数学建模与 Matlab

第一章对变化进行建模

韩建伟

信息学院
mm@hanjianwei.com

2016/09/13

课程介绍

内容 数学建模 + Matlab

教材 数学建模 (原书第 5 版), Frank R.Giordano, Willam P.Fox, Steven B.Horton, 叶其孝 / 姜启源译, 机械工业出版社

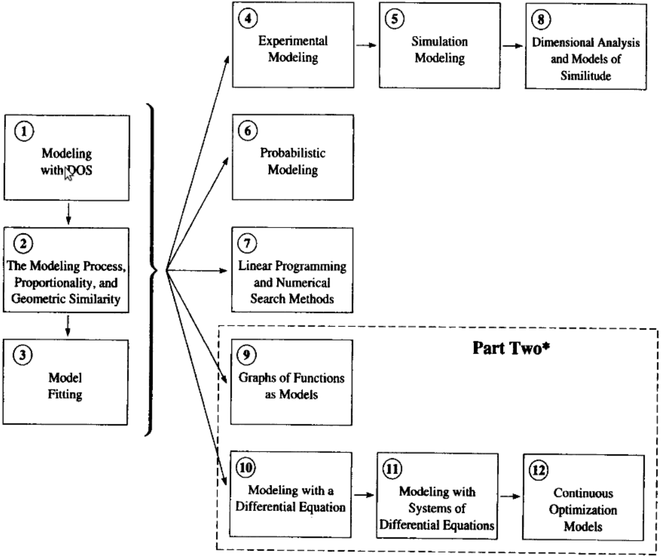
参考 数学模型 (第四版), 姜启源 / 谢金星 / 叶俊, 高等教育出版社
数学建模, 杨启帆 / 谈之奕 / 何勇, 浙江大学出版社
精通 MATLAB R2011a, 张志涌, 北京航天航空大学出版社

网站 http://zjsu.github.io/mm

课程安排

- **时间** 15 周 = 9 课堂 + 5 实验 + 1 假期
- **考试** 成绩 = 平时 × 30%+ 期末 (闭卷) × 70%
- **作业** 作业 = 小作业 + 大作业 (2-3 人一个团队)
- **提交** 相应的提交方式将在网站公布

课程概要

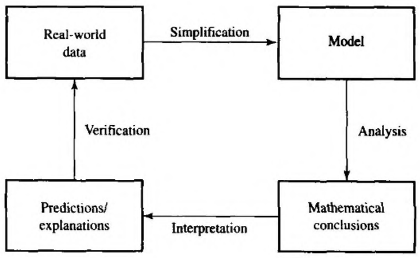


*Part Two requires single-variable calculus as a corequisite.

简介

数学模型是对现实世界现象的理想化表示而非完全精确的表示.

- 预测变化
- 对现实世界进行简化



例如: 比例性.

弹簧系统

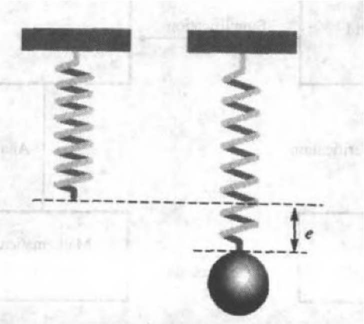


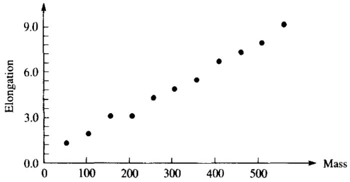
Table 1.1 Spring–mass system

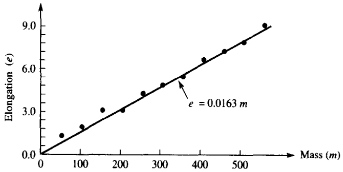
Mass	Elong
50	1.000
100	1.875
150	2.750
200	3.250
250	4.375
300	4.875
350	5.675
400	6.500
450	7.250
500	8.000
550	8.750

$$\textit{slope} = \frac{4.875 - 3.25}{300 - 200} = 0.01625$$

验证弹簧系统

$$e = 0.0163$$





对变化进行建模

未来值 = 现在值 + 变化
变化 = 未来值 - 现在值

- 离散时间 差分方程 (difference equation)
- 连续时间 微分方程 (第 11 章)

差分方程

定义

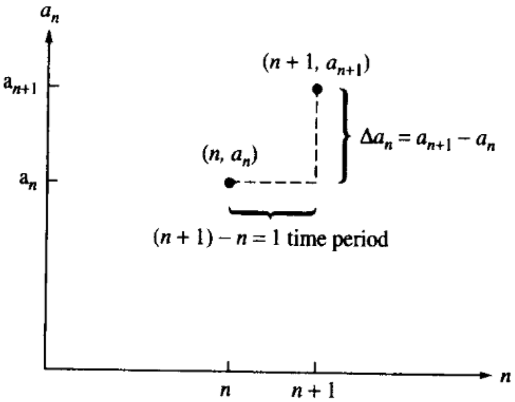
对于数列 $A = a_0, a_1, a_2, a_3, ...$, 其一阶差分定义为:

$$
\begin{aligned}
\Delta a_0 &= a_1 - a_0 \\
\Delta a_1 &= a_2 - a_1 \\
\Delta a_2 &= a_3 - a_2 \\
\Delta a_3 &= a_4 - a_3
\end{aligned}
$$

对于每个正整数 n, 第 n 个一阶差分为:

$$\Delta a_n = a_{n+1} - a_n$$

差分方程



储蓄问题

考虑本金 1000 美元, 月利息 1% 的储蓄问题
$A=(1000,1010,1020.10,1030.30,...)$.

