

云南大学数学与统计学院实验教学中心 实验报告

课程名称: 数学建模实验	学期: 2016~2017 学年下学期	
指导教师: 李朝迁		
学生: 刘鹏 20151910042 信计	学生: 王泽坤 20151910011 应数	学生: 段奕臣 20151910002 应数
实验名称: MATLAB 微分方程求解		成绩:
实验编号: 三	实验日期: 2017 年 4 月 21 日	实验学时: 2
学院: 数学与统计学院	专业: 信息与计算科学	年级: 2015 级

一、实验目的

二、实验内容

【实验内容和步骤】

1 题

求微分方程解析解的 MATLAB 命令为 `dsolve`, 请使用 `help` 命令学习。

Solution:

dsolve

Differential equations and systems solver

Syntax

```
S = dsolve(eqn)
S = dsolve(eqn,cond)
S = dsolve(eqn,cond,Name,Value)
[y1,...yN] = dsolve(__)
```

Description

`S = dsolve(eqn)` solves the differential equation `eqn`, where `eqn` is a symbolic equation. Use `diff` and `==` to represent differential equations. For example, `diff(y,x) == y` represents the equation $dy/dx = y$. Solve a system of differential equations by specifying `eqn` as a vector of those equations.

* `S = dsolve(eqn)` 能够求解符号形式的微分方程 `eqn`。通过 `diff` 命令与 `==` 可以构建一个微分方程。例如, `diff(y,x) == y` 就代表 $dy/dx = y$ 。通过将 `eqn` 指定为符号方程的向量形式, 可以解决微分方程组。

`S = dsolve(eqn,cond)` solves `eqn` with the initial or boundary condition `cond`.

* `S = dsolve(eqn,cond)` 可以解决带有初值条件或者边界条件的方程。

`S = dsolve(eqn,cond,Name,Value)` uses additional options specified by one or more `Name, Value` pair arguments.

* `S = dsolve(eqn,cond,Name,Value)` 使用额外的一个或多个 `Name, Value` 双参数选项。

`[y1,...,yN] = dsolve(__)` assigns the solutions to the variables `y1,...,yN`.

* `[y1,...,yN] = dsolve(__)` 将解赋值到变量 `y1,...,yN`。

2 题

Page85 2 题: 用欧拉方法和龙格-库塔方法求下列微分方程初值的数值解, 画出解的图形, 对结果进行分析比较。

$$(1) \begin{cases} y' = y + 2x, \\ y(0) = 1 \end{cases}, \quad (0 \leq x \leq 1), \quad \text{精确解 } y = 3e^x - 2x - 2;$$

$$(2) \begin{cases} y' = x^2 - y^2, \\ y(0) = 0 \text{ 或 } y(0) = 1 \end{cases}$$

$$(3) \begin{cases} x^2 y'' + xy' + (x^2 - n^2)y = 0, \\ y\left(\frac{\pi}{2}\right) = 2, y'\left(\frac{\pi}{2}\right) = -\frac{2}{\pi} \end{cases} \quad (\text{贝塞尔方程, 令 } n=0.5), \text{ 精确解 } y = \sin x \sqrt{\frac{2\pi}{x}}.$$

注释:

- 用欧拉方法进行编程;
- 对于龙格-库塔方法进行直接调用, ode23 或者 ode45。

Solution:

算法总分析:

这些微分方程都有初值, 而所谓的初值, 指的就是所求函数在定义区间起始点的真实取值或者观测值。

数值微分的核心思想就是将连续的区间片段化, 然后根据初值, 利用导数, 算出第二个区间的起始值或者称为第一个区间的结束值。然后将这个值, 作为第二个区间的初值, 一路迭代下去, 最终的到这些区间的端点值, 这就是一个离散的但是很密集 (因为区间取得很小) 的数值解。可以应用在工程中。

