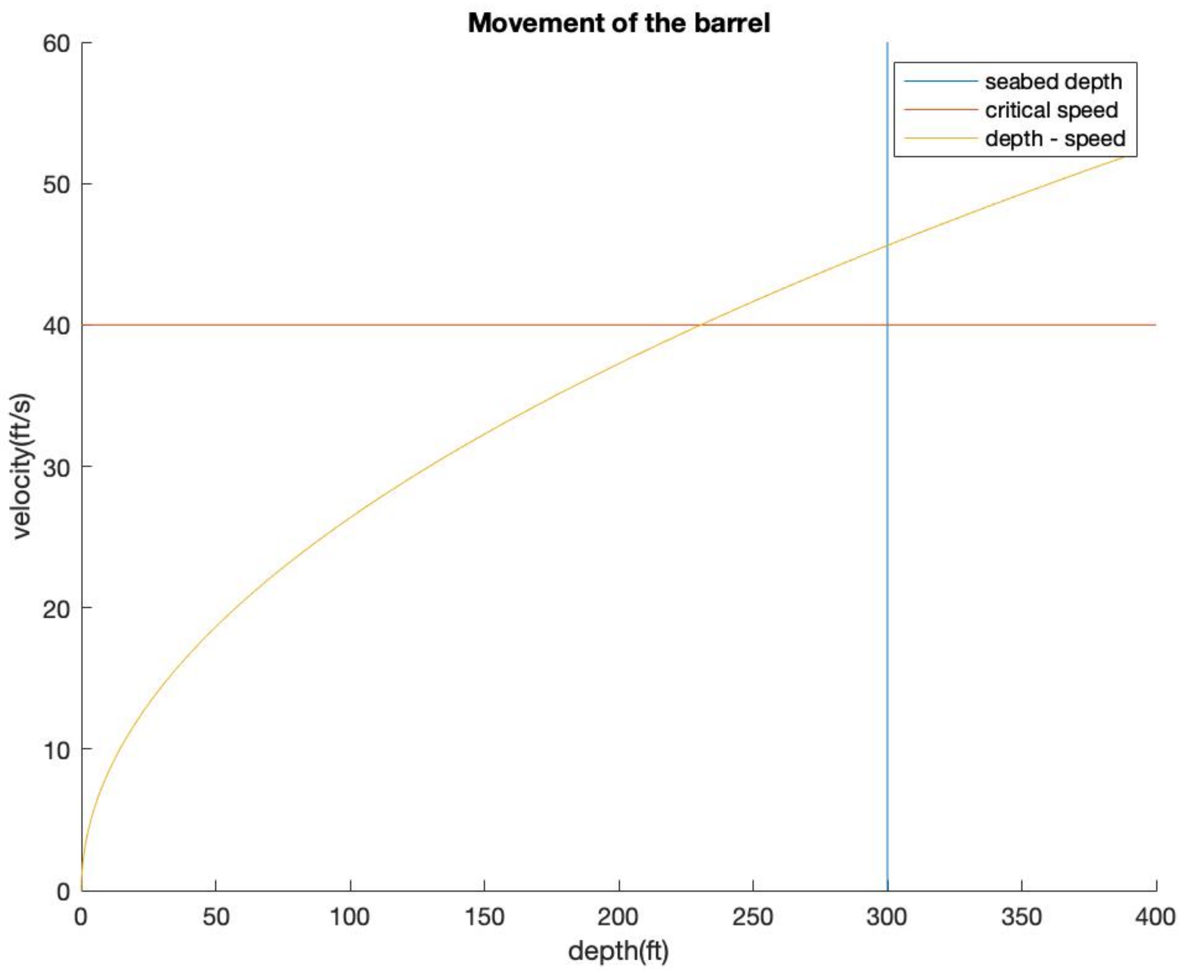
1.5 程序

```
format long;

H = 300; % 海底深度(ft)
V = 40; % 临界速度(ft/s)
m2ft = 3.2808399; % 米和英尺转换因数

% 解微分方程
N = 15;
t = 0 : N / 100: N;
[t, v] = ode45(@findSpeed, t, 0);

% 计算达到特定速度时的深度
h = cumtrapz(t, v);

% 计算达到临界速度时的深度
y = spline(v, h, V / m2ft) * m2ft

% 画图
hold on;
p1 = plot([H, H], [0, 60]); % 画海底深度线
p2 = plot([0, 400], [V, V]); % 画临界速度线
```

```
p3 = plot(h * m2ft, v * m2ft); % 画深度-速度线
xlabel("depth(ft)");
ylabel("velocity(ft/s)");
legend("seabed depth", "critical speed", "depth - speed");
title("Movement of the barrel")
hold off;


function dv = findSpeed(t, v)
mg = 527.436 * 0.4536 * 9.8;
f = 470.32 * 0.4536 * 9.8;
u = 0.08 * 0.4536 * 9.8 / 3.2808399;
g = 9.8;
m = mg / g;
dv = g - f / m - u * v(1) / m;
end
```

1.6 计算结果

圆桶的深度-速度图:



通过插值计算可知圆桶到达临界速度时的深度为：$230.569ft$ 。

1.7 结果数学分析

Matlab 的结果和解析解一致，这说明了数值方法对于解决实际问题有很大的应用空间。

1.8 结果实际意义

$230.569ft < 300ft$ ，圆桶尚未到达海底时已经超过安全速度，故圆桶会破裂。

1.9 结论

圆桶会破裂，工程师赢得了官司。

2 4-6 小船渡河

2.1 问题分析

建立坐标系，通过物理方法构造小船移动的方程式，利用数学方法求解即可得到答案。

2.2 模型假设与建立

假设小船只受河水流速影响，不受其他阻力影响，由题干的描述可以得到如下算式：

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = v_1 - v_2 \cdot \frac{x}{\sqrt{x^2+(d-s)^2}} \\ \frac{\mathrm{d}s}{\mathrm{d}t} = v_2 \cdot \frac{d-s}{\sqrt{x^2+(d-s)^2}} \\ v1 = k \cdot v2 \end{cases} \tag{8}$$

其中 $v1, v2$ 分别是水流速度和船的速度， $s = s(t) \in [0, d]$ 代表小船在 t 时刻相对于出发岸边的距离，$x = x(t)$ 代表小船在 t 时刻相对于 A 点向水流流向移动的距离。

2.3 解析求解

将方程组 (8) 的第一条式子与第二条式子相除，并利用第三条式子消元，可得：

$$\frac{\mathrm{d}x}{\mathrm{d}s} = \frac{k\sqrt{x^2+(d-s)^2} - x}{d-s} \tag{9}$$