$$\begin{array}{l}\Delta a_0=a_1-a_0=1010.0-1000.0=10.0\\ \Delta a_1=a_2-a_1=1020.1-1010.0=10.1\\ \Delta a_2=a_3-a_2=1030.3-1020.1=10.2\end{array}$$

$$\begin{array}{c}\Delta a_n=a_{n+1}-a_n=0.01a_n\\ a_{n+1}=a_n+0.01a_n=1.01a_n,n=0,1,2,3\\ a_0=1000\end{array}$$

如果每月取出 50 美元...

$$\Delta a_n=a_{n+1}-a_n=0.01a_n-50$$

如何找出变化

在多数情况下，很难象上述例子那样精确表述，因此我们通过如下步骤找出变化:

1. 画出变化
2. 观察变化规律
3. 用数学术语描述变化

变化 $= \Delta a_n =$ 某个函数 f
对于离散情况:
变化 $= \Delta a_n = a_{n+1} - a_n = f($序列中的项, 外部项$)$

按揭买房

六年前按揭 20 年买了一套 80000 美元的房子, 月供 880.87 美元并付每月 1% 的利息. 问现在还欠银行多少?

$$\Delta b_n = b_{n+1} - b_n = 0.01 b_n - 880.87$$

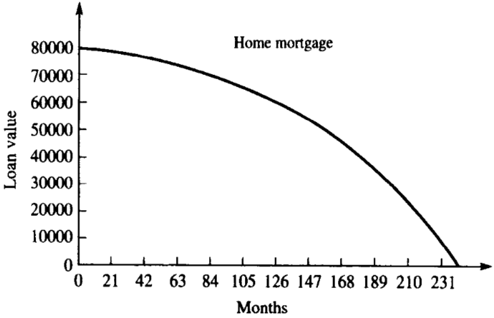
求解下列方程即可:

$$b_{n+1} = b_n + 0.01 b_n - 880.87$$
$$b_0 = 80000$$

$B = (80000, 79919.13, 79837.45, ...)$

按揭买房

Months n	Amount Owed b_n
0	80000.00
1	79919.13
2	79837.45
3	79754.96
4	79671.64
5	79587.48
6	79502.49
7	79416.64
8	79329.94
9	79242.37
10	79153.92
11	79064.59
12	78974.37

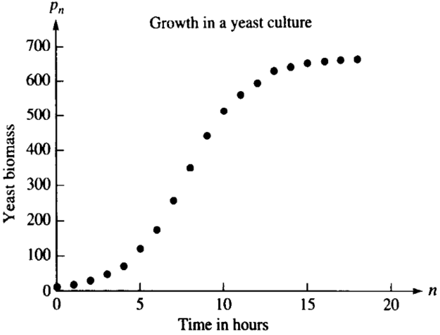


用差分方程来近似变化

- 变化 $= \Delta a_n =$ 某个函数 f
- 离散变化与连续变化
- 模型的细化: 生、死、资源

酵母培养 – 找出模型

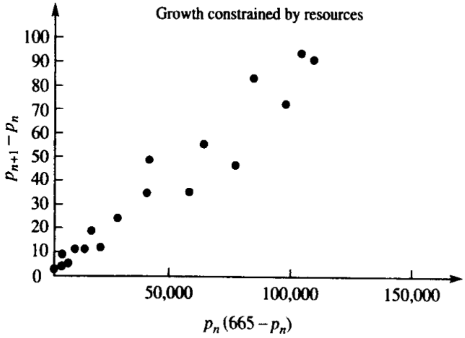
Time in hours n	Yeast biomass p_n	Change/ hour $p_{n+1}-p_n$
0	9.6	8.7
1	18.3	10.7
2	29.0	18.2
3	47.2	23.9
4	71.1	48.0
5	119.1	55.5
6	174.6	82.7
7	257.3	93.4
8	350.7	90.3
9	441.0	72.3
10	513.3	46.4
11	559.7	35.1
12	594.8	34.6
13	629.4	11.4
14	640.8	10.3
15	651.1	4.8
16	655.9	3.7
17	659.6	2.2
18	661.8	



$$\Delta p_n = p_{n+1} - p_n = k(665 - p_n)p_n$$

酵母培养 – 模型数值求解

$p_{n+1}-p_n$	$p_n(665-p_n)$
8.7	6291.84
10.7	11,834.61
18.2	18,444.00
23.9	29,160.16
48.0	42,226.29
55.5	65,016.69
82.7	85,623.84
93.4	104,901.21
90.3	110,225.01
72.3	98,784.00
46.4	77,867.61
35.1	58,936.41
34.6	41,754.96
11.4	22,406.64
10.3	15,507.36
4.8	9050.29
3.7	5968.69
2.2	3561.84

