

数学实验四

1. 常见信贷方式：

A. **到期一次性还本息**，名义利率等于实际利率，客户在银行的定期存款就是这种方式。

B. **先按月还息，到期还本**，实际利率接近 A 的利率，是信用贷款常用的做法。

C. **等额本息法**，按月等额还本金和利息，银行贷款的最常见方式。

D. **等额本金法**，按月等还本金，并还相应利息，银行房贷的次常见方式。

E. **借贷时先一次性扣除利息**，接下来每月还本金。

A 君于本日向银行存入银行固定存款 100 万元，整存整取，期限 2 年，约定到期自动转存2 年，且假设利率不变。银行当日转手将该款项贷给 B 君，期限为 10 年。

(1) 请建立数学模型，描述 A 君和 B 君在该银行所设专项账户逐年余额情况。

(2) 请建立数学模型，分别描述 **A-E** 5 种情况下 B 君在该银行所设专项账户逐期余额情况。

(3) 根据以上模型，比较 A-E 5 种情况下，存贷款的实际利率。

(4) 讨论银行盈亏情况。

(5) 如果利率允许波动(如 LPR)，做合理假设，重新考虑以上问题。

请查阅银行最新存贷款利率

解：(1)请建立数学模型，描述 A 君和 B 君在该银行所设专项账户逐年余额情况。

- **a)** 查阅资料：查阅华夏银行的存款与贷款利率，得知 2 年期存款年利率 $r_1=1.8\%$ ，5 年期以上的贷款年利率 $r_2=3.95\%$ 。

2024年华夏银行存款利率一览表

最近更新日期：2024年5月8日（单位：%）

活期利率

银行名称 \ 利率	利率
华夏银行	0.200

定期利率

银行名称 \ 期限	三个月	半年	一年	两年	三年	五年
华夏银行	1.200	1.450	1.650	1.800	2.000	2.050

零存整取/整存零取/存本取息

银行名称 \ 期限	一年	三年	五年
华夏银行	1.200	1.400	1.450

通知存款

银行名称 \ 期限	1天	7天
华夏银行	0.250	0.800

全国银行间同业拆借中心发布贷款市场报价利率（LPR）

发布日期：2024年02月21日

期限品种	利率
1年期人民币贷款市场报价利率	3.45%
5年期以上人民币贷款市场报价利率	3.95%

• b) 建立模型并求解：

- A 君账户模型—— A_i 为账户余额： $A_i = (1 + r_1)A_{i-1}$, $i = 1, 2, \dots, 5$, 且 $A_0 = 1000000$, 计算 A 君未来 10 年的账户余额。

% 设定时间单位为两年，则取时间单位长度为 5

$n = 5$;

% 设定初始金额与利率

$A_0 = 1000000$;

$r_1 = 0.018$;

$r_{11} = r_1 * 2$;

% 初始化账户余额

$A = \text{zeros}(n+1, 1)$;

$A(1) = A_0$;

% 计算余额

for $i = 1:n$

$\text{interest} = A(i) * r_{11}$; % 计算利息

$A(i+1) = A(i) + \text{interest}$; % 计算余额

end

% 转换为每年的余额情况

$AA = \text{zeros}(n*2+1, 1)$;

$AA(1) = A(1)$;

```

for i = 1:n
    AA(i*2)=A(i+1);
    AA(i*2+1)=A(i+1);
end

```

- 呈现 A 君未来 10 年的账户余额：

```

% 显示
years=0:1:10;
A_str = arrayfun(@(x) sprintf('%.0f', x), AA, 'UniformOutput', false);
T = table(years', A_str, 'VariableNames', {'年份', 'A 君账户余额'});
disp(T);

```

年份	A 君账户余额
0	{'1000000'}
1	{'1036000'}
2	{'1036000'}
3	{'1073296'}
4	{'1073296'}
5	{'1111935'}
6	{'1111935'}
7	{'1151964'}
8	{'1151964'}
9	{'1193435'}
10	{'1193435'}

- 存储银行最后的支出：

```

PA=A(6);
save('P_fund.mat', 'PA', '-append')

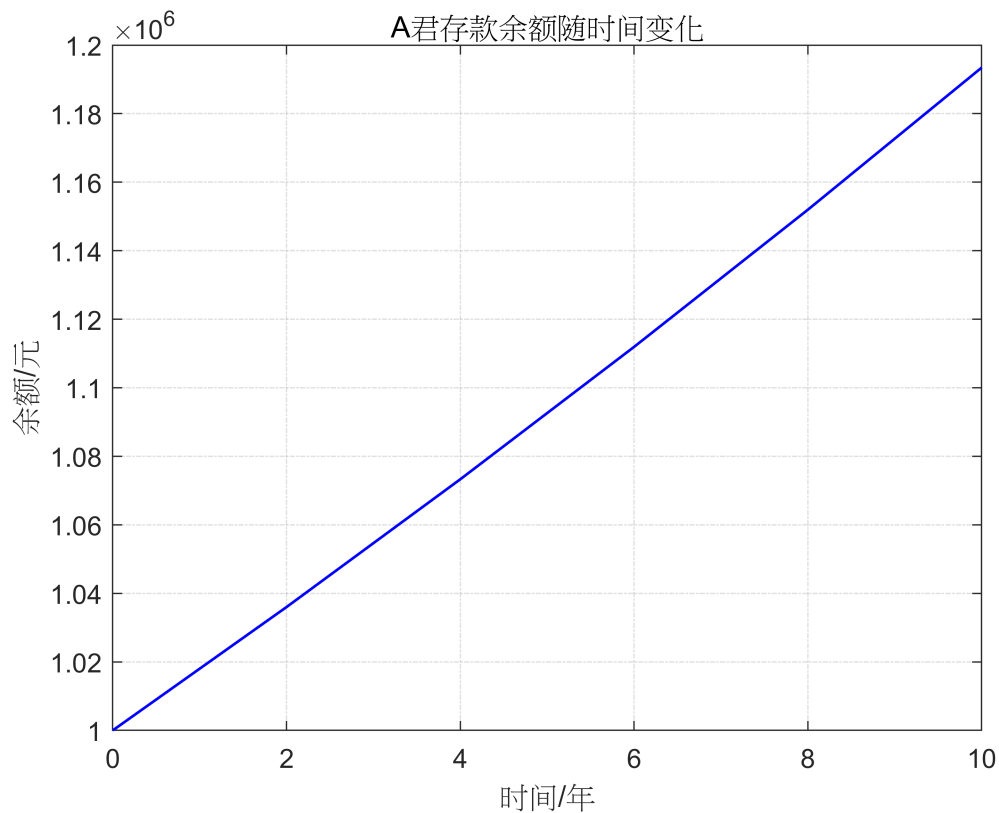
```

- 绘制图像呈现 A 君账户余额：

```

% 得到时间横轴
x=0:1:n;
x=x*2;
% 绘制余额随时间的变化
fig=plot(x, A, 'b-');
% 设置线宽
fig(1).LineWidth =1.0;
% 打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
% 设置坐标轴和标题
xlabel('时间/年');
ylabel('余额/元');
title('A 君存款余额随时间变化');

```



- B 君账户模型（假设等额本息法）—— B_k 为账户余额： $B_k = (1 + r_2)B_{k-1} - m, k = 1, 2, \dots, 120$ ，且 $B_0 = 1000000$ ，计算 B 君未来 10 年的账户余额。

```
clear;
% 设定时间单位为一月，则取时间单位长度为 120
n = 10 * 12 ;
% 设定初始金额与利率
B0 = 1e6;
r2 = 0.0395;
r22 = r2 / 12;
% 计算每月还款额
m = r22*B0*(1+r22)^n / ((1+r22)^n - 1);
% 初始化余额数组
B = zeros(n+1, 1);
B(1) = B0;
% 计算余额
for i = 2:n+1
    B(i) = B(i-1)*(1+r22) - m; % 计算本月的余额
end
% 转换为逐年余额
x=0:1:10;
balance_year=zeros(11, 1);
for i=1:11
    balance_year(i)=B(12*(i-1)+1);
```