令 $z = \frac{x}{d-s}$ ，即 $x = (d-s)z$ 有：

$$\frac{\mathrm{d}x}{\mathrm{d}s} = \frac{\mathrm{d}z}{\mathrm{d}s}(d-s) - z$$

化简得：

$$\frac{\mathrm{d}z}{\mathrm{d}s} = \frac{\frac{\mathrm{d}x}{\mathrm{d}s} + z}{d-s} = \frac{\frac{k\sqrt{x^2+(d-s)^2}-x}{d-s} + \frac{x}{d-s}}{d-s} = \frac{k\sqrt{(\frac{x}{d-s})^2+1}}{d-s} = \frac{k\sqrt{z^2+1}}{d-s}$$

分离变量并且两边同时积分得：

$$\ln\left(z + \sqrt{1+z^2}\right) = -k\ln(d-s) + C \tag{10}$$

代入初值 $x = 0, s = 0, z = \frac{x}{d-s} = 0$ ，得到：

$$0 = \ln(1) = -k\ln(d) + C$$

这时 (10) 式变为:

$$\ln\left(z+\sqrt{1+z^2}\right)=k\ln\left(\frac{d}{d-s}\right)$$

化简得:

$$x=\frac{d-s}{2}\left[\left(\frac{d}{d-s}\right)^k-\left(\frac{d}{d-s}\right)^{-k}\right],s\in[0,d]$$

2.4 算法设计

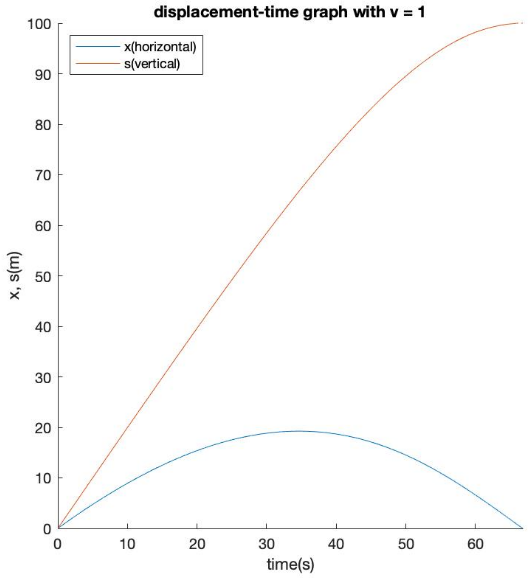
利用 Matlab 的 `ode15s` （龙格-库塔）方法解方程式组 (8)，便可得到结果。

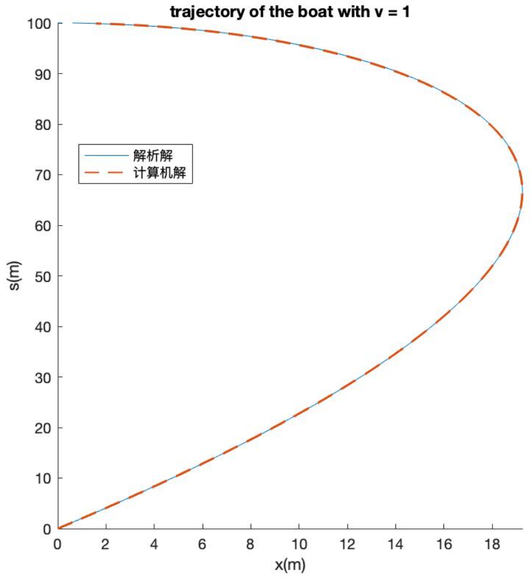
2.5 程序

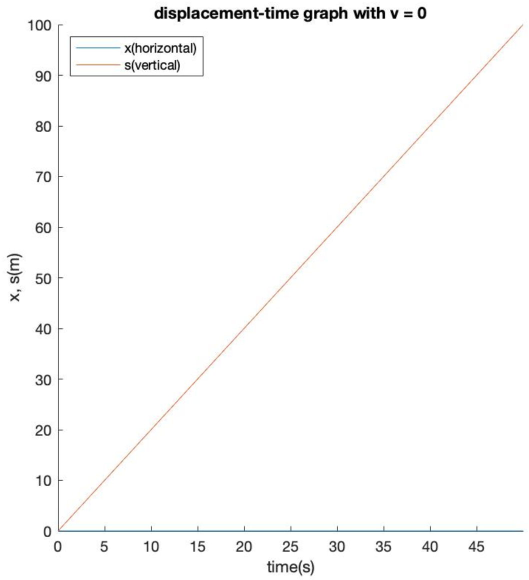
```
format long;
global v0;

cnt = 0;
% 不同的水流速度
v = [1 0 0.5 1.5 2];
% 需时估计
t0 = [80 50 70 120 150];

% 画图
for i = v
    v0 = i;
    cnt = cnt + 1;

    % 解 ode
    t = 0: 1/100: t0(cnt);
    x0 = [0 0];
    [t, x] = ode15s(@boat, t, x0);

    % 计算解析解
    s = 0: 0.1: 100;
    X = myBoat(s);

    % 清除旧图像
    clf;

    % 画时间-位移图
    subplot(1, 2, 1);
    hold on;
    axis([0, inf, 0, 100]);
    plot(t, x(:, 1));
```