欧拉方法的基本思想就是对于连续函数, 在一个很小的区间 $[x_n, x_{n+1}]$ 上用数值微分方法的前差公式 (向右), 代替方程 $y'(x) = f(x, y)$ 中的左端的导数 y' , 而右端函数 $f(x, y(x))$ 中的 x 取 $[x_n, x_{n+1}]$ 中的某一点, 于是得到

$$y(x_{n+1}) = y(x_n) + hf(x, y(x)), \quad x \in [x_n, x_{n+1}].$$

这个公式毫无疑问是精确的。因为, 这符合差分的定义, 而且根据拉格朗日中值理, 肯定存在一个位于 $[x_n, x_{n+1}]$ 中的 x 使得上面的公式成立。但是这样还是写不出程序, 因为我们无法解决存在性数值确定解的选取问题, 所以必须给定一个能拿得出手的 x 。

由于区间很小, 所以连续函数在这个区间上的变化范围定然不会太剧烈, 即接近一个直线段, 故不妨取左端点数值 x_n , 这样就可以给出一个近似:

$$y_{n+1} = y_n + hf(x_n, y(x_n)).$$

当然也可以给出一个靠右的近似。 $y_{n+1} = y_n + hf(x_{n+1}, y(x_{n+1}))$ 。这就是所谓的向前与向后欧拉公式。因为向后欧拉公式涉及到对于自身的调用, 而且其精度没有理由会更好, 所以我们将向前欧拉公式写成程序。

(1)

程序代码:

```
1 % filename: Euler Forward 1
2 step = 0.1;
3 %% The real graph
4 x = [0 : step : 1];
5 y = 3 * exp(x) - 2 * x - 2;
6 plot(x, y, 'ro');
7 hold on;
8
9 %% Euler Method
10 y_cal = [1];
11 y_0 = 1;
12 for i = 0.1 : step : 1
```

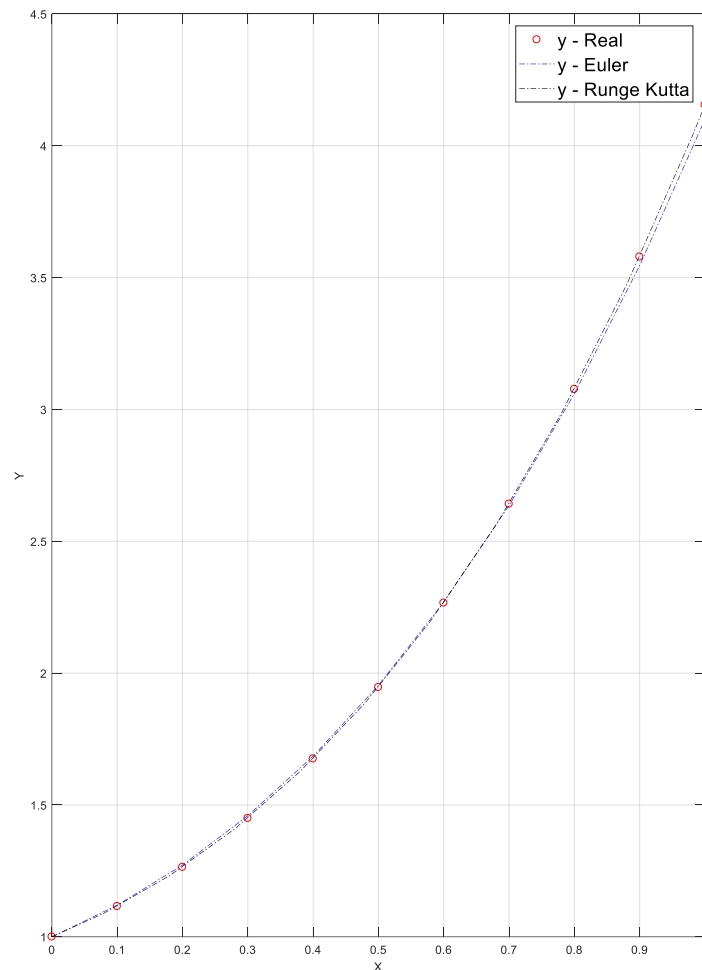
```

13     tmp = y0 + (y0 + 2 * i) * step;
14     ycal = [y cal, tmp];
15     y0 = tmp;
16 end
17 plot(x, y cal, 'b-.');
18
19 %% Runge-Kutta methods
20 tspan = [0 1];
21 y0 = 1;
22 [t, y] = ode23(@(t, y) 2 * t + y, tspan, y0);
23 plot(t, y, '-.k');
24 grid on
25 xlabel('X');
26 ylabel('Y');
27 legend('\fontsize{16}y - Real', '\fontsize{16}y - Euler', ...
28        '\fontsize{16}y - Runge Kutta');