end

- 呈现 B 君未来 10 年的账户余额：

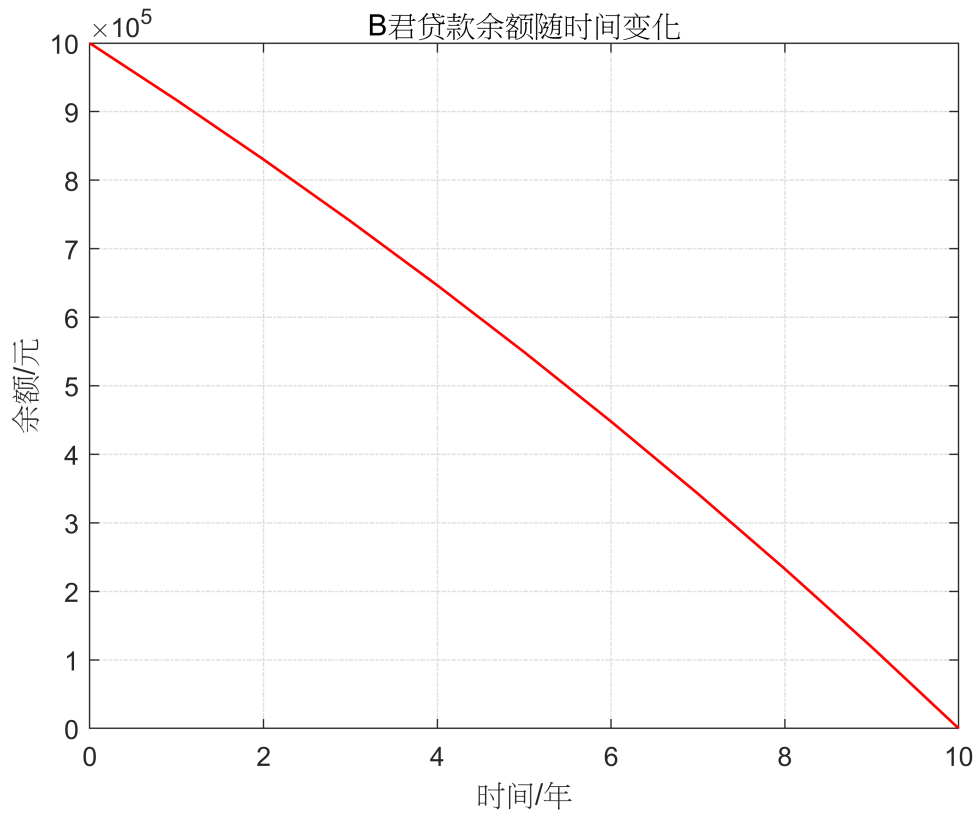
% 显示

```
years=0:1:10;  
B_str = arrayfun(@(x) sprintf('%.0f', x), balance_year, 'UniformOutput', false);  
T = table(years', B_str, 'VariableNames', {'年份', 'B 君贷款余额'});  
disp(T);
```

年份	B 君贷款余额
0	{'1000000'}
1	{'916795' }
2	{'830244' }
3	{'740211' }
4	{'646556' }
5	{'549135' }
6	{'447795' }
7	{'342379' }
8	{'232722' }
9	{'118655' }
10	{'0' }

- 绘制图像呈现 B 君贷款余额：

```
fig=plot(x, balance_year, 'r-');  
% 设置线宽  
fig(1).LineWidth =1.0;  
% 打开网格  
grid on;  
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');  
% 设置坐标轴和标题  
xlabel('时间/年');  
ylabel('余额/元');  
title('B 君贷款余额随时间变化');
```



(2)请建立数学模型，分别描述 A-E5 种情况下 B 君在该银行所设专项账户逐期余额情况。

- a) 方式 A：到期一次性还本息，名义利率等于实际利率，客户在银行的定期存款就是这种方式。

```
clear;
% 设置初值，计算总利息
B0 = 1e6;
r22 = 0.0395 * 10;
interest=B0*r22;
% 计算账户余额
balance_year = zeros(11, 1)+B0;
balance_year(end)=0;
```

- 显示 B 君逐期余额情况：

```
years=0:1:10;
balance_str = arrayfun(@(x) sprintf('%0.0f', x), balance_year, 'UniformOutput', false);
T = table(years', balance_str, 'VariableNames', {'年份', 'B 君贷款余额 (方式 A)'});
disp(T);
```

年份	B 君贷款余额 (方式 A)
0	{'1000000'}
1	{'1000000'}
2	{'1000000'}

```

3      {'1000000'}
4      {'1000000'}
5      {'1000000'}
6      {'1000000'}
7      {'1000000'}
8      {'1000000'}
9      {'1000000'}
10     {'0'       }

```

- 存储银行最后的收入：

```

P1=balance_year(1)+interest;
save('P_fund.mat','P1','-append');

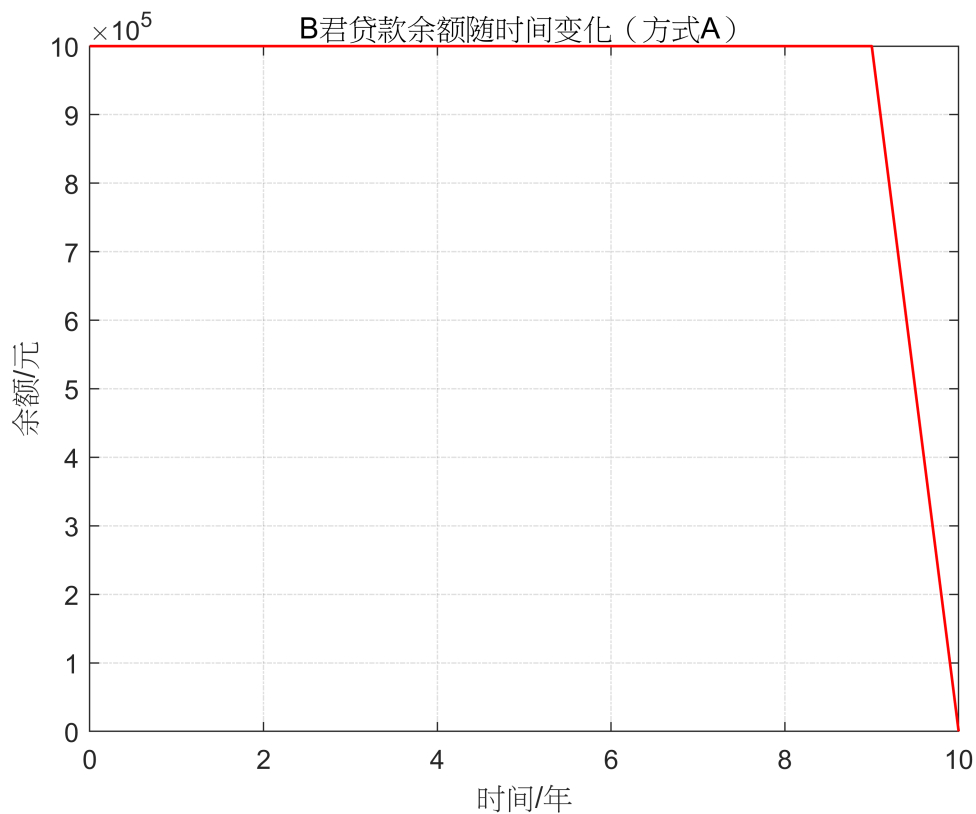
```

- 绘制图像呈现 B 君贷款余额：

```

fig=plot(0:1:10, balance_year,'r-');
% 设置线宽
fig(1).LineWidth =1.0;
% 打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
% 设置坐标轴和标题
xlabel('时间/年');
ylabel('余额/元');
title('B 君贷款余额随时间变化（方式 A）');

```



- **方式 B**：先按月还息，到期还本，实际利率接近 A 的利率，是信用贷款常用的做法。

```
clear;
% 设定时间单位为两年为一月，则取时间单位长度为 120
n=10*12;
% 设置初值，计算月利息
B0 = 1e6;
r2=0.0395;
r22 = r2 /12;
interest = zeros(n, 1);
% 计算账户余额
balance = zeros(n+1, 1)+B0;
balance(end)=0;
```

- 显示 B 君逐期余额情况：

```
months=0:1:n;
balance_str = arrayfun(@(x) sprintf('%.0f', x), balance, 'UniformOutput', false);
T = table(months', balance_str, 'VariableNames', {'月份', 'B 君贷款余额 (方式 B)'});
disp(T);
```