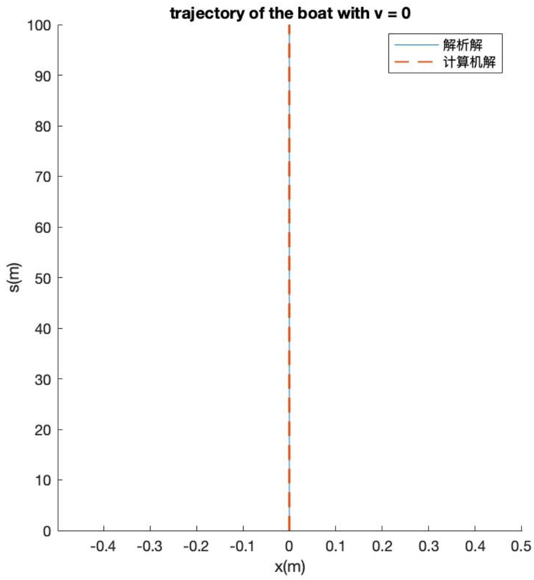
```
    plot(t, x(:, 2));
    xlabel("time(s)");
    ylabel("x, s(m)");
    legend("x(horizontal)","s(vertical)", "Location", "NorthWest");
    title("displacement-time graph with v = " + string(i));
    hold off;

    % 画轨迹图
    subplot(1, 2, 2);
    hold on;
    axis([0, inf, 0, 100]);
    plot(x(:,1), x(:,2));
    plot(X, s, "--", "LineWidth", 1);
    xlabel("x(m)");
    ylabel("s(m)");
    legend("解析解", "计算机解", "Location", "best");
    title("trajectory of the boat with v = " + string(i));
    hold off;

end

function dy = boat(t, x)
global v0;
v1 = v0;
d = 100;
v2 = 2;
a = sqrt(x(1) ^ 2 + (d - x(2)) ^ 2);
dy = [v1 - (v2 * x(1)) / a; v2 * (d - x(2)) / a];
end

function x = myBoat(s)
global v0;
v1 = v0;
d = 100;
v2 = 2;
x = ((d - s) / 2) .* (((d ./ (d - s)) .^ (v1 / v2)) - ...
    (d ./ (d - s)) .^ (-v1 / v2));
end
```

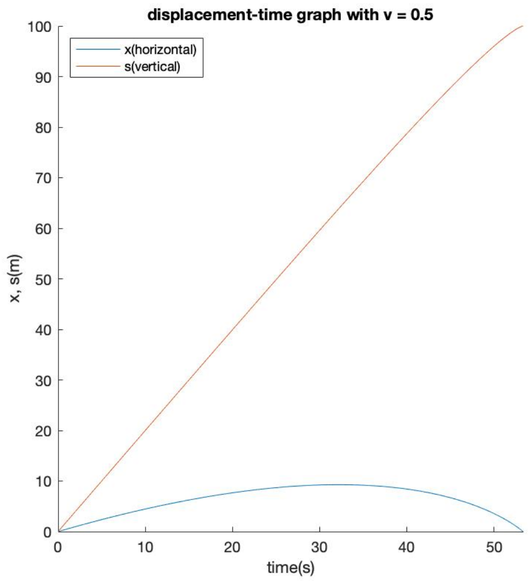
2.6 计算结果

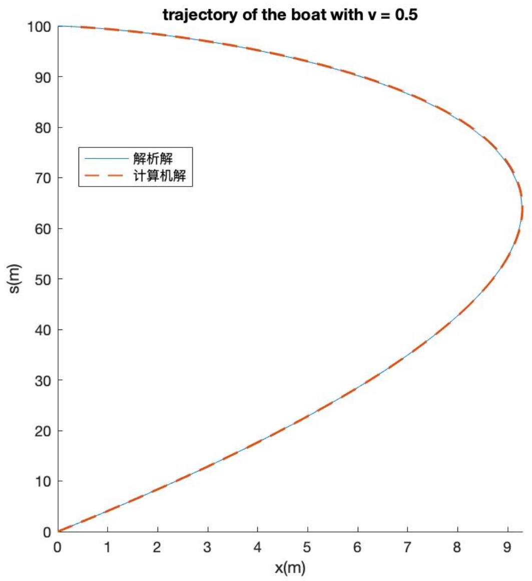

displacement-time graph with v = 1
x(horizontal)
s(vertical)
x, s(m)
time(s)

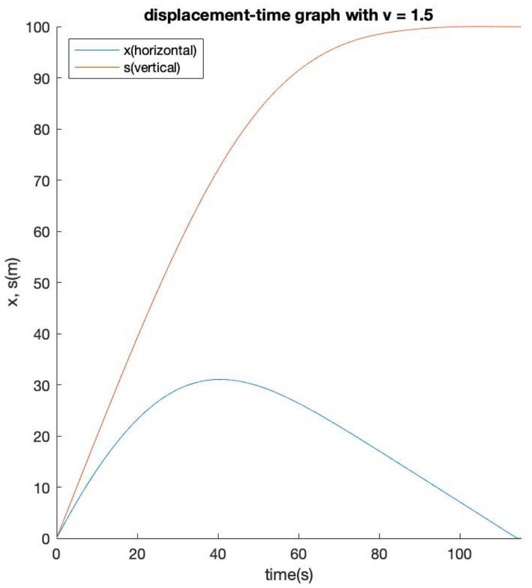

trajectory of the boat with v = 1
解析解
计算机解
s(m)
x(m)

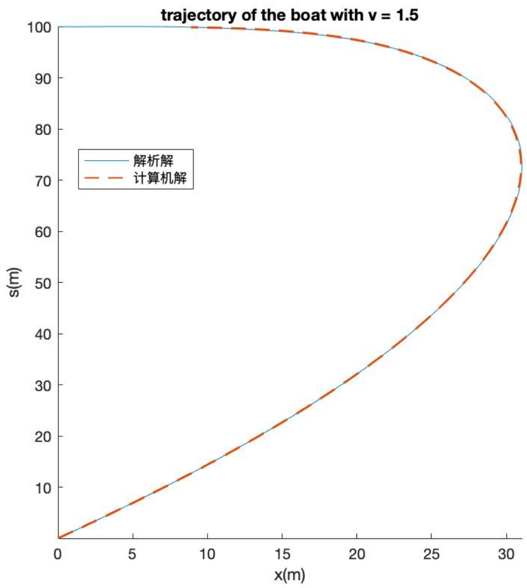

displacement-time graph with v = 0
x(horizontal)
s(vertical)
x, s(m)
time(s)

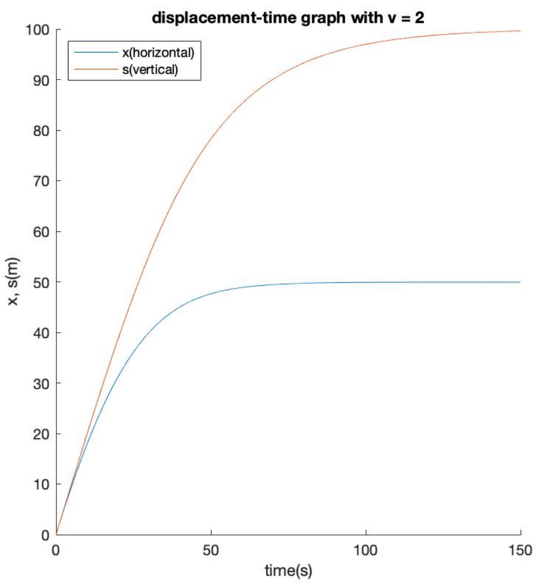

trajectory of the boat with v = 0
解析解
计算机解
s(m)
x(m)