```

程序代码 1

运行结果:



运行结果 1

(2)

程序代码:

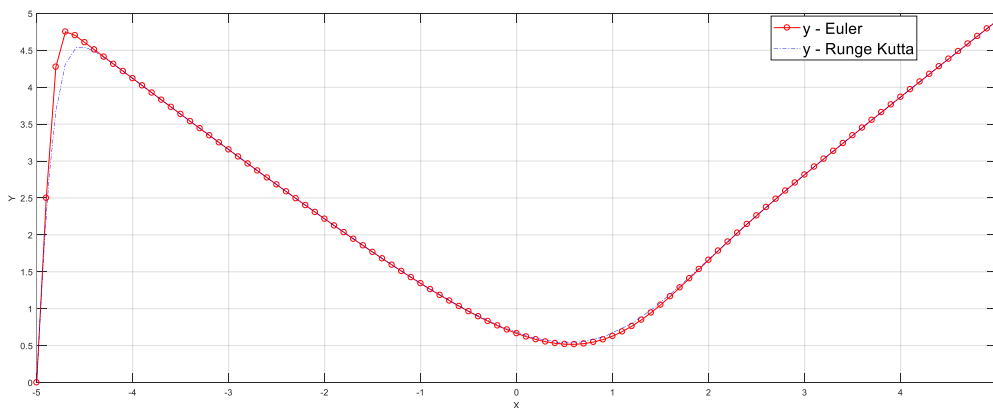
```

1  % filename: Euler Forward 2
2  %% 欧拉向前方法
3  step = 0.1;
4  y 0 = 0;
5  y 1 = [];
6  x = -5 : step : 5;
7  for i = -5 : step : 5
8      y 1 = [y 1, y 0];
9      y 0 = y 0 + (i^2 - y 0^2) * step;
10 end
11 plot(x, y 1, 'r-o')
12 hold on;
13
14 %% Runge-Kutta methods
15 tspan = [-5 5];
16 y0 = 0;
17 [t, y] = ode23(@(t, y) t^2 - y^2, tspan, y0);
18 plot(t, y, 'b-.');
19 tspan = [-5 5];
20 grid on
21 xlabel('X');
22 ylabel('Y');
23 legend('\fontsize{18}y - Euler', '\fontsize{18}y - Runge Kutta');

```

程序代码 2

运行结果:



运行结果 2

(3)

二阶常微分方程的向前欧拉方法，需要给出两个初值，从而进行迭代。而对应的龙格·库塔方法，就需要一个常微分方程组，利用二维向量进行函数的定义。

程序代码:

```

1  % filename: Euler Forward 3
2  step = 0.1;
3  %% The real graph
4  x = [pi / 2 : step : 5 * pi];
5  y = sin(x) .* sqrt(2 * pi ./ x);
6  plot(x, y, 'ro');
7  hold on;
8
9  %% Euler method