月份	B 君贷款余额 (方式 B)
0	{'1000000'}
1	{'1000000'}
2	{'1000000'}
3	{'1000000'}
4	{'1000000'}
5	{'1000000'}
6	{'1000000'}
7	{'1000000'}
8	{'1000000'}
9	{'1000000'}
10	{'1000000'}
11	{'1000000'}
12	{'1000000'}
13	{'1000000'}
14	{'1000000'}
15	{'1000000'}
16	{'1000000'}
17	{'1000000'}
18	{'1000000'}
19	{'1000000'}
20	{'1000000'}
21	{'1000000'}
22	{'1000000'}
23	{'1000000'}
24	{'1000000'}
25	{'1000000'}
26	{'1000000'}
27	{'1000000'}
28	{'1000000'}
29	{'1000000'}
30	{'1000000'}
31	{'1000000'}

32	{'1000000'}
33	{'1000000'}
34	{'1000000'}
35	{'1000000'}
36	{'1000000'}
37	{'1000000'}
38	{'1000000'}
39	{'1000000'}
40	{'1000000'}
41	{'1000000'}
42	{'1000000'}
43	{'1000000'}
44	{'1000000'}
45	{'1000000'}
46	{'1000000'}
47	{'1000000'}
48	{'1000000'}
49	{'1000000'}
50	{'1000000'}
51	{'1000000'}
52	{'1000000'}
53	{'1000000'}
54	{'1000000'}
55	{'1000000'}
56	{'1000000'}
57	{'1000000'}
58	{'1000000'}
59	{'1000000'}
60	{'1000000'}
61	{'1000000'}
62	{'1000000'}
63	{'1000000'}
64	{'1000000'}
65	{'1000000'}
66	{'1000000'}
67	{'1000000'}
68	{'1000000'}
69	{'1000000'}
70	{'1000000'}
71	{'1000000'}
72	{'1000000'}
73	{'1000000'}
74	{'1000000'}
75	{'1000000'}
76	{'1000000'}
77	{'1000000'}
78	{'1000000'}
79	{'1000000'}
80	{'1000000'}
81	{'1000000'}
82	{'1000000'}
83	{'1000000'}
84	{'1000000'}
85	{'1000000'}
86	{'1000000'}
87	{'1000000'}
88	{'1000000'}
89	{'1000000'}
90	{'1000000'}
91	{'1000000'}
92	{'1000000'}
93	{'1000000'}
94	{'1000000'}
95	{'1000000'}

```

96      {'1000000'}
97      {'1000000'}
98      {'1000000'}
99      {'1000000'}
100     {'1000000'}
101     {'1000000'}
102     {'1000000'}
103     {'1000000'}
104     {'1000000'}
105     {'1000000'}
106     {'1000000'}
107     {'1000000'}
108     {'1000000'}
109     {'1000000'}
110     {'1000000'}
111     {'1000000'}
112     {'1000000'}
113     {'1000000'}
114     {'1000000'}
115     {'1000000'}
116     {'1000000'}
117     {'1000000'}
118     {'1000000'}
119     {'1000000'}
120     {'0'       }

```

- 存储银行最后的收入：

```

P2=0;
for i=1:n
    interest(i)=B0*r22;
    P2=P2+interest(i);
end
P2=balance(1)+P2;
save('P_fund.mat','P2','-append');

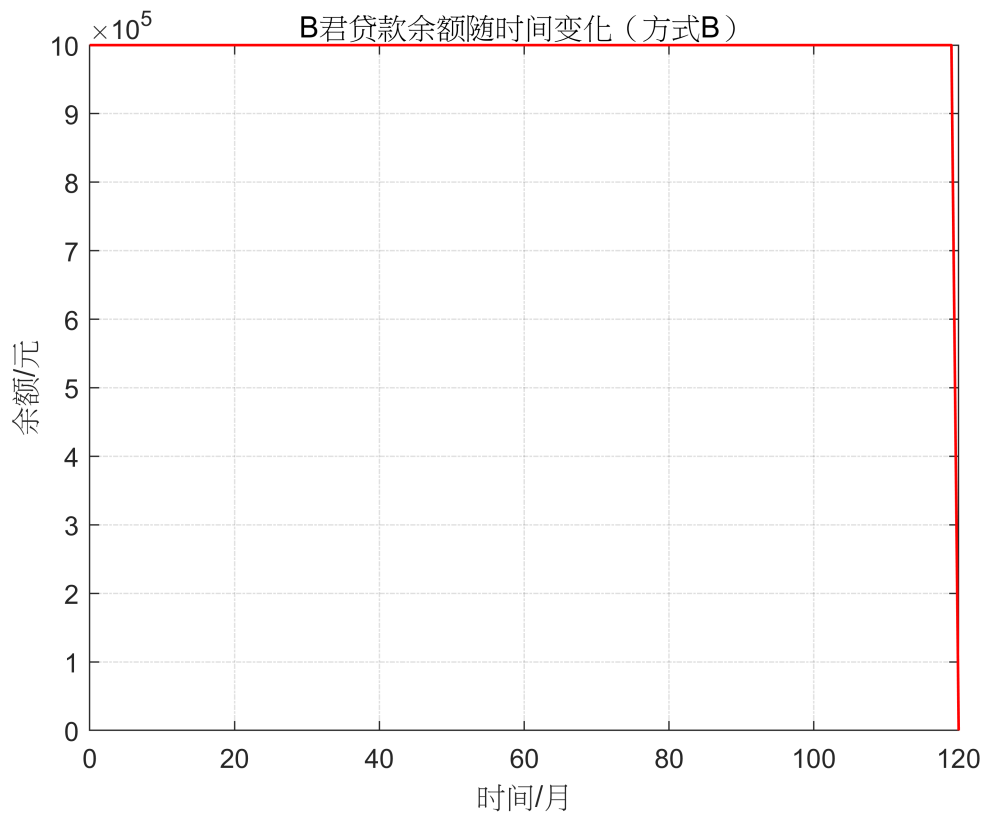
```

- 绘制图像呈现 B 君贷款余额：

```

fig=plot(months, balance,'r-');
% 设置线宽
fig(1).LineWidth =1.0;
% 打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
% 设置坐标轴和标题
xlabel('时间/月');
ylabel('余额/元');
title('B 君贷款余额随时间变化（方式 B）');

```



- c) 方式 C : 等额本息法, 按月等额度还本金和利息, 银行贷款的最常见方式。

```
clear;
% 设定时间单位为两年为一月, 则取时间单位长度为 120
n = 10 * 12 ;
% 设定初始金额与利率
B0 = 1e6;
r2 = 0.0395;
r22 = r2 / 12;
% 计算每月还款额
m = r22*B0*(1+r22)^n / ((1+r22)^n - 1);
% 初始化余额数组
balance = zeros(n+1, 1);
balance(1) = B0;
% 计算余额
for i = 2:n+1
    balance(i) = balance(i-1)*(1+r22) - m; % 计算本月的余额
end
```

- 显示 B 君逐期余额情况 :

```
months=0:1:n;
balance_str = arrayfun(@(x) sprintf('%.0f', x), balance, 'UniformOutput', false);
T = table(months', balance_str, 'VariableNames', {'月份', 'B 君贷款余额 (方式 C) '});
```

disp(T);

月份 B 君贷款余额 (方式 C)

0	{ '1000000' }
1	{ '993191' }
2	{ '986359' }
3	{ '979505' }
4	{ '972629' }
5	{ '965730' }
6	{ '958808' }
7	{ '951863' }
8	{ '944895' }
9	{ '937905' }
10	{ '930891' }
11	{ '923855' }
12	{ '916795' }
13	{ '909712' }
14	{ '902606' }
15	{ '895476' }
16	{ '888323' }
17	{ '881146' }
18	{ '873946' }
19	{ '866722' }
20	{ '859474' }
21	{ '852202' }
22	{ '844907' }
23	{ '837587' }
24	{ '830244' }
25	{ '822876' }
26	{ '815484' }
27	{ '808067' }
28	{ '800626' }
29	{ '793161' }
30	{ '785671' }
31	{ '778156' }
32	{ '770617' }
33	{ '763053' }
34	{ '755464' }
35	{ '747850' }
36	{ '740211' }
37	{ '732546' }
38	{ '724857' }
39	{ '717142' }
40	{ '709402' }
41	{ '701636' }
42	{ '693845' }
43	{ '686028' }
44	{ '678186' }
45	{ '670317' }
46	{ '662423' }
47	{ '654503' }
48	{ '646556' }
49	{ '638584' }
50	{ '630585' }
51	{ '622560' }
52	{ '614508' }
53	{ '606430' }
54	{ '598326' }
55	{ '590194' }
56	{ '582036' }
57	{ '573852' }
58	{ '565640' }