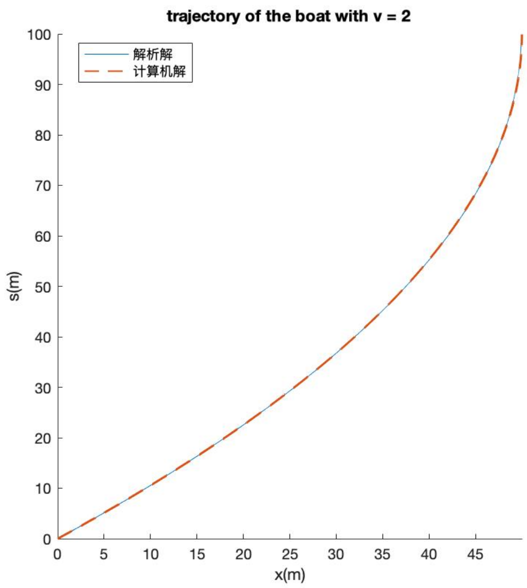

displacement-time graph with v = 0.5
x(horizontal)
s(vertical)
x, s(m)
time(s)


trajectory of the boat with v = 0.5
解析解
计算机解
s(m)
x(m)









2.7 结果数学分析

Matlab 解和解析解相当接近，在图中几乎没有区别。

2.8 结果实际意义

水流速度从零开始，越接近船速则过河时间越大，水平移动方向也越长。当水流速度超过船速时，不存在 A 点到 B 点的方法。

2.9 结论

具体结果见 2.6 节。

3 4-9 种群竞争

3.1 模型假设与建立

题目已给出模型，模型如下：

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

3.2 算法设计

利用 Matlab 自带的龙格-库塔方法（`ode15s`）求解即可。

3.3 程序

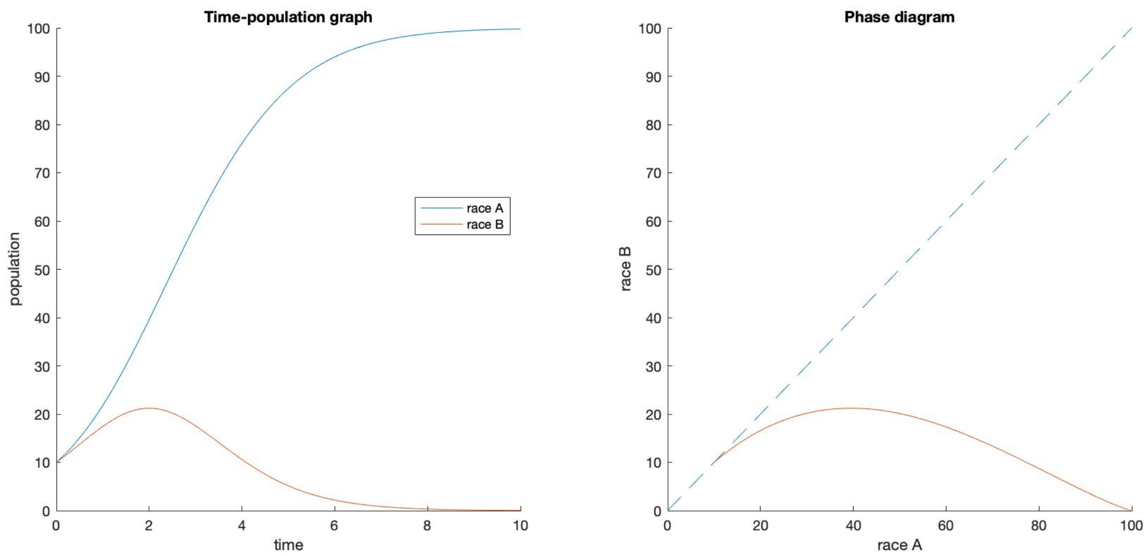
```matlab
format long;

% 时间与初始值
t = 0: 1/100: 10;
x0 = [10 10];

% 解 ode
[t, x] = ode15s(@module, t, x0);

% 时间-种群数量图
subplot(1, 2, 1);
hold on;
plot(t, x(:, 1));
plot(t, x(:, 2));
xlabel("time");
ylabel("population");
legend("race A", "race B", "Location", "best");
title("Time-population graph");
hold off;

% 相图
subplot(1, 2, 2);
hold on;
tx = 0: 100;
plot(tx, tx, "--");
plot(x(:, 1), x(:, 2));
xlabel("race A");
ylabel("race B");
title("Phase diagram");
hold off;

function dy = module(t, x)
```

```
r1 = 1;
r2 = 1;
n1 = 100;
n2 = 100;
s1 = 0.5;
s2 =2;
dy = [r1 * x(1) * (1 - x(1) / n1 - s1 * x(2) / n2);
     r2 * x(2) * (1 - s2 * x(1) / n1 - x(2) / n2)];
end
```

3.4 结果及分析

所有数据未修改前的数值如下：$r_1 = r_2 = 1, n_1 = n_2 = 100, s_1 = 0.5, s_2 = 2, x_0 = y_0 = 10$ ，每次讨论只修改对应的值，其余值不变。