```

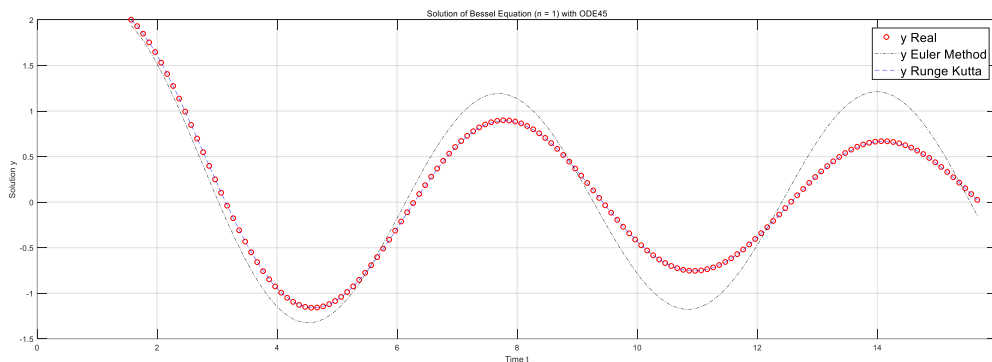
```

10 y 0 = 2;
11 dy 0 = - 2 / pi;
12 y cal = [];
13 n = 0.5;
14 for i = pi / 2 : step : 5 * pi
15     ddy 0 = ( - i * dy 0 - (i^2 - n^2) * y 0 ) / i^2;
16     % get the value of the second derivative test
17     y 0 = y 0 + step * dy 0;
18     dy 0 = dy 0 + ddy 0 * step;
19     % get the value of the first derivative test by the second
20     y cal = [y cal,y 0];
21 end
22 plot(x,y cal,'k-.');
23 grid on;
24 %% Runge-Kutta method
25 tspan = [pi/2, 5 * pi];
26 [t,y] = ode45(@vdp1,tspan,[2;-2 / pi]);
27 plot(t,y(:,1),'b--');
28 title('Solution of Bessel Equation (n = 1) with ODE45');
29 xlabel('Time t');
30 ylabel('Solution y');
31 legend('\fontsize{16}y Real','\fontsize{16}y Euler Method',...
32         '\fontsize{16}y Runge Kutta');
33 hold off;

```

程序代码 3

运行结果:



运行结果 3

之前如果做一个取值区间比较小的测试，就会发现向前欧拉公式得出的结果与实际结果几乎没有差异，然而，当我们把区间长度取到比较大的时候，就会发现，实际的结果与数值结果相差越来越大。这就是误差累积的结果。

3 题

Page85 ~87 3 题~7 题，9 题，任选一题。

两种群相互竞争模型如下：

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

其中 $x(t)$, $y(t)$ 分别为甲乙两种群的数量; r_1 , r_2 为他们的固有增长率; n_1 , n_2 为它们的最大容量。 s_1 的含义是, 对于供养甲的资源而言, 单位数量乙 (相对 n_2) 的消耗为单位数量甲 (相对 n_1) 消耗的 s_1 倍, 对 s_2 可作相应的解释。

该模型无解析解，试用数值解法研究以下问题：

- (1) 设 $r_1 = r_2 = 1$, $n_1 = n_2 = 100$, $s_1 = 0.5$, $s_2 = 2$, 初值 $x_0 = y_0 = 10$, 计算 $x(t)$, $y(t)$, 画出它们的图形及相图 (x, y) , 说明相同时间 t 充分大以后 $x(t)$, $y(t)$ 的变化趋势（人们今天看到的已经是自然界长期演变的结果）。
- (2) 改变 r_1 , r_2 , n_1 , n_2 , x_0 , y_0 , 但 s_1 , s_2 不变（或保持 $s_1 < 1$, $s_2 > 1$ ），计算并分析所得结果；若 $s_1 = 1.5 (> 1)$, $s_2 = 0.7 (< 1)$, 再分析结果。由此你能得到什么结论，请用各参数生态学上的含义做出解释。
- (3) 试验当 $s_1 = 0.8 (< 1)$, $s_2 = 0.7 (< 1)$ 时会有什么结果；当 $s_1 = 1.5 (> 1)$, $s_2 = 1.7 (> 1)$ 时又会有什么结果。能解释这些结果吗？

Solution:

(1)

程序代码：

```
1 % filename: odefun
2 % definition of the differential equation group
3 function dydt = odefun(t,y,r1,r2,n1,n2,s1,s2)
4     dydt = zeros(2,1);
5     dydt(1) = r1 * y(1) * (1 - y(1) / n1 - s1 * y(2) / n2);
6     dydt(2) = r2 * y(2) * (1 - s2 * y(1) / n1 - y(2) / n2);
7 end
```

程序代码 4

(2)