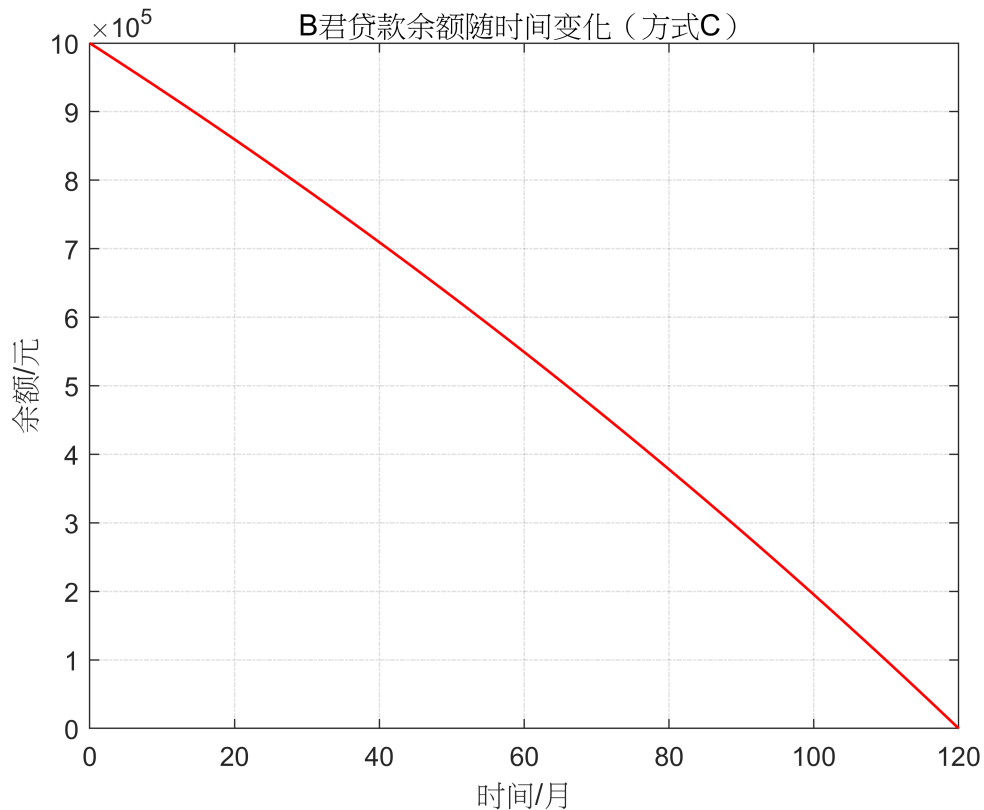
59	{ '557401' }
60	{ '549135' }
61	{ '540842' }
62	{ '532521' }
63	{ '524173' }
64	{ '515798' }
65	{ '507395' }
66	{ '498964' }
67	{ '490506' }
68	{ '482020' }
69	{ '473506' }
70	{ '464964' }
71	{ '456393' }
72	{ '447795' }
73	{ '439168' }
74	{ '430513' }
75	{ '421829' }
76	{ '413117' }
77	{ '404376' }
78	{ '395606' }
79	{ '386808' }
80	{ '377980' }
81	{ '369124' }
82	{ '360238' }
83	{ '351323' }
84	{ '342379' }
85	{ '333405' }
86	{ '324402' }
87	{ '315369' }
88	{ '306306' }
89	{ '297213' }
90	{ '288091' }
91	{ '278938' }
92	{ '269756' }
93	{ '260543' }
94	{ '251300' }
95	{ '242026' }
96	{ '232722' }
97	{ '223388' }
98	{ '214022' }
99	{ '204626' }
100	{ '195199' }
101	{ '185740' }
102	{ '176251' }
103	{ '166730' }
104	{ '157178' }
105	{ '147595' }
106	{ '137980' }
107	{ '128334' }
108	{ '118655' }
109	{ '108945' }
110	{ '99203' }
111	{ '89429' }
112	{ '79622' }
113	{ '69784' }
114	{ '59912' }
115	{ '50009' }
116	{ '40073' }
117	{ '30104' }
118	{ '20102' }
119	{ '10068' }
120	{ '0' }

- 存储银行最后的收入：

```
P3=m*n;  
save('P_fund.mat','P3','-append');
```

- 绘制图像呈现 B 君贷款余额：

```
fig=plot(months, balance,'r-');  
% 设置线宽  
fig(1).LineWidth =1.0;  
% 打开网格  
grid on;  
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');  
% 设置坐标轴和标题  
xlabel('时间/月');  
ylabel('余额/元');  
title('B 君贷款余额随时间变化（方式 C）');
```



- **d) 方式 D：等额本金法**，按月等还本金，并还相应利息(利息随本金减少而减少)，银行房贷的次常见方式。

```
clear;  
% 设定时间单位为两年为一月，则取时间单位长度为 120
```

```

n=10*12;
% 设置初值, 计算月利率
B0 = 1e6;
r2=0.0395;
r22 = r2 / 12;
% 每月还款额
m = B0 / n;
% 初始化余额数组
balance = zeros(n+1, 1);
interest = zeros(n, 1);
P4=0;
balance(1) = B0;
% 计算每个月的余额
for i = 2:n+1
    balance(i) = balance(i-1) - m; % 计算余额
    interest(i-1) = balance(i-1)*r22;% 计算利息
    P4=P4+interest(i-1);% 计算利息和
end

```

- 显示 B 君逐期余额情况：

```

months=0:1:n;
balance_str = arrayfun(@(x) sprintf('%.0f', x), balance, 'UniformOutput', false);
T = table(months, balance_str, 'VariableNames', {'月份', 'B 君贷款余额 (方式 D)'});
disp(T);

```

月份	B 君贷款余额 (方式 D)
0	{ '1000000' }
1	{ '991667' }
2	{ '983333' }
3	{ '975000' }
4	{ '966667' }
5	{ '958333' }
6	{ '950000' }
7	{ '941667' }
8	{ '933333' }
9	{ '925000' }
10	{ '916667' }
11	{ '908333' }
12	{ '900000' }
13	{ '891667' }
14	{ '883333' }
15	{ '875000' }
16	{ '866667' }
17	{ '858333' }
18	{ '850000' }
19	{ '841667' }
20	{ '833333' }
21	{ '825000' }
22	{ '816667' }
23	{ '808333' }
24	{ '800000' }
25	{ '791667' }
26	{ '783333' }

27	{ '775000' }
28	{ '766667' }
29	{ '758333' }
30	{ '750000' }
31	{ '741667' }
32	{ '733333' }
33	{ '725000' }
34	{ '716667' }
35	{ '708333' }
36	{ '700000' }
37	{ '691667' }
38	{ '683333' }
39	{ '675000' }
40	{ '666667' }
41	{ '658333' }
42	{ '650000' }
43	{ '641667' }
44	{ '633333' }
45	{ '625000' }
46	{ '616667' }
47	{ '608333' }
48	{ '600000' }
49	{ '591667' }
50	{ '583333' }
51	{ '575000' }
52	{ '566667' }
53	{ '558333' }
54	{ '550000' }
55	{ '541667' }
56	{ '533333' }
57	{ '525000' }
58	{ '516667' }
59	{ '508333' }
60	{ '500000' }
61	{ '491667' }
62	{ '483333' }
63	{ '475000' }
64	{ '466667' }
65	{ '458333' }
66	{ '450000' }
67	{ '441667' }
68	{ '433333' }
69	{ '425000' }
70	{ '416667' }
71	{ '408333' }
72	{ '400000' }
73	{ '391667' }
74	{ '383333' }
75	{ '375000' }
76	{ '366667' }
77	{ '358333' }
78	{ '350000' }
79	{ '341667' }
80	{ '333333' }
81	{ '325000' }
82	{ '316667' }
83	{ '308333' }
84	{ '300000' }
85	{ '291667' }
86	{ '283333' }
87	{ '275000' }
88	{ '266667' }
89	{ '258333' }
90	{ '250000' }


```

91      {'241667' }
92      {'233333' }
93      {'225000' }
94      {'216667' }
95      {'208333' }
96      {'200000' }
97      {'191667' }
98      {'183333' }
99      {'175000' }
100     {'166667' }
101     {'158333' }
102     {'150000' }
103     {'141667' }
104     {'133333' }
105     {'125000' }
106     {'116667' }
107     {'108333' }
108     {'100000' }
109     {'91667'  }
110     {'83333'  }
111     {'75000'  }
112     {'66667'  }
113     {'58333'  }
114     {'50000'  }
115     {'41667'  }
116     {'33333'  }
117     {'25000'  }
118     {'16667'  }
119     {'8333'   }
120     {'-0'     }