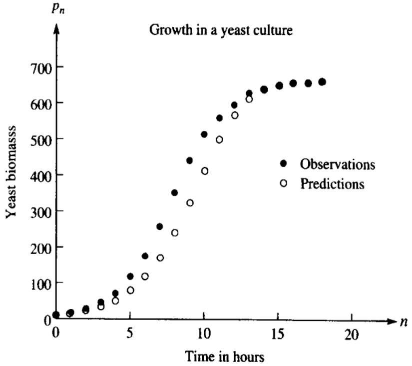


$$k \approx 0.00082$$

$$p_{n+1} = p_n + 0.00082(665 - p_n)p_n$$

酵母培养 – 模型验证

Time in hours	Observations	Predictions
0	9.6	9.6
1	18.3	14.8
2	29.0	22.6
3	47.2	34.5
4	71.1	52.4
5	119.1	78.7
6	174.6	116.6
7	257.3	169.0
8	350.7	237.8
9	441.0	321.1
10	513.3	411.6
11	559.7	497.1
12	594.8	565.6
13	629.4	611.7
14	640.8	638.4
15	651.1	652.3
16	655.9	659.1
17	659.6	662.3
18	661.8	663.8



$$p_{n+1} = p_n + 0.00082(665 - p_n)p_n$$

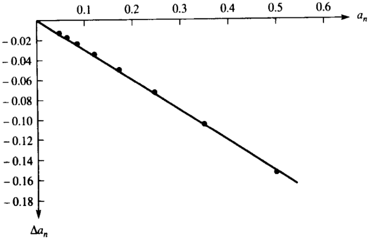
思考: 看书上例 2 传染病传播模型, 考虑如何为该模型加入其它因素?

地高辛在血流中的变化

Table 1.2 The change a_n in digoxin in a patient's bloodstream

n	0	1	2	3	4	5	6	7	8
a_n	0.500	0.345	0.238	0.164	0.113	0.078	0.054	0.037	0.026
Δa_n	−0.155	−0.107	−0.074	−0.051	−0.035	−0.024	−0.017	−0.011	

Figure 1.11
A plot of Δa_n versus a_n from Table 1.2 suggests a straight line through the origin

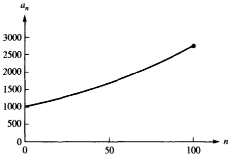


$$\Delta a_n = -0.31 a_n$$
$$a_{n+1} - a_n = -0.31 a_n$$
$$a_{n+1} = 0.69 a_n$$

动态系统的解法 – 猜测

存款问题: $a_{n+1} = 1.01a_n$, $a_0 = 1000$

n	a_n
0	1000.00
1	1010.00
2	1020.10
3	1030.30
4	1040.60
5	1051.01
6	1061.52
7	1072.14
8	1082.86
9	1093.69
10	1104.62
11	1115.67
12	1126.83
13	1138.09
14	1149.47
15	1160.97
16	1172.58
17	1184.30



$$
\begin{aligned}
a_1 &= 1010.0 = 1.01(1000)\\
a_2 &= 1020.1 = 1.01(1010) = 1.01^2(1000)\\
a_3 &= 1030.3 = 1.01(1020.1) = 1.01^3(1000)\\
a_4 &= 1040.6 = 1.01(1030.3) = 1.01^4(1000)
\end{aligned}
$$

动态系统的解法 – 猜测

猜测: $a_k = 1.01^k(1000)$
验证、结论: ...

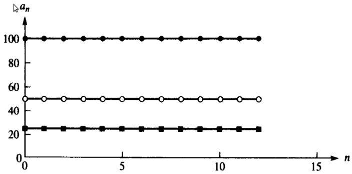
推测法的一般步骤

1. 观察模式
2. 猜测动力系统的形式
3. 用带入法来测试该猜测
4. 接受或拒绝该推测：取决于代入和代数运算后结果是否满足该动力系统。

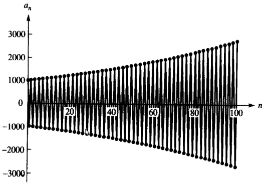
推论：形式为 $a_{n+1} = ra_n$ 的动态系统的解为 $a_k = r^k a_0$.

$a_{n+1} = ra_n$

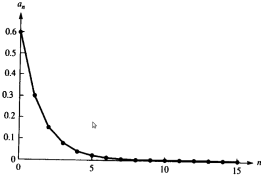
$r = ?$



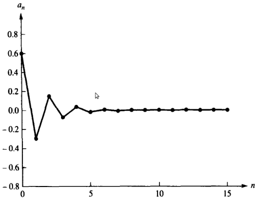
n	a_n
0	1000
1	−1010.00
2	1020.10
3	−1030.30
4	1040.60
5	−1051.01
6	1061.52
7	−1072.14
8	1082.86
9	−1093.69
10	1104.62
11	−1115.67
12	1126.83
13	−1138.09
14	1149.47
15	−1160.97
16	1172.58



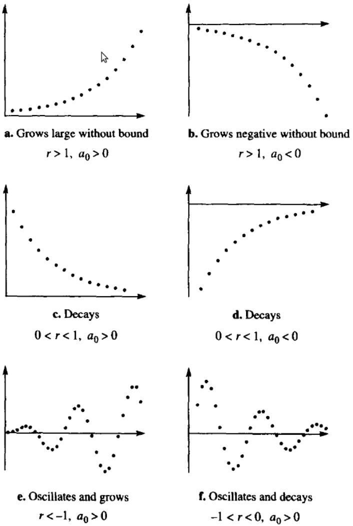
n	a_n
0	0.6
1	0.3
2	0.15
3	0.075
4	0.0375
5	0.01875
6	0.009375
7	0.0046875
8	0.00234375
9	0.00117188
10	0.00058594
11	0.00029297
12	0.00014648
13	7.3242E-05
14	3.6621E-05
15	1.8311E-05



n	a_n
0	0.6
1	−0.3
2	0.15
3	−0.075
4	0.0375
5	−0.01875
6	0.009375
7	−0.0046875
8	0.00234375
9	−0.00117188
10	0.00058594
11	−0.00029297
12	0.00014648
13	−7.3242E-05
14	3.6621E-05
15	−1.8311E-05