程序代码：

```
1 % filename: predator prey
2 % Testing the influence of the coefficients
3 %% NO.1 default
4 tspan = [0 20];
5 opt=odeset('reltol',1e-6,'abstol',1e-9);
6 [t,y] = ode45(@odefun,tspan,[10,10],opt,1,1,100,100,0.5,2);
7 figure;
8 subplot(1,2,1);
9 plot(t,y(:,1),'b--',t,y(:,2),'r--');
10 grid on
11 xlabel('\fontsize{14}Time');
12 ylabel('\fontsize{14}Numbers');
13 legend('\fontsize{16}species 1','\fontsize{16}species 2');
14 title('\fontsize{16}NO.1 default')
15 subplot(1,2,2);
16 plot(y(:,1),y(:,2));
17 grid on;
18 xlabel('\fontsize{16}species 1');
19 ylabel('\fontsize{16}species 2');
20 legend('\fontsize{16}relative');
21
22 %% NO.2 [r1, r2] --> [0.3, 0.3]
23 tspan = [0 20];
24 opt=odeset('reltol',1e-6,'abstol',1e-9);
25 [t,y] = ode45(@odefun,tspan,[10,10],opt,0.3,0.3,100,100,0.5,2);
26 figure;
27 subplot(1,2,1);
28 plot(t,y(:,1),'b--',t,y(:,2),'r--');
29 grid on
30 xlabel('\fontsize{14}Time');
```

```

31 ylabel('\fontsize{14}Numbers');
32 legend('\fontsize{16}species 1','\fontsize{16}species 2');
33 title('\fontsize{16}NO.2 [r1, r2] --> [0.3, 0.3]');
34 subplot(1,2,2);
35 plot(y(:,1),y(:,2));
36 grid on;
37 xlabel('\fontsize{16}species 1');
38 ylabel('\fontsize{16}species 2');
39 legend('\fontsize{16}relative');
40
41 %% NO.3 [n1, n2] --> [100, 6000]
42 tspan = [0 20];
43 opt=odeset('reltol',1e-6,'abstol',1e-9);
44 [t,y] = ode45(@odefun,tspan,[10,10],opt,1,1,100,6000,0.5,2);
45 figure;
46 subplot(1,2,1);
47 plot(t,y(:,1),'b--',t,y(:,2),'r--');
48 grid on
49 xlabel('\fontsize{14}Time');
50 ylabel('\fontsize{14}Numbers');
51 legend('\fontsize{16}species 1','\fontsize{16}species 2');
52 title('\fontsize{16}NO.3 [n1, n2] --> [100, 6000]');
53 subplot(1,2,2);
54 plot(y(:,1),y(:,2));
55 grid on;
56 xlabel('\fontsize{16}species 1');
57 ylabel('\fontsize{16}species 2');
58 legend('\fontsize{16}relative');
59
60 %% NO.4 [x0, y0] --> [5, 500]
61 tspan = [0 20];
62 opt=odeset('reltol',1e-6,'abstol',1e-9);
63 [t,y] = ode45(@odefun,tspan,[5,500],opt,1,1,100,100,0.5,2);
64 figure;
65 subplot(1,2,1);
66 plot(t,y(:,1),'b--',t,y(:,2),'r--');
67 grid on
68 xlabel('\fontsize{14}Time');
69 ylabel('\fontsize{14}Numbers');