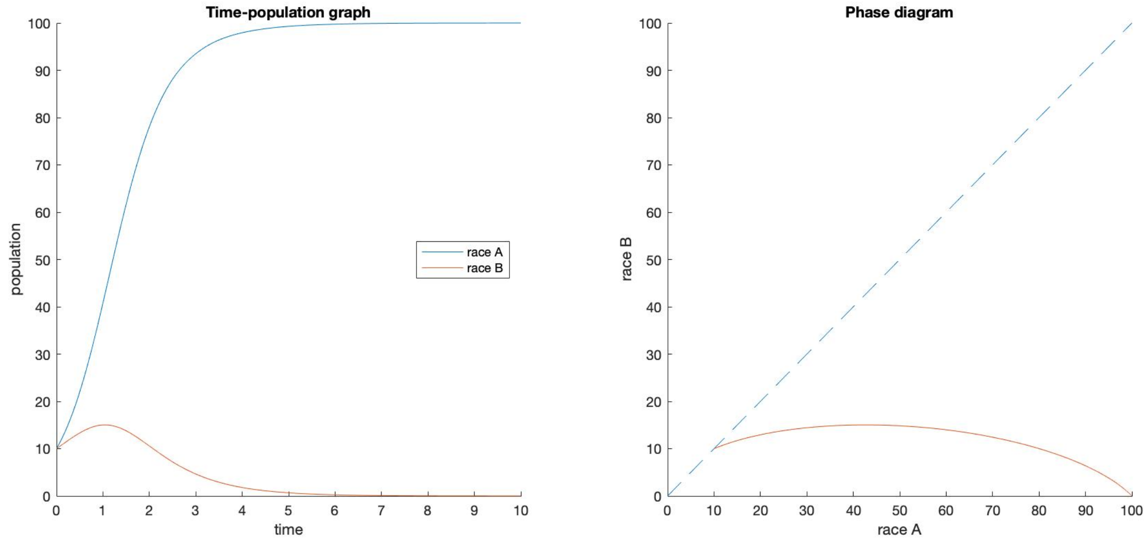
3.4.1 初始情况



从左图中可见，种群甲的数量不断上升（直至上限 100），而乙的数量增长到 20 后便不断往下。从右图则可以看到红线从未超过 $y = x$ ，也就是说明种群甲的优势一直比乙大，这种差距随着时间增大而不断增大，直至最后乙消失，甲到达上限。

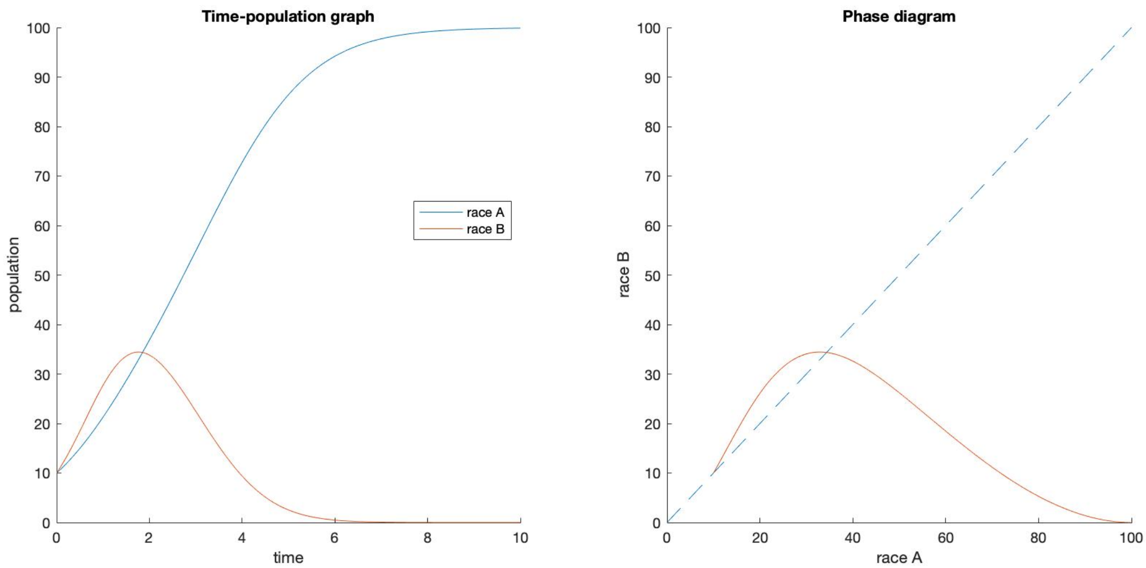
3.4.2 改变 $r_1, r_2, n_1, n_2, x_0, y_0$ ，而 s_1, s_2 不变

3.4.2.1 改变增长率 $r_1 = 2$



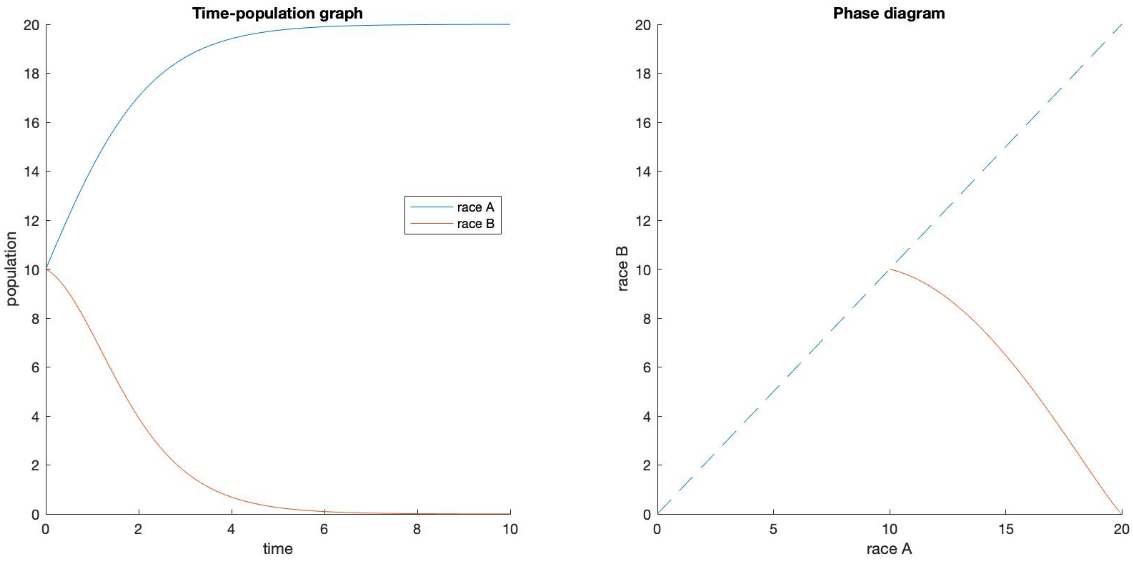
从左图中可见，种群甲的数量不断上升（直至上限 100），而乙的数量增长到 15 后便不断往下。从右图则可以看到红线从未超过 $y = x$ ，也就是说明种群甲的优势一直比乙大，与 3.4.1 相比，甲由于增长率更大，因此获得的优势更为明显，乙灭绝的时间更早。