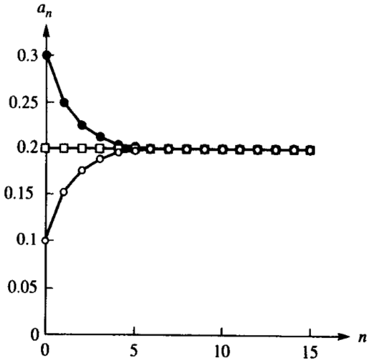


$$a_{n+1} = ra_n + b$$


a. Grows large without bound
$r > 1,\ a_0 > 0$
b. Grows negative without bound
$r > 1,\ a_0 < 0$
c. Decays
$0 < r < 1,\ a_0 > 0$
d. Decays
$0 < r < 1,\ a_0 < 0$
e. Oscillates and grows
$r < -1,\ a_0 > 0$
f. Oscillates and decays
$-1 < r < 0,\ a_0 > 0$

不动点 (平衡点)

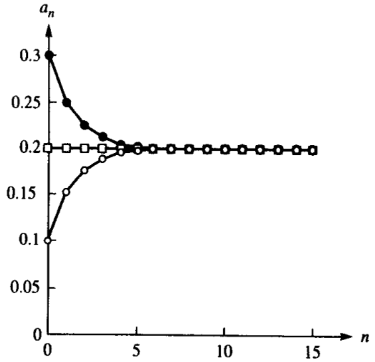
	A	B	C
n	a_n	a_n	a_n
0	0.1	0.2	0.3
1	0.15	0.2	0.25
2	0.175	0.2	0.225
3	0.1875	0.2	0.2125
4	0.19375	0.2	0.20625
5	0.196875	0.2	0.203125
6	0.1984375	0.2	0.2015625
7	0.19921875	0.2	0.20078125
8	0.19960938	0.2	0.20039063
9	0.19980469	0.2	0.20019531
10	0.19990234	0.2	0.20009766
11	0.19995117	0.2	0.20004883
12	0.19997559	0.2	0.20002441
13	0.19998779	0.2	0.20001221
14	0.1999939	0.2	0.2000061
15	0.19999695	0.2	0.20000305



Digoxin 浓度变化

不动点

	A	B	C
n	a_n	a_n	a_n
0	0.1	0.2	0.3
1	0.15	0.2	0.25
2	0.175	0.2	0.225
3	0.1875	0.2	0.2125
4	0.19375	0.2	0.20625
5	0.196875	0.2	0.203125
6	0.1984375	0.2	0.2015625
7	0.19921875	0.2	0.20078125
8	0.19960938	0.2	0.20039063
9	0.19980469	0.2	0.20019531
10	0.19990234	0.2	0.20009766
11	0.19995117	0.2	0.20004883
12	0.19997559	0.2	0.20002441
13	0.19998779	0.2	0.20001221
14	0.1999939	0.2	0.2000061
15	0.19999695	0.2	0.20000305

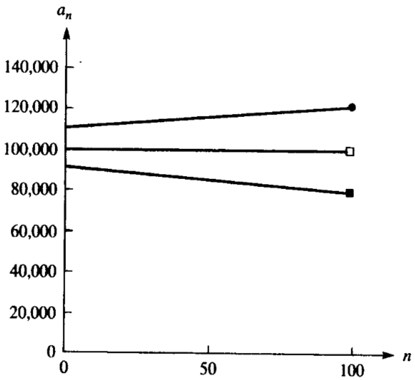


Digoxin 浓度变化

不动点

$$a_{n+1} = 1.01a_n - 1000$$

	A	B	C
n	a_n	a_n	a_n
0	90000	100000	110000
1	89900	100000	110100
2	89799	100000	110201
3	89697	100000	110303
4	89594	100000	110406
5	89490	100000	110510
6	89385	100000	110615
7	89279	100000	110721
8	89171	100000	110829
9	89063	100000	110937
10	88954	100000	111046
11	88843	100000	111157
12	88732	100000	111268
13	88619	100000	111381
14	88505	100000	111495
15	88390	100000	111610

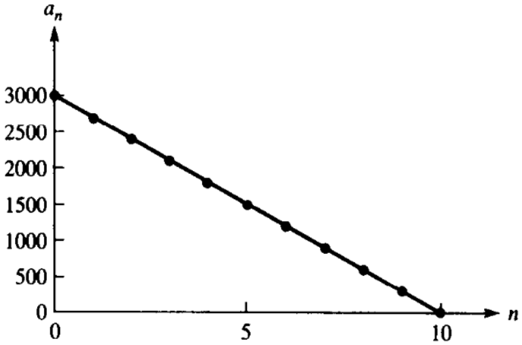


投资

$r = 1$

$$a_{n+1} = a_n - 300$$

n	a_n
0	3000
1	2700
2	2400
3	2100
4	1800
5	1500
6	1200
7	900
8	600
9	300
10	0



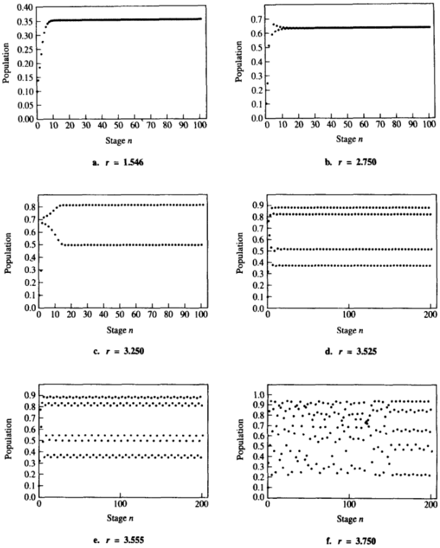
不动点

$a_{n+1} = ra_n + b, r \neq 1$ 的不动点为：

$$a = \frac{b}{1-r}$$

上述动态系统的解为：$a_k = r^k c + \frac{b}{1-r}$.
思考：r 的值不同时，长期来说系统如何变化?