70 legend('\fontsize{16}species 1','\fontsize{16}species 2');
71 title('\fontsize{16}NO.4 [x0, y0] --> [5, 500]');
72 subplot(1,2,2);
73 plot(y(:,1),y(:,2));
74 grid on;
75 xlabel('\fontsize{16}species 1');
76 ylabel('\fontsize{16}species 2');
77 legend('\fontsize{16}relative');
78
79 %% NO.5 [s1, s2] --> [0.8, 0.7]
80 tspan = [0 20];
81 opt=odeset('reltol',1e-6,'abstol',1e-9);
82 [t,y] = ode45(@odefun,tspan,[10,10],opt,1,1,100,100,0.8,7);
83 figure;
84 subplot(1,2,1);
85 plot(t,y(:,1),'b--',t,y(:,2),'r--');
86 grid on
87 xlabel('\fontsize{14}Time');
88 ylabel('\fontsize{14}Numbers');

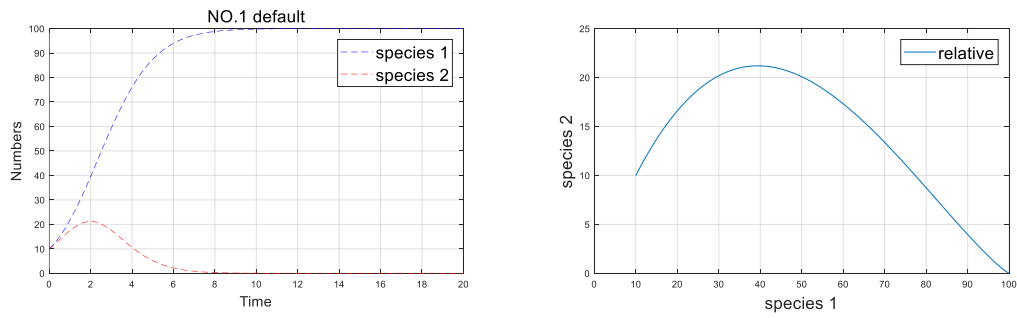
```

```

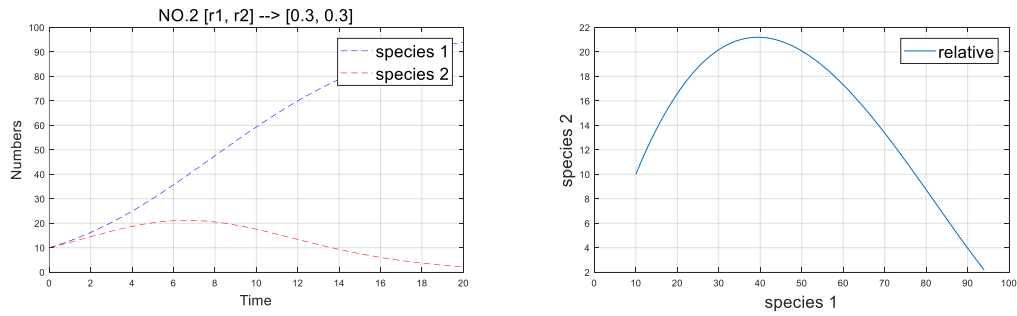
89 legend('\fontsize{16}species 1','\fontsize{16}species 2');
90 title('\fontsize{16}NO.5 [s1, s2] --> [0.8, 0.7]');
91 subplot(1,2,2);
92 plot(y(:,1),y(:,2));
93 grid on;
94 xlabel('\fontsize{16}species 1');
95 ylabel('\fontsize{16}species 2');
96 legend('\fontsize{16}relative');
97
98 %% NO.6 [s1, s2] --> [1.5, 1.7]
99 tspan = [0 20];
100 opt=odeset('reltol',1e-6,'abstol',1e-9);
101 [t,y] = ode45(@odefun,tspan,[10,10],opt,1,1,100,100,1.5,1.7);
102 figure;
103 subplot(1,2,1);
104 plot(t,y(:,1),'b--',t,y(:,2),'r--');
105 grid on
106 xlabel('\fontsize{14}Time');
107 ylabel('\fontsize{14}Numbers');
108 legend('\fontsize{16}species 1','\fontsize{16}species 2');
109 title('\fontsize{16}NO.6 [s1, s2] --> [1.5, 1.7]');
110 subplot(1,2,2);
111 plot(y(:,1),y(:,2));
112 grid on;
113 xlabel('\fontsize{16}species 1');
114 ylabel('\fontsize{16}species 2');
115 legend('\fontsize{16}relative');
116
117 %% NO.7 [s1, s2] --> [1.9, 1.7]
118 tspan = [0 20];
119 opt=odeset('reltol',1e-6,'abstol',1e-9);
120 [t,y] = ode45(@odefun,tspan,[10,10],opt,1,1,100,100,1.9,1.7);
121 figure;
122 subplot(1,2,1);
123 plot(t,y(:,1),'b--',t,y(:,2),'r--');
124 grid on;
125 xlabel('\fontsize{14}Time');
126 ylabel('\fontsize{14}Numbers');
127 legend('\fontsize{16}species 1','\fontsize{16}species 2');
128 title('\fontsize{16}NO.7 [s1, s2] --> [1.9, 1.7]');
129 subplot(1,2,2);
130 plot(y(:,1),y(:,2));
131 grid on;
132 xlabel('\fontsize{16}species 1');
133 ylabel('\fontsize{16}species 2');
134 legend('\fontsize{16}relative');