3.4.2.2 改变增长率 $r_2 = 2$



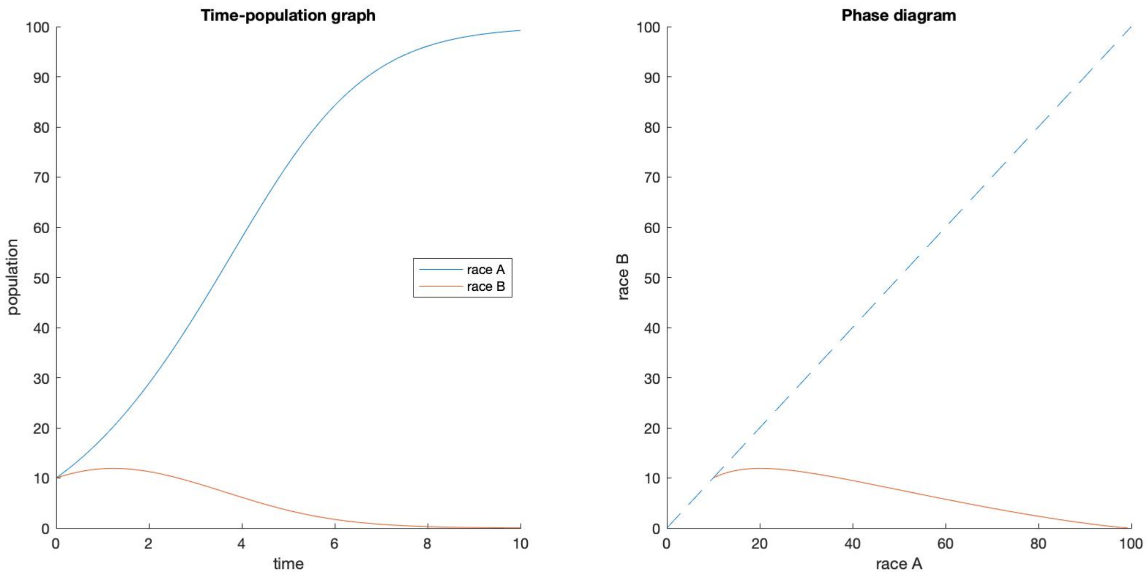
从左图中可见，种群甲的数量不断上升（直至上限 100），而乙的数量增长到 35 后便不断往下，在前期的时候乙的数量较甲为多，但一段时间后甲的数量便反超了乙。从右图则可以看到红线在部分时间在 $y = x$ 的上方，也就是说乙曾经取得优势，但由于乙消耗的资源更多，因此在时间变长后甲仍然取得优势，最终乙灭绝。