非线性系统


Population
Stage n
a. r = 1.546
b. r = 2.750
c. r = 3.250
d. r = 3.525
e. r = 3.555
f. r = 3.750

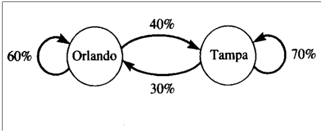
差分方程组

- 找出不动点
- 当初始值在不动点附近时，系统如何变化

研究系统的长期变化，看系统对如下条件是否敏感：

- 初始条件
- 对模型中的常量进行扰动

汽车租赁公司



定义

- O_n = 第 n 天营业结束时在奥兰多的车辆数
- T_n = 第 n 天营业结束时在坦帕的车辆数

$$O_{n+1} = 0.6O_n + 0.3T_n$$

$$T_{n+1} = 0.4O_n + 0.7T_n$$

计算平衡点

如果存在平衡点 O, T:

$$O = O_{n+1} = O_n$$

$$T = T_{n+1} = T_n$$

推导出:

$$O = 0.6O + 0.3T$$

$$T = 0.4O + 0.7T$$

方程求解

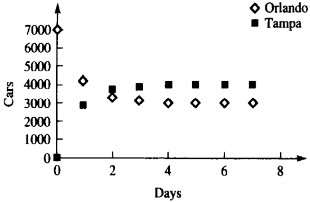
$O = \frac{3}{4}T$ 满足上述方程组. 如果公司有 7000 辆车，则 $(O, T) = (3000, 4000)$ 处开始，保持不变。
分析下述四种初始条件:

Four starting values for the car rental problem

	Orlando	Tampa
Case 1	7000	0
Case 2	5000	2000
Case 3	2000	5000
Case 4	0	7000

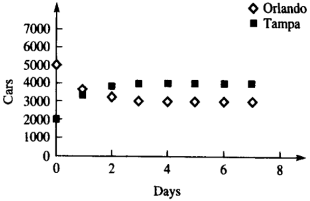
分析

n	Orlando	Tampa
0	7000	0
1	4200	2800
2	3360	3640
3	3108	3892
4	3032.4	3967.6
5	3009.72	3990.28
6	3002.916	3997.084
7	3000.875	3999.125



a. Case 1

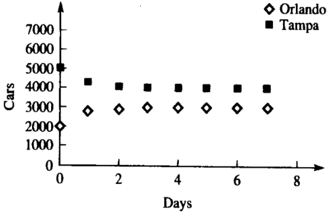
n	Orlando	Tampa
0	5000	2000
1	3600	3400
2	3180	3820
3	3054	3946
4	3016.2	3983.8
5	3004.86	3995.14
6	3001.458	3998.542
7	3000.437	3999.563



b. Case 2

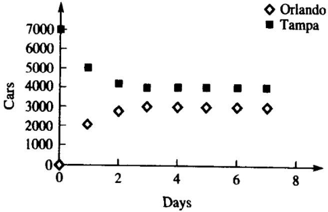
分析

n	Orlando	Tampa
0	2000	5000
1	2700	4300
2	2910	4090
3	2973	4027
4	2991.9	4008.1
5	2997.57	4002.43
6	2999.271	4000.729
7	2999.781	4000.219



c. Case 3

n	Orlando	Tampa
0	0	7000
1	2100	4900
2	2730	4270
3	2919	4081
4	2975.7	4024.3
5	2992.71	4007.29
6	2997.813	4002.187
7	2999.344	4000.656



d. Case 4

结论

四种情形中每一种情形在一周内都是和平衡点 $(3000, 4000)$ 很接近的，甚至在其中一个城市没有车的情况也是如此。结果显示，平衡点是稳定的而且对初始值不敏感的。

思考: 该系统是否对 O_{n+1} 和 T_{n+1} 的系数敏感?

特拉法尔加战斗

法西联军 33 艘战舰，英军 27 艘战舰，在一次遭遇战中每方的战舰损失都是对方战舰的 10%。

动力系统模型 令 n 表示战斗过程中遭遇战的阶段并定义：

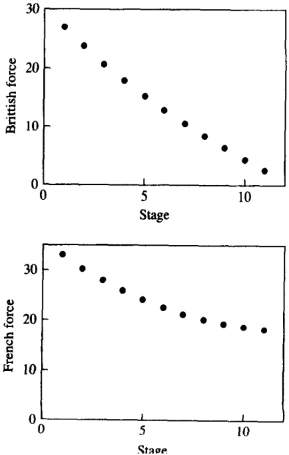
定义

- B_n = 第 n 阶段英军的战舰数
- F_n = 第 n 阶段法西联军的战舰数

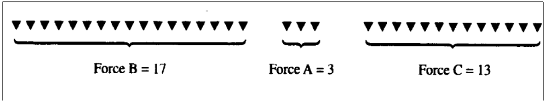
死拼打法

战斗结束：英军全面战败，剩 3 艘战舰其中一艘严重损坏，法军大约还有 18 艘战舰。

Stage	Brittish force	French force
1	27.0000	33.0000
2	23.7000	30.3000
3	20.6700	27.9300
4	17.8770	25.8630
5	15.2907	24.0753
6	12.8832	22.5462
7	10.6285	21.2579
8	8.5028	20.1951
9	6.4832	19.3448
10	4.5488	18.6965
11	2.6791	18.2416