```

- 存储银行最后的收入：

```

% 计算还款总额
P4=P4+B0;
save('P_fund.mat','P4','-append');

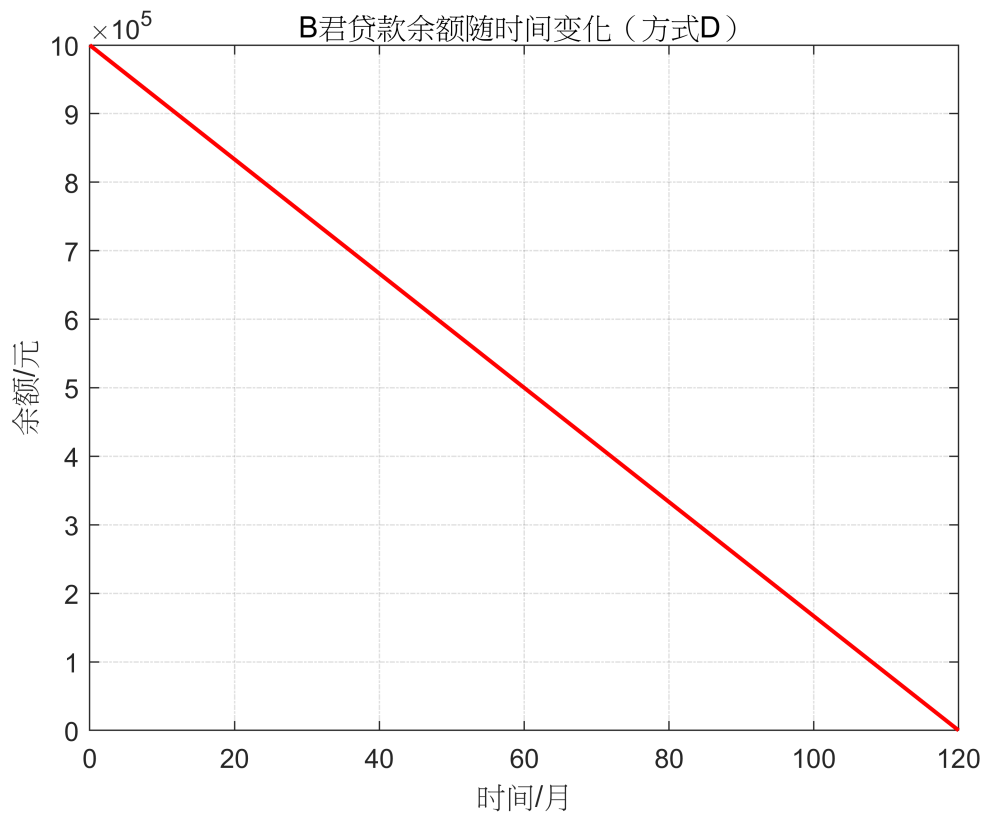
```

- 绘制图像呈现 B 君贷款余额：

```

fig=plot(months, balance,'r-');
%设置线宽
fig(1).LineWidth =1.5;
%打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
%设置坐标轴和标题
ylim([0,1e6]);
xlabel('时间/月');
ylabel('余额/元');
title('B 君贷款余额随时间变化（方式 D）');

```



- e) 方式 E：借贷时先一次性扣除利息，接下来每月还本金。

```
clear;
% 设置初值，计算总利息
B0 = 1e6;
r22 = 0.0395 * 10;
interest=B0*r22;%扣除利息
% 设定时间单位为两年为一月，则取时间单位长度为 120
n=10*12;
% 初始化余额数组
balance = zeros(n+1, 1);
P4=0;
balance(1) =B0-interest;
% 计算每月还款额
m=balance(1)/n;
% 计算每个月的余额
for i = 2:n+1
    balance(i) = balance(i-1) - m;
end
```

- 显示 B 君逐期余额情况：

```
months=0:1:n;
balance_str = arrayfun(@(x) sprintf('%.0f', x), balance, 'UniformOutput', false);
```

```
T = table(months', balance_str, 'VariableNames', {'月份', 'B 君贷款余额 (方式 E)'});
disp(T);
```

月份	B 君贷款余额 (方式 E)
0	{'605000'}
1	{'599958'}
2	{'594917'}
3	{'589875'}
4	{'584833'}
5	{'579792'}
6	{'574750'}
7	{'569708'}
8	{'564667'}
9	{'559625'}
10	{'554583'}
11	{'549542'}
12	{'544500'}
13	{'539458'}
14	{'534417'}
15	{'529375'}
16	{'524333'}
17	{'519292'}
18	{'514250'}
19	{'509208'}
20	{'504167'}
21	{'499125'}
22	{'494083'}
23	{'489042'}
24	{'484000'}
25	{'478958'}
26	{'473917'}
27	{'468875'}
28	{'463833'}
29	{'458792'}
30	{'453750'}
31	{'448708'}
32	{'443667'}
33	{'438625'}
34	{'433583'}
35	{'428542'}
36	{'423500'}
37	{'418458'}
38	{'413417'}
39	{'408375'}
40	{'403333'}
41	{'398292'}
42	{'393250'}
43	{'388208'}
44	{'383167'}
45	{'378125'}
46	{'373083'}
47	{'368042'}
48	{'363000'}
49	{'357958'}
50	{'352917'}
51	{'347875'}
52	{'342833'}
53	{'337792'}
54	{'332750'}
55	{'327708'}
56	{'322667'}
57	{'317625'}

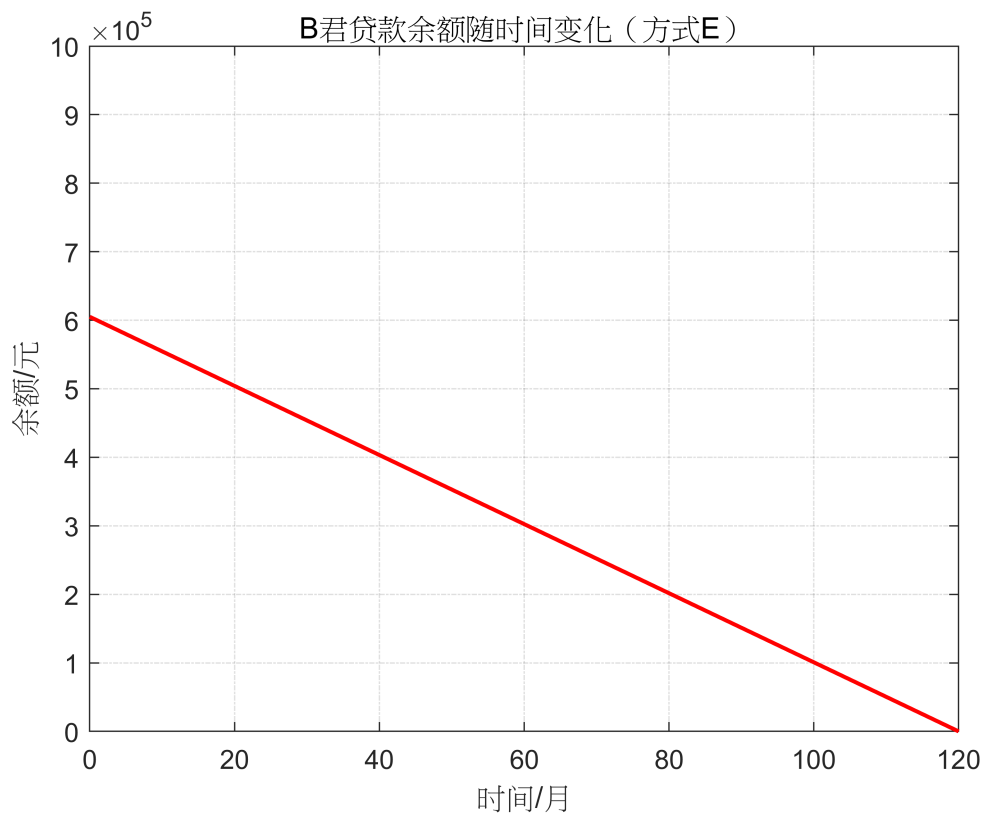
58	{'312583'}
59	{'307542'}
60	{'302500'}
61	{'297458'}
62	{'292417'}
63	{'287375'}
64	{'282333'}
65	{'277292'}
66	{'272250'}
67	{'267208'}
68	{'262167'}
69	{'257125'}
70	{'252083'}
71	{'247042'}
72	{'242000'}
73	{'236958'}
74	{'231917'}
75	{'226875'}
76	{'221833'}
77	{'216792'}
78	{'211750'}
79	{'206708'}
80	{'201667'}
81	{'196625'}
82	{'191583'}
83	{'186542'}
84	{'181500'}
85	{'176458'}
86	{'171417'}
87	{'166375'}
88	{'161333'}
89	{'156292'}
90	{'151250'}
91	{'146208'}
92	{'141167'}
93	{'136125'}
94	{'131083'}
95	{'126042'}
96	{'121000'}
97	{'115958'}
98	{'110917'}
99	{'105875'}
100	{'100833'}
101	{'95792' }
102	{'90750' }
103	{'85708' }
104	{'80667' }
105	{'75625' }
106	{'70583' }
107	{'65542' }
108	{'60500' }
109	{'55458' }
110	{'50417' }
111	{'45375' }
112	{'40333' }
113	{'35292' }
114	{'30250' }
115	{'25208' }
116	{'20167' }
117	{'15125' }
118	{'10083' }
119	{'5042' }
120	{'-0' }