```

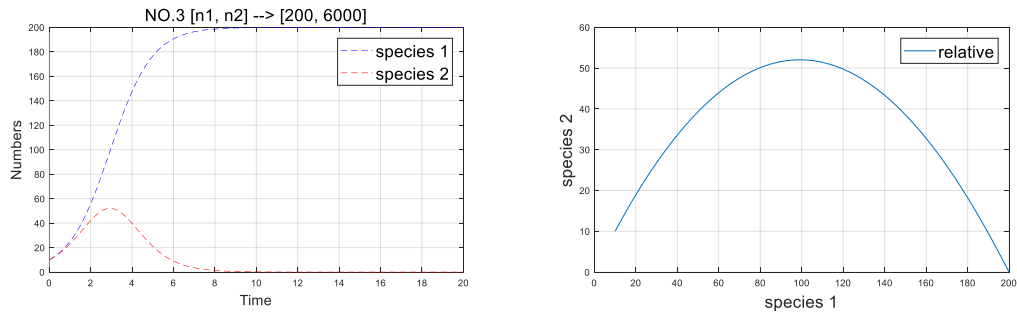
程序代码 5



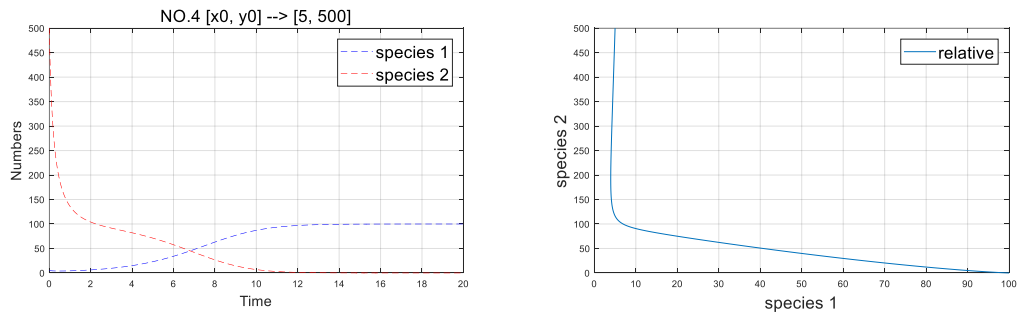
运行结果 4



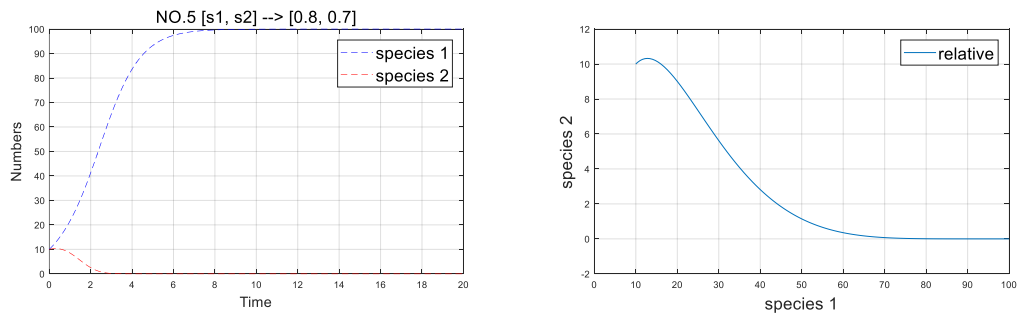
运行结果 5



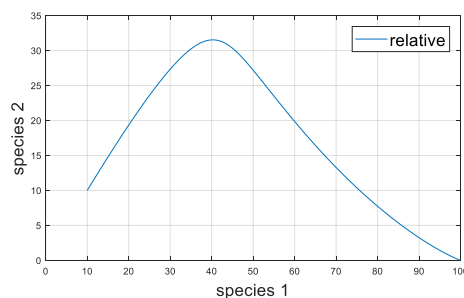
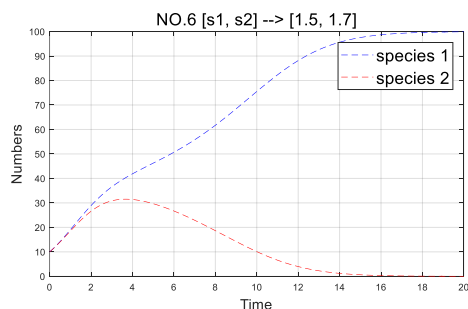
运行结果 6



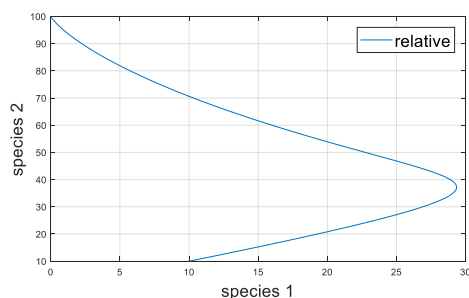
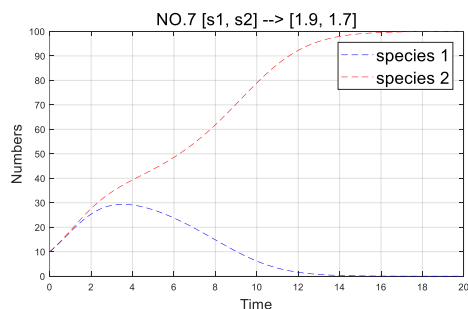
运行结果 7



运行结果 8



运行结果 9



运行结果 10

注意：以上图片都保留了相对比例。

可以看出，（2（3）问中的调节参数问题，在图形中得到了很好的诠释：

- 决定最终走向的是 s 参数， s 参数可以视为是竞争力参数， s 相对高的那一个物种将最终存活下去。
- 从运行结果 5 来看， r 参数，即固有增长率，决定了演化的速度，这个参数指的就是该物种的繁殖能力；
- 从运行结果 6 来看， n 参数，即最大容量，决定了胜出一方的繁殖上线，这个参数指的就是环境对于该物种的承载能力；
- 从运行结果 7 来看，初始值参数并不能代表一段时间以后的最终走向，而仅仅是代表了最开始一段时间的种群数量相对大小，而如果种群数量较大的那一个 s 值小，那么必然会有一天，该种群的种群数量变小，最终消亡。