各个击破

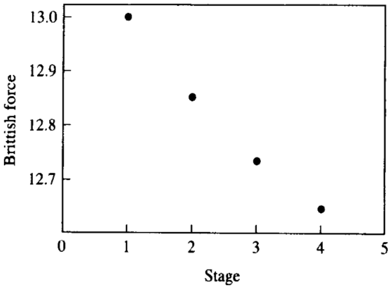


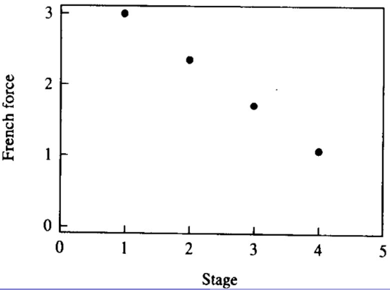
策略：英军 13 艘攻击 A；然后，全力攻击 B，最后攻击 C。

战斗 A

Battle A

Stage	British force	French force
1	13.0000	3.00000
2	12.8500	2.35000
3	12.7325	1.70750
4	12.6471	1.07088

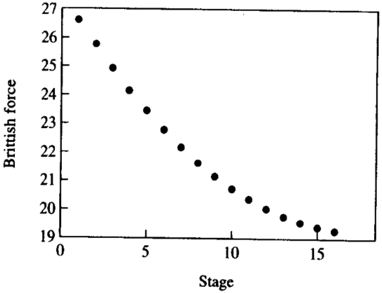


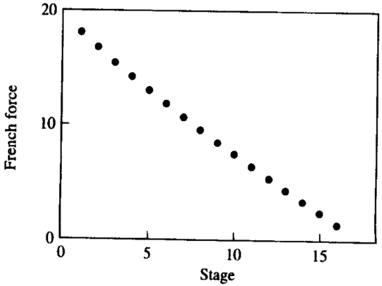


战斗 B

Battle B

Stage	British force	French force
1	26.6471	18.0709
2	25.7436	16.7385
3	24.9066	15.4513
4	24.1341	14.2060
5	23.4238	12.9993
6	22.7738	11.8281
7	22.1824	10.6894
8	21.6479	9.5803
9	21.1689	8.4979
10	20.7440	7.4395
11	20.3720	6.4023
12	20.0519	5.3837
13	19.7827	4.3811
14	19.5637	3.3919
15	19.3941	2.4138
16	19.2734	1.4441

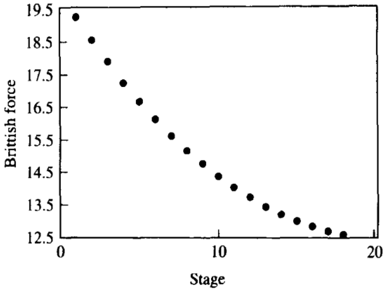


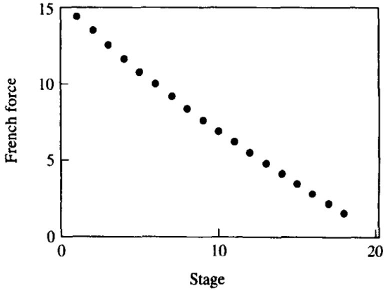


战斗 C

Battle C

Stage	British force	French force
1	19.2734	14.4441
2	18.5512	13.4804
3	17.8772	12.5529
4	17.2495	11.6590
5	16.6666	10.7965
6	16.1268	9.9632
7	15.6286	9.1569
8	15.1707	8.3754
9	14.7520	7.6169
10	14.3711	6.8793
11	14.0272	6.1607
12	13.7191	5.4594
13	13.4462	4.7734
14	13.2075	4.1011
15	13.0024	3.4407
16	12.8304	2.7906
17	12.6909	2.1491
18	12.5834	1.5146





战果

英军大获全胜。现实世界：法西联军没有参加战斗 C，而是把剩下的约 13 艘战舰撤回法国。

斑点猫头鹰和隼

- O_n, H_n 分别表示第 n 天猫头鹰和隼的数量

斑点猫头鹰和隼

- O_n, H_n 分别表示第 n 天猫头鹰和隼的数量
- $\Delta O_n = k_1 O_n$, $\Delta H_n = k_2 H_n$ (不考虑竞争)

斑点猫头鹰和隼

- O_n, H_n 分别表示第 n 天猫头鹰和隼的数量
- $\Delta O_n = k_1 O_n$, $\Delta H_n = k_2 H_n$ (不考虑竞争)
- $\Delta O_n = k_1 O_n - k_3 O_n H_n$, $\Delta H_n = k_2 H_n - k_4 O_n H_n$ (考虑竞争)

斑点猫头鹰和隼

- O_n, H_n 分别表示第 n 天猫头鹰和隼的数量
- $\Delta O_n = k_1 O_n$, $\Delta H_n = k_2 H_n$ (不考虑竞争)
- $\Delta O_n = k_1 O_n - k_3 O_n H_n$, $\Delta H_n = k_2 H_n - k_4 O_n H_n$ (考虑竞争)
- $O_{n+1} = (1 + k_1) O_n - k_3 O_n H_n$
- $H_{n+1} = (1 + k_2) H_n - k_4 O_n H_n$

求解平衡点

定义

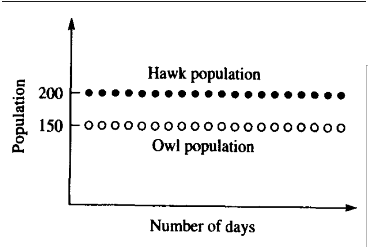
- $O_{n+1} = 1.2O_n - 0.001O_nH_n$
- $H_{n+1} = 1.3H_n - 0.002O_nH_n$