- 存储银行最后的收入：

```
% 计算实际借贷额度
P5=balance(1);
save('P_fund.mat','P5','-append');
```

- 绘制图像呈现 B 君贷款余额：

```
fig=plot(months, balance,'r-');
%设置线宽
fig(1).LineWidth =1.5;
%打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
%设置坐标轴和标题
ylim([0,1e6]);
xlabel('时间/月');
ylabel('余额/元');
title('B 君贷款余额随时间变化（方式 E）');
```



(3)根据以上模型，比较 A-E 5 种情况下，存贷款的实际利率。

- 计算实际利率：（请将目录设置成 EXP4）

```

clear;
% 读取各方式下实际还款总额
load('P_fund.mat');
B0=1e6;
R=zeros(5,1);
R(1)=((P1-B0)/B0)/10;% 计算方式 A 的实际利率
R(2)=((P2-B0)/B0)/10;% 计算方式 B 的实际利率
R(3)=((P3-B0)/B0)/10;% 计算方式 C 的实际利率
R(4)=((P4-B0)/B0)/10;% 计算方式 D 的实际利率
R(5)=((B0-P5)/P5)/10;% 计算方式 E 的实际利率
% 显示
R_str = arrayfun(@(x) sprintf('%.4f%%', x), R*100, 'UniformOutput', false);
T = table(['方式 A';'方式 B';'方式 C';'方式 D';'方式 E'], R_str, 'VariableNames', {'信贷方式', '实际利率'});
disp(T);

```

信贷方式	实际利率
方式 A	{'3.9500%'}
方式 B	{'3.9500%'}
方式 C	{'2.1209%'}
方式 D	{'1.9915%'}
方式 E	{'6.5289%'}

- 经过比较可得实际利率：方式 E>方式 A（名义利率）=方式 B>方式 C>方式 D。

(4)讨论银行盈亏情况。

- 计算银行利润：（请将目录设置成 EXP4）

```

clear;
% 读取各方式下银行的收入和支出
load('P_fund.mat');
B0=1e6;
profit=zeros(5,1);
profit(1)=P1-PA;% 计算方式 A 的银行利润
profit(2)=P2-PA;% 计算方式 B 的银行利润
profit(3)=P3-PA;% 计算方式 C 的银行利润
profit(4)=P4-PA;% 计算方式 D 的银行利润
profit(5)=2*B0-P5-PA;% 计算方式 E 的银行利润
% 不同方式的盈亏
profit_str = arrayfun(@(x) sprintf('%.0f', x), profit, 'UniformOutput', false);
T = table(['方式 A';'方式 B';'方式 C';'方式 D';'方式 E'], profit_str, 'VariableNames', {'信贷方式', '银行利润'});
disp(T);

```

信贷方式	银行利润
方式 A	{'201565'}
方式 B	{'201565'}
方式 C	{'18657' }

方式 D {'5711' }
方式 E {'201565'}

- 经过比较可得银行利润：方式 A=方式 B=方式 E>方式 C>方式 D，且 5 种方式下银行均不会出现亏损。

(5) 如果利率允许波动(如 LPR)，做合理假设，重新考虑以上问题。

- 假设利率波动为正态随机波动，则信贷方式中方式 C 无法执行，方式 A,E 结果一样，只有方式 B,D 会有所不同。现计算波动的利率：

```
clear;  
% 设定时间单位为两年为一月，则取时间单位长度为 120  
n=10*12;  
mu = 0.0395;  
sigma = 0.01;  
% 生成标准正态分布的随机波动  
standard_normal=zeros(n,1);  
r2 = zeros(n, 1);  
for i=1:n  
    while(1)  
        standard_normal(i) = randn;  
        r2(i) = mu + sigma * standard_normal(i);  
        if(r2(i)>0&&r2(i)<mu*2)  
            break;  
        end  
    end  
end  
% 储存正态随机波动的年利率  
save('RandomChange_r2.mat','r2');
```

- **方式 B：先按月还息，到期还本**，实际利率接近 A 的利率，是信用贷款常用的做法。

```
clear;  
% 设定时间单位为两年为一月，则取时间单位长度为 120  
n=10*12;  
% 读取随机波动的利率  
load('RandomChange_r2.mat');  
% 设置初值，计算月利息  
B0 = 1e6;  
r22 = r2 ./12;  
interest = zeros(n, 1);  
% 计算账户余额  
balance = zeros(n+1, 1)+B0;  
balance(end)=0;  
% 显示  
months=0:1:n;  
balance_str = arrayfun(@(x) sprintf('%.0f', x), balance, 'UniformOutput', false);
```

```
T = table(months', balance_str, 'VariableNames', {'年份', 'B 君贷款余额'});
disp(T);
```

年份	B 君贷款余额
0	{'1000000'}
1	{'1000000'}
2	{'1000000'}
3	{'1000000'}
4	{'1000000'}
5	{'1000000'}
6	{'1000000'}
7	{'1000000'}
8	{'1000000'}
9	{'1000000'}
10	{'1000000'}
11	{'1000000'}
12	{'1000000'}
13	{'1000000'}
14	{'1000000'}
15	{'1000000'}
16	{'1000000'}
17	{'1000000'}
18	{'1000000'}
19	{'1000000'}
20	{'1000000'}
21	{'1000000'}
22	{'1000000'}
23	{'1000000'}
24	{'1000000'}
25	{'1000000'}
26	{'1000000'}
27	{'1000000'}
28	{'1000000'}
29	{'1000000'}
30	{'1000000'}
31	{'1000000'}
32	{'1000000'}
33	{'1000000'}
34	{'1000000'}
35	{'1000000'}
36	{'1000000'}
37	{'1000000'}
38	{'1000000'}
39	{'1000000'}
40	{'1000000'}
41	{'1000000'}
42	{'1000000'}
43	{'1000000'}
44	{'1000000'}
45	{'1000000'}
46	{'1000000'}
47	{'1000000'}
48	{'1000000'}
49	{'1000000'}
50	{'1000000'}
51	{'1000000'}
52	{'1000000'}
53	{'1000000'}
54	{'1000000'}
55	{'1000000'}
56	{'1000000'}
57	{'1000000'}