- 从运行结果 8、运行结果 9、运行结果 10 来看， s 的相对大小都决定了最终的演化走向，在两者的 s 值的方差较小而初始值很接近的情况下，两者的走向是前段发展时间较为接近，后一段时间完全两极分化，一个繁荣一个灭亡。若 s 的方差很大，而初始值很接近，那么两者的走向是立即分化，一个锐减，一个猛增。

三、实验环境

Windows10 Enterprise 1703 中文版操作系统；
MATLAB R2017a 中文版。

四、实验过程

设计算法，编写程序并调试。

五、实验总结

这一章涉及范围比较广，数学分析、概率统计、常微分方程，都有涉及，而且包含了数学实现，复杂度相对较高。好在基础还可以，有些问题结合一下当时的原理能很快领会，也可以较快明白 MATLAB 代码的写法。

在题 3 中，详细进行了一个模型的数学分解，虽然没有进行建模，而是直接使用了书上的模型，不过应该算是这门课程的第一次模型接触。通过对数据的直观感受，分析出数据中参数的意义，然后产生相关猜测，最后用 MATLAB 进行试验，验证与细化猜测，最终分析完整个模型。这是数学建模的第二步（第一步当然是建模与抽象）。

六、参考文献

- [1] 大学数学实验/姜启源，谢金星，邢文训，张立平，北京：清华大学出版社，2010.12
- [2] MATLAB 教程/张志涌，杨祖樱，北京：北京航空航天大学出版社，2015.1

七、教师评语