3.4.2.3 改变最大容量 $n_1 = 20$



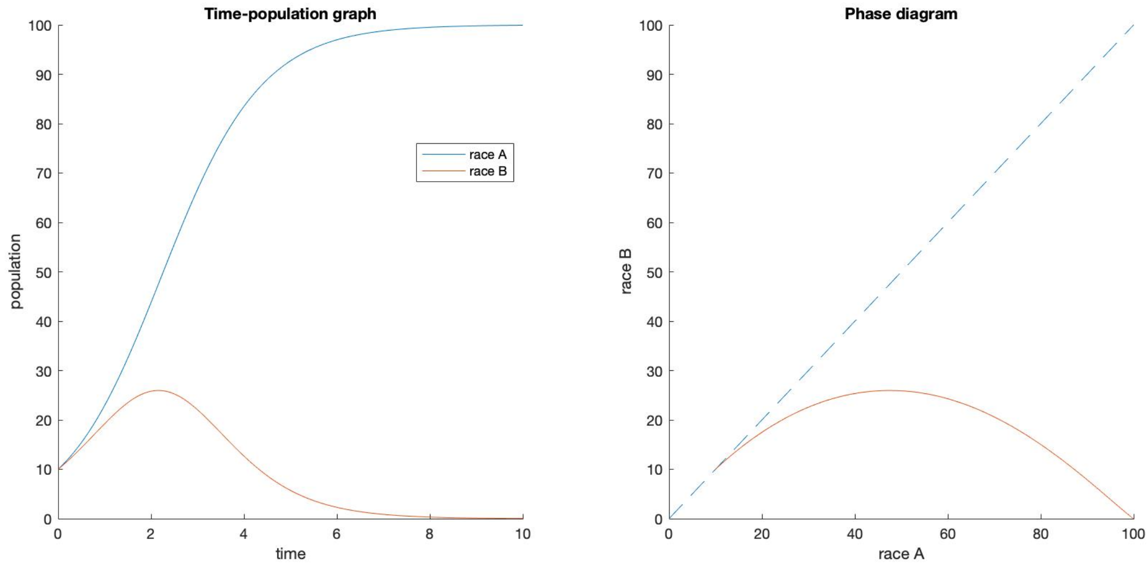
即使将甲的最大容量减少，乙仍然不能与甲竞争，因此最终乙灭绝。

3.4.2.4 改变最大容量 $n_2 = 20$



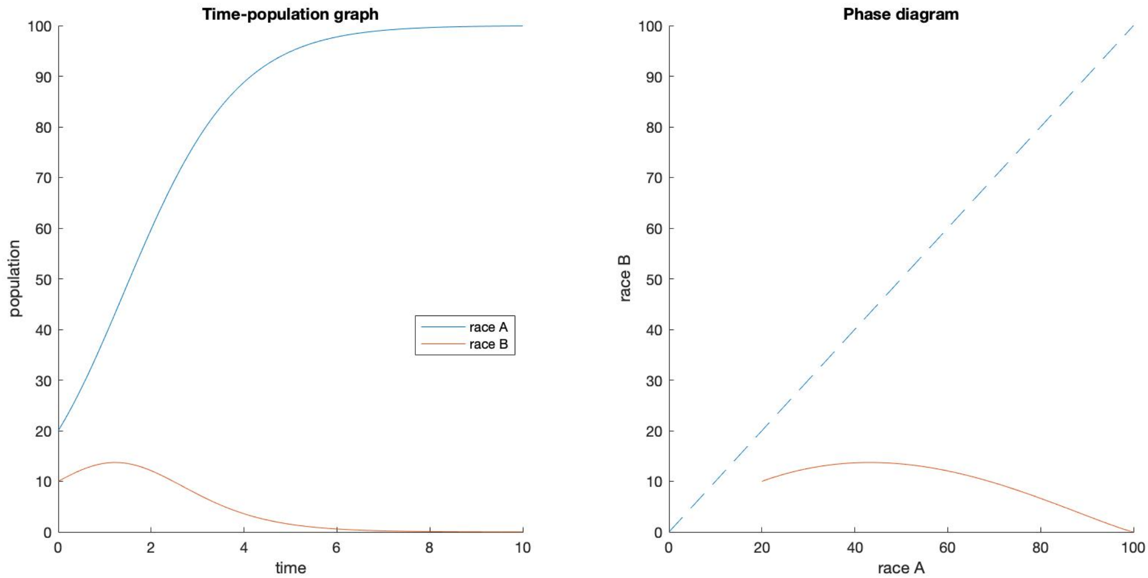
降低最大容量后，乙的竞争力与 `3.4.1` 相比进一步下降，最终乙灭绝。

3.4.2.5 改变最大容量 $n_2 = 500$



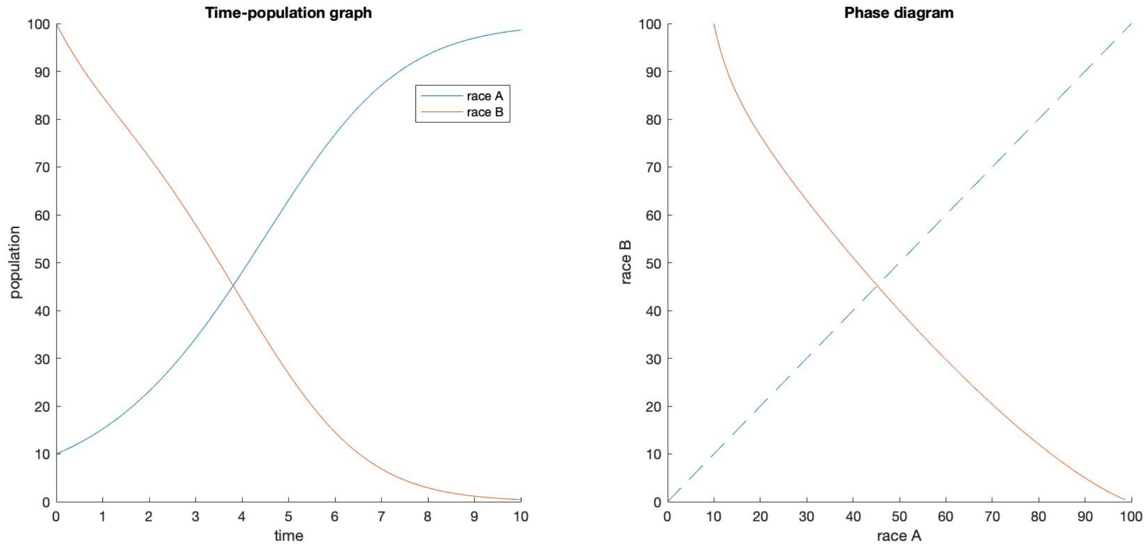
增加最大容量后，乙的竞争力与 3.4.1 相比有所提升，但增长到一定程度后乙的数量会不断减少，最终仍然是乙灭绝。