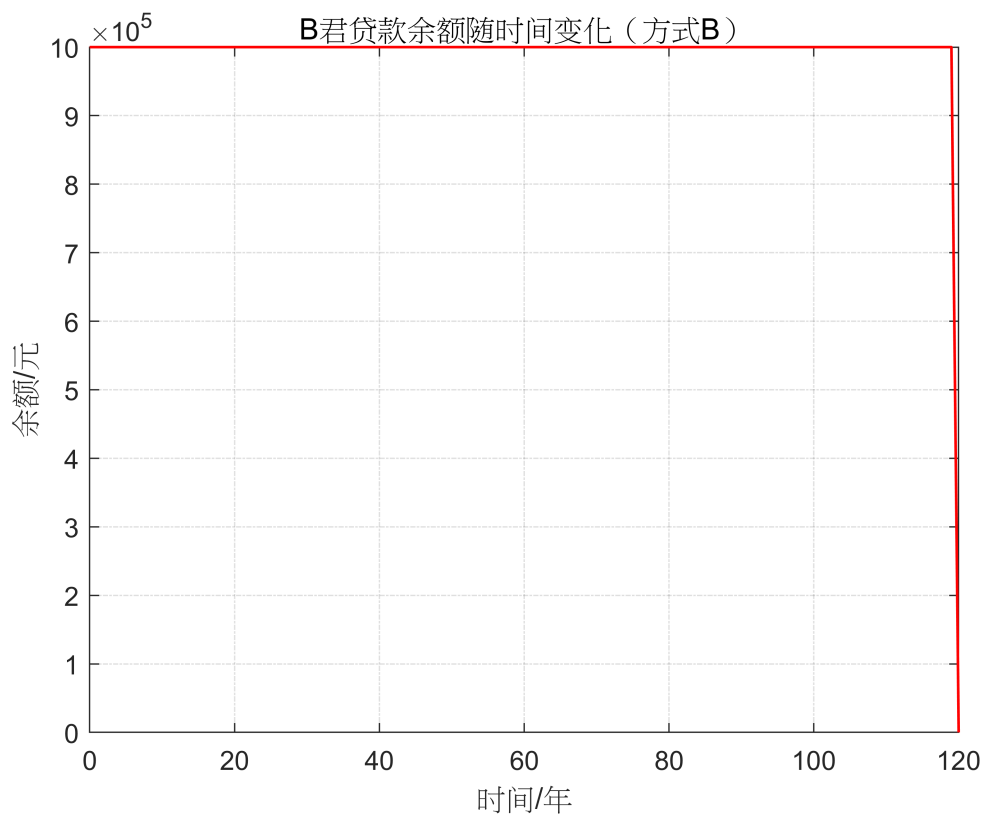

```
58      {'1000000'}
59      {'1000000'}
60      {'1000000'}
61      {'1000000'}
62      {'1000000'}
63      {'1000000'}
64      {'1000000'}
65      {'1000000'}
66      {'1000000'}
67      {'1000000'}
68      {'1000000'}
69      {'1000000'}
70      {'1000000'}
71      {'1000000'}
72      {'1000000'}
73      {'1000000'}
74      {'1000000'}
75      {'1000000'}
76      {'1000000'}
77      {'1000000'}
78      {'1000000'}
79      {'1000000'}
80      {'1000000'}
81      {'1000000'}
82      {'1000000'}
83      {'1000000'}
84      {'1000000'}
85      {'1000000'}
86      {'1000000'}
87      {'1000000'}
88      {'1000000'}
89      {'1000000'}
90      {'1000000'}
91      {'1000000'}
92      {'1000000'}
93      {'1000000'}
94      {'1000000'}
95      {'1000000'}
96      {'1000000'}
97      {'1000000'}
98      {'1000000'}
99      {'1000000'}
100     {'1000000'}
101     {'1000000'}
102     {'1000000'}
103     {'1000000'}
104     {'1000000'}
105     {'1000000'}
106     {'1000000'}
107     {'1000000'}
108     {'1000000'}
109     {'1000000'}
110     {'1000000'}
111     {'1000000'}
112     {'1000000'}
113     {'1000000'}
114     {'1000000'}
115     {'1000000'}
116     {'1000000'}
117     {'1000000'}
118     {'1000000'}
119     {'1000000'}
120     {'0'      }
```

- 存储银行最后的收入：

```
p2=0;
for i=1:n
    interest(i)=B0*r22(i);
    p2=p2+interest(i);
end
p2=balance(1)+p2;
save('P_fund.mat','p2','-append');
```

- 绘制图像呈现 B 君贷款余额：

```
fig=plot(months, balance,'r-');
% 设置线宽
fig(1).LineWidth =1.0;
% 打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
% 设置坐标轴和标题
xlabel('时间/年');
ylabel('余额/元');
title('B 君贷款余额随时间变化（方式 B）');
```



- 方式 D：等额本金法，按月等还本金，并还相应利息(利息随本金减少而减少)，银行房贷的次常见方式。

```
clear;
% 设定时间单位为两年为一月，则取时间单位长度为 120
n=10*12;
% 读取随机波动的利率
load('RandomChange_r2.mat');
% 设置初值，计算月利率
B0 = 1e6;
r22 = r2 ./ 12;
% 每月还款额
m = B0 / n;
% 初始化余额数组
balance = zeros(n+1, 1);
interest = zeros(n, 1);
p4=0;
balance(1) = B0;
% 计算每个月的余额
for i = 1:n
    balance(i+1) = balance(i) - m; % 计算余额
    interest(i) = balance(i)*r22(i); % 计算利息
    p4=p4+interest(i); % 计算利息和
end
% 显示
months=0:1:n;
balance_str = arrayfun(@(x) sprintf('%.0f', x), balance, 'UniformOutput', false);
T = table(months, balance_str, 'VariableNames', {'月份', 'B 君贷款余额'});
disp(T);
```

月份	B 君贷款余额
0	{'1000000'}
1	{'991667' }
2	{'983333' }
3	{'975000' }
4	{'966667' }
5	{'958333' }
6	{'950000' }
7	{'941667' }
8	{'933333' }
9	{'925000' }
10	{'916667' }
11	{'908333' }
12	{'900000' }
13	{'891667' }
14	{'883333' }
15	{'875000' }
16	{'866667' }
17	{'858333' }
18	{'850000' }
19	{'841667' }
20	{'833333' }
21	{'825000' }
22	{'816667' }

23 {'808333' }
24 {'800000' }
25 {'791667' }
26 {'783333' }
27 {'775000' }
28 {'766667' }
29 {'758333' }
30 {'750000' }
31 {'741667' }
32 {'733333' }
33 {'725000' }
34 {'716667' }
35 {'708333' }
36 {'700000' }
37 {'691667' }
38 {'683333' }
39 {'675000' }
40 {'666667' }
41 {'658333' }
42 {'650000' }
43 {'641667' }
44 {'633333' }
45 {'625000' }
46 {'616667' }
47 {'608333' }
48 {'600000' }
49 {'591667' }
50 {'583333' }
51 {'575000' }
52 {'566667' }
53 {'558333' }
54 {'550000' }
55 {'541667' }
56 {'533333' }
57 {'525000' }
58 {'516667' }
59 {'508333' }
60 {'500000' }
61 {'491667' }
62 {'483333' }
63 {'475000' }
64 {'466667' }
65 {'458333' }
66 {'450000' }
67 {'441667' }
68 {'433333' }
69 {'425000' }
70 {'416667' }
71 {'408333' }
72 {'400000' }
73 {'391667' }
74 {'383333' }
75 {'375000' }
76 {'366667' }
77 {'358333' }
78 {'350000' }
79 {'341667' }
80 {'333333' }
81 {'325000' }
82 {'316667' }
83 {'308333' }
84 {'300000' }
85 {'291667' }
86 {'283333' }

```

87      {'275000' }
88      {'266667' }
89      {'258333' }
90      {'250000' }
91      {'241667' }
92      {'233333' }
93      {'225000' }
94      {'216667' }
95      {'208333' }
96      {'200000' }
97      {'191667' }
98      {'183333' }
99      {'175000' }
100     {'166667' }
101     {'158333' }
102     {'150000' }
103     {'141667' }
104     {'133333' }
105     {'125000' }
106     {'116667' }
107     {'108333' }
108     {'100000' }
109     {'91667' }
110     {'83333' }
111     {'75000' }
112     {'66667' }
113     {'58333' }
114     {'50000' }
115     {'41667' }
116     {'33333' }
117     {'25000' }
118     {'16667' }
119     {'8333' }
120     {'-0' }