如果 (O, H) 为平衡点则 $O_{n+1} = O_n = O,\ H_{n+1} = H_n = H$:

- $O = 1.2O - 0.001OH \Rightarrow O = 0 or H = 200$
- $H = 1.3H - 0.002OH \Rightarrow H = 0 or O = 150$

平衡点分析

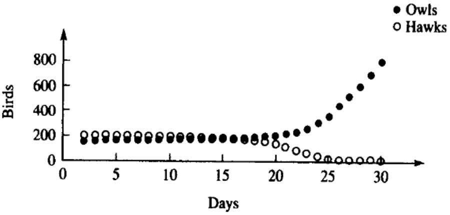
两个平衡点: $(0,0)$, $(150,200)$. 为什么?



	Owls	Hawks
Case 1	151	199
Case 2	149	201
Case 3	10	10

情况 1

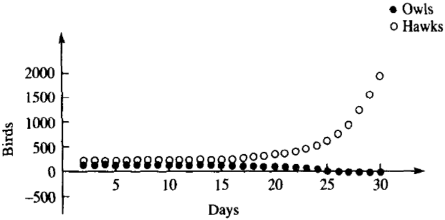
n	Owls	Hawks
1	151	199
2	151.151	198.602
3	151.3623	198.1448
4	151.6431	197.6049
5	152.0063	196.9556
6	152.4691	196.1653
7	153.0538	195.1966
8	153.7889	194.0044
9	154.711	192.5343
10	155.866	190.7202
11	157.3124	188.4827
12	159.1242	185.7261
13	161.3956	182.3369
14	164.2463	178.1812
15	167.83	173.1044
16	172.3438	166.9315
17	178.043	159.4717
18	185.2588	150.5276
19	194.424	139.9128
20	206.1064	127.4818
21	221.0528	113.1767
22	240.2454	97.09366
23	264.9681	79.56915
24	296.8785	61.27332
25	338.0634	43.27385
26	391.0468	26.99739
27	458.6989	13.98212
28	544.0252	5.349589
29	649.9199	1.133844
30	779.1669	0.000182



a. Case 1

情况 2

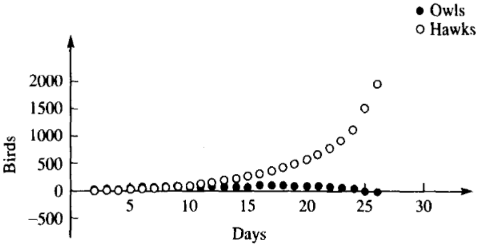
n	Owls	Hawks
1	149	201
2	148.851	201.402
3	148.6423	201.8648
4	148.3651	202.413
5	148.0071	203.0748
6	147.552	203.8842
7	146.9789	204.8824
8	146.2613	206.1204
9	145.3661	207.6616
10	144.2524	209.5862
11	142.8696	211.9954
12	141.1558	215.0186
13	139.0358	218.822
14	136.4189	223.6204
15	133.1966	229.6944
16	129.2414	237.4137
17	124.406	247.2705
18	118.5253	259.9277
19	111.4223	276.29
20	102.9219	297.6073
21	92.87598	325.6289
22	81.20808	362.8313
23	67.98486	412.751
24	53.52101	480.4547
25	38.51079	573.1623
26	24.14002	700.9651
27	12.04671	877.412
28	3.886124	1119.496
29	0.31285	1446.643
30	−0.07716	1879.731



b. Case 2

情况 3

n	Owls	Hawks
1	10	10
2	11.9	12.8
3	14.12768	16.33536
4	16.72244	20.77441
5	19.71952	26.31193
6	23.14457	33.16779
7	27.00583	41.58282
8	31.28402	51.81171
9	35.91994	64.11347
10	40.80098	78.7416
11	45.74844	95.93862
12	50.50908	115.9421
13	54.75477	139.0125
14	58.09413	165.493
15	60.09878	195.9126
16	60.34443	231.1382
17	58.46541	272.5838
18	54.22177	322.4855
19	47.58039	384.2597
20	38.81324	462.9712
21	28.60647	565.9237
22	18.13869	703.3227
23	9.009075	888.8048
24	2.803581	1139.432
25	0.169808	1474.872
26	−0.04668	1916.833



c. Case 3

对初始条件的敏感性和长期行为

如果在栖息地安置 350 只猫头鹰和隼：

1. 如果 150 头为猫头鹰：猫头鹰和隼的数量不变（150、200）

对初始条件的敏感性和长期行为

如果在栖息地安置 350 只猫头鹰和隼：

1. 如果 150 头为猫头鹰：猫头鹰和隼的数量不变（150、200）
2. 如果 149 头或更少猫头鹰：猫头鹰将灭绝

对初始条件的敏感性和长期行为

如果在栖息地安置 350 只猫头鹰和隼：

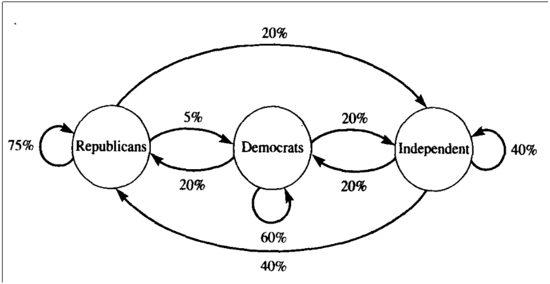
1. 如果 150 头为猫头鹰：猫头鹰和隼的数量不变（150、200）
2. 如果 149 头或更少猫头鹰：猫头鹰将灭绝
3. 如果 151 头或更多猫头鹰：隼将灭绝