3.4.2.6 改变初值 $x_0 = 20$



增加甲的初始数量后，乙更难与甲竞争，因此灭绝的时间更早。

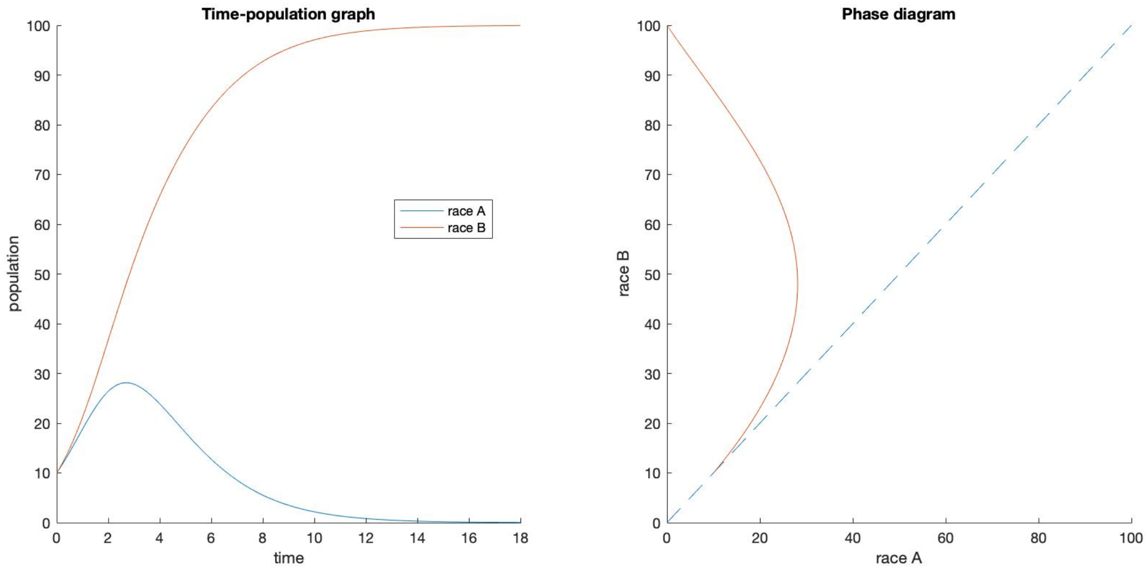
3.4.2.7 改变初值 $y_0 = 100$



增加乙的初始数量后，虽然乙在前期的数量较多，但总数却因为消耗的资源较多而不断减少，直至最终灭绝。

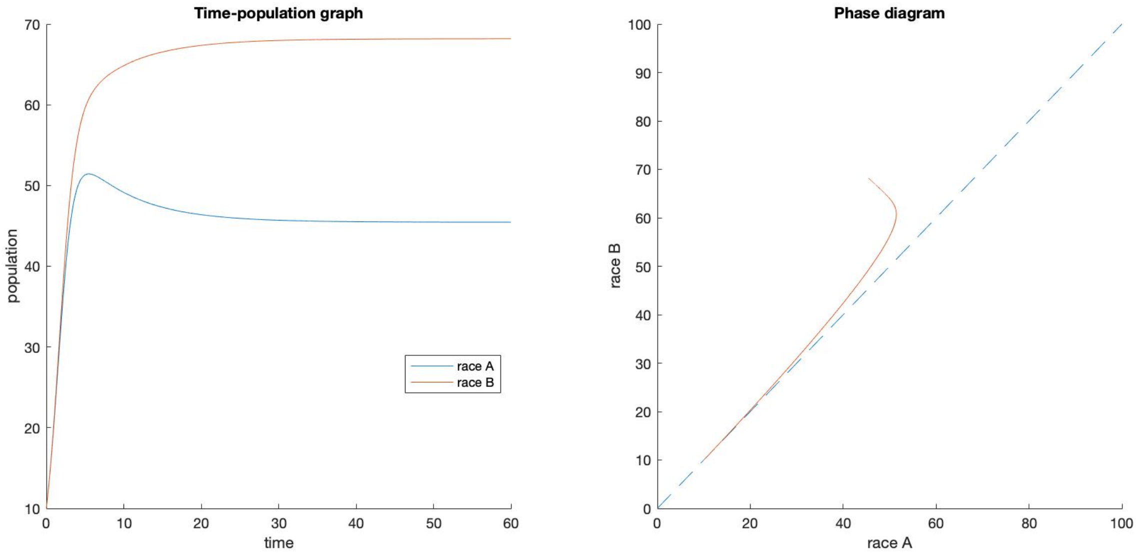
3.4.3 改变消耗率 s_1, s_2

3.4.3.1 $s_1 = 1.5 > 1, s_2 = 0.7 < 1$



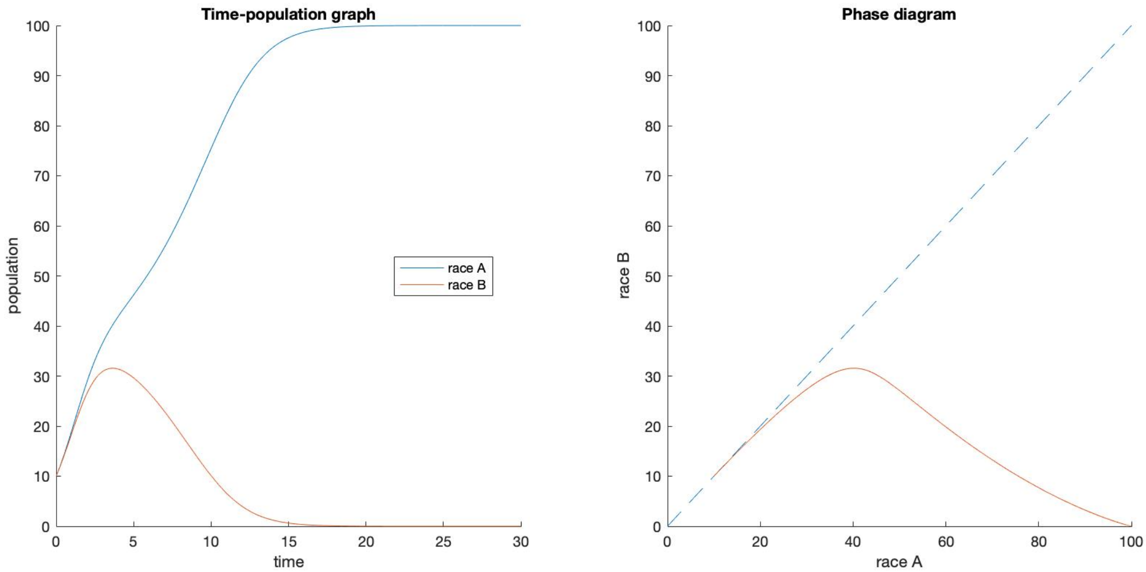
情况与 3.4.1 相反，乙由于消耗率较多而取得优势，两个物种的情况完全相反，最终甲灭绝。

3.4.3.2 $s_1 = 0.8 < 1, s_2 = 0.7 < 1$



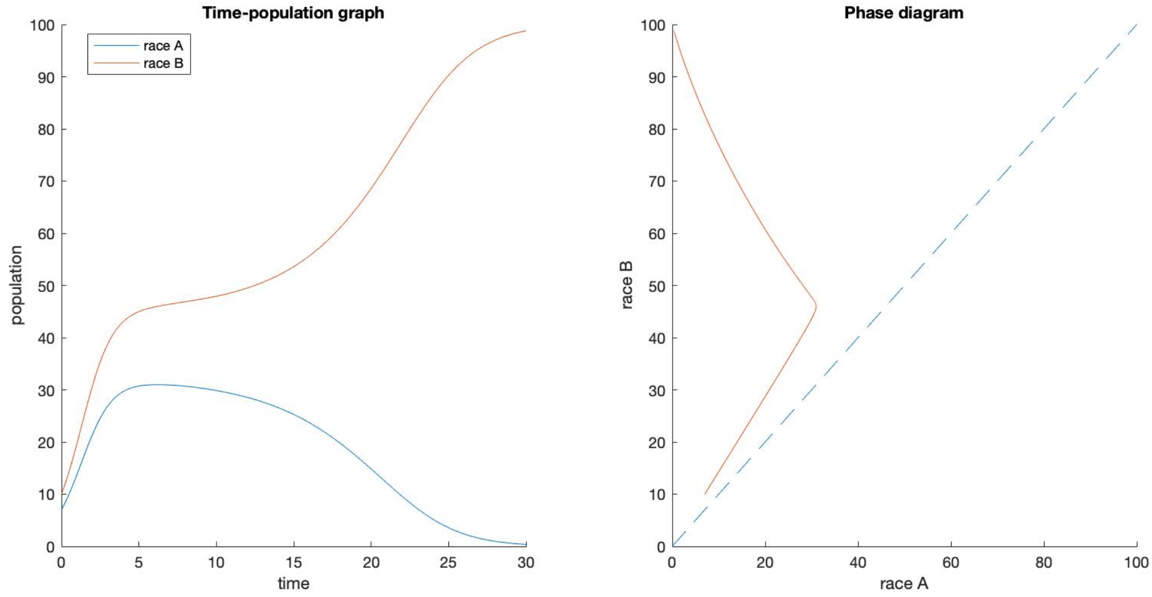
乙由于消耗率较低，因此在竞争中取得了优势，但因为两者消耗的资源都小于1，在资源稳定的情况下两个物种可以相互共存，最终甲约有 45 只，而乙有 68 只。

3.4.3.3 $s_1 = 1.5 > 1, s_2 = 1.7 > 1$



在双方都大量消耗资源的情况下，由于初始值相同而乙的消耗率较高，故甲对乙的优势不断变大（如右图），最终乙灭绝。

3.4.3.4 $s_1 = 1.5 > 1, s_2 = 1.7 > 1, x_0 = 6$



在双方都大量消耗资源的情况下，虽然乙的消耗率较高，但甲的初始数量比乙少时乙仍然取得了绝对的优势，最终消灭了物种甲。

3.5 结果实际意义与结论

从各种情况下可以推测 s_1, s_2 代表了对方对己方的压力，当这种压力小于 1 时两者可以共存。当其中一方压力小于 1 而另外一方大于 1 时，压力较低者会取代另一种群。倘若两者的压力都大于 1 ，则需要综合考虑其他条件，这时双方都有机会取代另一物种。