对初始条件的敏感性和长期行为

如果在栖息地安置 350 只猫头鹰和隼：

1. 如果 150 头为猫头鹰：猫头鹰和隼的数量不变（150、200）
2. 如果 149 头或更少猫头鹰：猫头鹰将灭绝
3. 如果 151 头或更多猫头鹰：隼将灭绝
4. 该模型对初始条件极其敏感，平衡点不稳定。

政党投票



定义

R_n = 第 n 次选举投共和党票的人数
D_n = 第 n 次选举投民主党票的人数
I_n = 第 n 次选举投独立派票的人数

差分方程组

$$R_{n+1} = 0.75R_n + 0.20D_n + 0.40I_n$$

$$D_{n+1} = 0.05R_n + 0.60D_n + 0.20I_n$$

$$I_{n+1} = 0.20R_n + 0.20D_n + 0.40I_n$$

平衡点 $R_{n+1} = R_n = R$, $D_{n+1} = D_n = D$, $I_{n+1} = I_n = I$:

$$-0.25R + 0.20D + 0.40I = 0$$

$$0.05R - 0.40D + 0.20I = 0$$

$$0.20R + 0.20D - 0.60I = 0$$

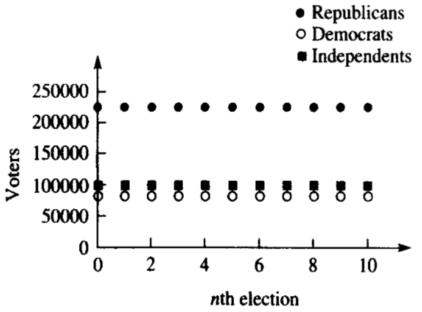
平衡点分析

$R : D : I = 2.2221 : 0.7777694 : 1$

	Republicans	Democrats	Independents
Case 1	222,221	77,777	100,000
Case 2	227,221	82,777	90,000
Case 3	100,000	100,000	199,998
Case 4	0	0	399,998

情况 1

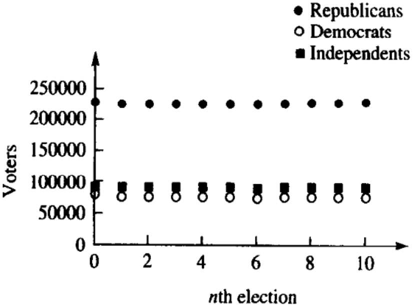
n	Republicans	Democrats	Independents
0	222221	77777	100000
1	222221.2	77777.25	99999.6
2	222221.2	77777.33	99999.52
3	222221.1	77777.36	99999.5
4	222221.1	77777.37	99999.5
5	222221.1	77777.38	99999.5
6	222221.1	77777.38	99999.5
7	222221.1	77777.39	99999.5
8	222221.1	77777.39	99999.5
9	222221.1	77777.39	99999.5
10	222221.1	77777.39	99999.5



a. Case 1

情况 2

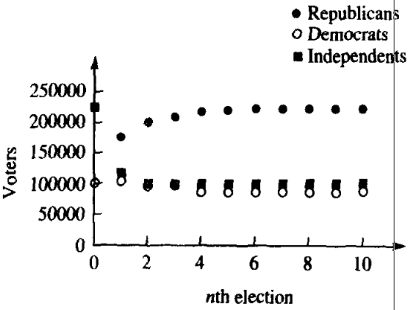
n	Republicans	Democrats	Independents
0	227221	82777	90000
1	222971.2	79027.25	97999.6
2	222233.7	78164.83	99599.52
3	222148	77930.48	99919.5
4	222164.9	77849.59	99983.5
5	222187	77814.7	99996.3
6	222201.7	77797.43	99998.86
7	222210.3	77788.32	99999.37
8	222215.1	77783.38	99999.47
9	222217.8	77780.68	99999.49
10	222219.3	77779.2	99999.5



b. Case 2

情况 3

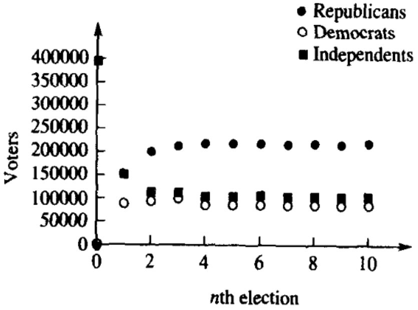
n	Republicans	Democrats	Independents
0	100000	100000	199998
1	174999.2	104999.6	119999.2
2	200249	95749.56	103999.4
3	210936.4	88262.07	100799.5
4	216174.5	83663.96	100159.5
5	218927.5	81039	100031.5
6	220416	79576.08	100005.9
7	221229.6	78767.63	100000.8
8	221676	78322.21	99999.76
9	221921.4	78077.08	99999.55
10	222056.3	77942.23	99999.51



c. Case 3

情况 4

n	Republicans	Democrats	Independents
0	0	0	399998
1	159999.2	79999.6	159999.2
2	199999	87999.56	111999.4
3	212398.9	85199.57	102399.5
4	217298.9	82219.59	100479.5
5	219609.9	80292.6	100095.5
6	220804.1	79175.15	100018.7
7	221445.6	78549.04	100003.3
8	221795.4	78202.37	100000.3
9	221987.1	78011.25	99999.65
10	222092.4	77906.03	99999.53



d. Case 4

总结

选举系统相当稳定，即使刚开始没有人选共和党或者民主党，最后也会稳定.