```

- 存储银行最后的收入：

```

% 计算还款总额
p4=p4+B0;
save('P_fund.mat','p4','-append');

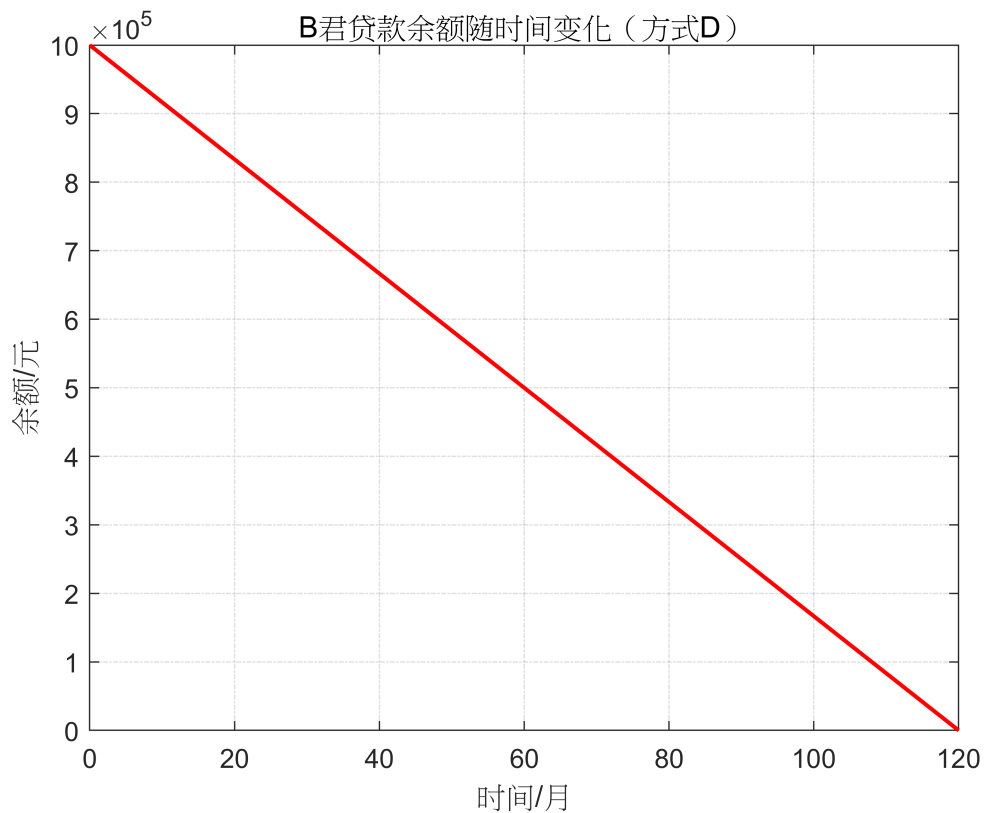
```

- 绘制图像呈现 B 君贷款余额：

```

fig=plot(months, balance,'r-');
%设置线宽
fig(1).LineWidth =1.5;
%打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
%设置坐标轴和标题
ylim([0,1e6]);
xlabel('时间/月');
ylabel('余额/元');
title('B 君贷款余额随时间变化（方式 D）');

```



- 根据以上模型，比较了 A,B,D,E 4 种情况下，存贷款的实际利率：（请将目录设置成 EXP4）

```
clear;
% 读取各方式下实际还款总额
load('P_fund.mat');
B0=1e6;
R=zeros(4,1);
R(1)=((P1-B0)/B0)/10;% 计算方式 A 的实际利率
R(2)=((p2-B0)/B0)/10;% 计算方式 B 的实际利率
R(3)=((p4-B0)/B0)/10;% 计算方式 D 的实际利率
R(4)=((B0-P5)/P5)/10;% 计算方式 E 的实际利率
% 显示
R_str = arrayfun(@(x) sprintf('%.4f%%', x), R*100, 'UniformOutput', false);
T = table(['方式 A';'方式 B';'方式 D';'方式 E'], R_str, 'VariableNames', {'信贷方式', '实际利率'});
disp(T);
```

信贷方式	实际利率
方式 A	{'3.9500%'}
方式 B	{'3.9221%'}
方式 D	{'2.0205%'}
方式 E	{'6.5289%'}

- 经过比较可得实际利率：方式 E>方式 A（名义利率）>方式 B>方式 D，而方式 B 与名义利率（0.0395）接近。
- 讨论银行盈亏：（请将目录设置成 EXP4）

```
clear;
% 读取各方式下银行的收入和支出
load('P_fund.mat');
profit=zeros(4,1);
B0=1e6;
profit(1)=P1-PA;% 计算方式 A 的银行利润
profit(2)=p2-PA;% 计算方式 B 的银行利润
profit(3)=p4-PA;% 计算方式 D 的银行利润
profit(4)=2*B0-P5-PA;% 计算方式 E 的银行利润
% 不同方式的盈亏
profit_str = arrayfun(@(x) sprintf('%.0f', x), profit, 'UniformOutput', false);
T = table(['方式 A';'方式 B';'方式 D';'方式 E'], profit_str, 'VariableNames', {'信贷方式', '银行利润'});
disp(T);
```

信贷方式	银行利润
方式 A	{ '201565' }
方式 B	{ '198773' }
方式 D	{ '8613' }
方式 E	{ '201565' }

- 经过比较可得银行利润：方式 A=方式 E>方式 D，而方式 B 与方式 A 接近；且 4 种方式下银行均（绝大多数）不会出现亏损。

2. 草场放牧

研究将鹿群放入草场后草和鹿两种群的相互作用。草的生长遵从 Logistic 规律，年固有增长率 0.8，最大密度为 3000(密度单位)，在草最茂盛时每只鹿每年可吃掉 1.6(密度单位)的草。若没有草，鹿群的年死亡率高达 0.9，而草的存在可使鹿的死亡得以补偿，在草最茂盛时补偿率为 1.5。作出一些简化假设，用差分方程模型描述草和鹿两种群数量的变化过程。就以下情况进行讨论：

(1) 比较将 100 只鹿放入密度为 1000 和密度为 3000 的草场两种情况。

(2) 适当改变参数，观察变化趋势。

(1) 比较将 100 只鹿放入密度为 1000 和密度为 3000 的草场两种情况。

- 符号说明：

a:草独立生存时的自然增长率；b: 鹿群掠取草的能力；c：鹿群独自存在时的死亡率；d: 草对鹿群的供养能力。并假设鹿食草能力与草的茂盛程度呈线性关系，草对鹿的补偿能力与草的茂盛程度呈线性关系，可以确定参数如下：

```

xm= 3000;%草的最大密度为 3000
a = 0.8;
bb= 1.6;
b = bb/xm;
c = 0.9;
dd= 1.5;
d = dd/xm;
T = 100; % 研究时间（年）

```

- 记草的密度为 x_k ; $x_m=3000$, 满足规律为 $x_{k+1} - x_k = x_k[a(1 - \frac{x_k}{x_m}) - by_k]$, $k = 0, 1, \dots$
- 记鹿群数量为 y_k ; $y_0=100$, 满足规律为 $y_{k+1} - y_k = y_k(-c + dx_k)$, $k = 0, 1, \dots$
- a) 将 100 只鹿放入密度为 1000 的草场：

```

%初始条件
x0 = 1000;
y0 = 100;
% 初始化每年的草场密度和鹿群数量
X = zeros(T+1, 1); % Grass population
Y = zeros(T+1, 1); % Deer population
X(1) = x0;
Y(1) = y0;

% 差分方程模型
for k = 1:T
    %草场增长密度
    dx = X(k) * (a * (1 - X(k)/xm) - b*Y(k));
    % 鹿群增长数量
    if X(k) > 0
        dy=Y(k)*(-c+d*X(k));
    else
        dy=Y(k)*(-c);
    end
    % 更新数值
    X(k+1) = X(k) + dx ;
    Y(k+1) = Y(k) + dy;
end

```

- b) 将 100 只鹿放入密度为 3000 的草场：

```

%初始条件
xx0 = 3000; % 草
yy0 = 100; % 鹿
% 初始化每年的草场密度和鹿群数量
XX = zeros(T+1, 1); % 草
YY = zeros(T+1, 1); % 鹿
XX(1) = xx0;
YY(1) = yy0;

```



```

% 差分方程模型
for k = 1:T
    %草场增长密度
    dxx = XX(k) * (a * (1 - XX(k)/xm) - b*YY(k));
    % 鹿群增长数量
    % Deer population change
    if XX(k) > 0
        dyy=YY(k)*(-c+d*XX(k));
    else
        dyy=YY(k)*(-c);
    end
    % 更新数值
    XX(k+1) = XX(k) + dxx ;
    YY(k+1) = YY(k) + dyy;
end

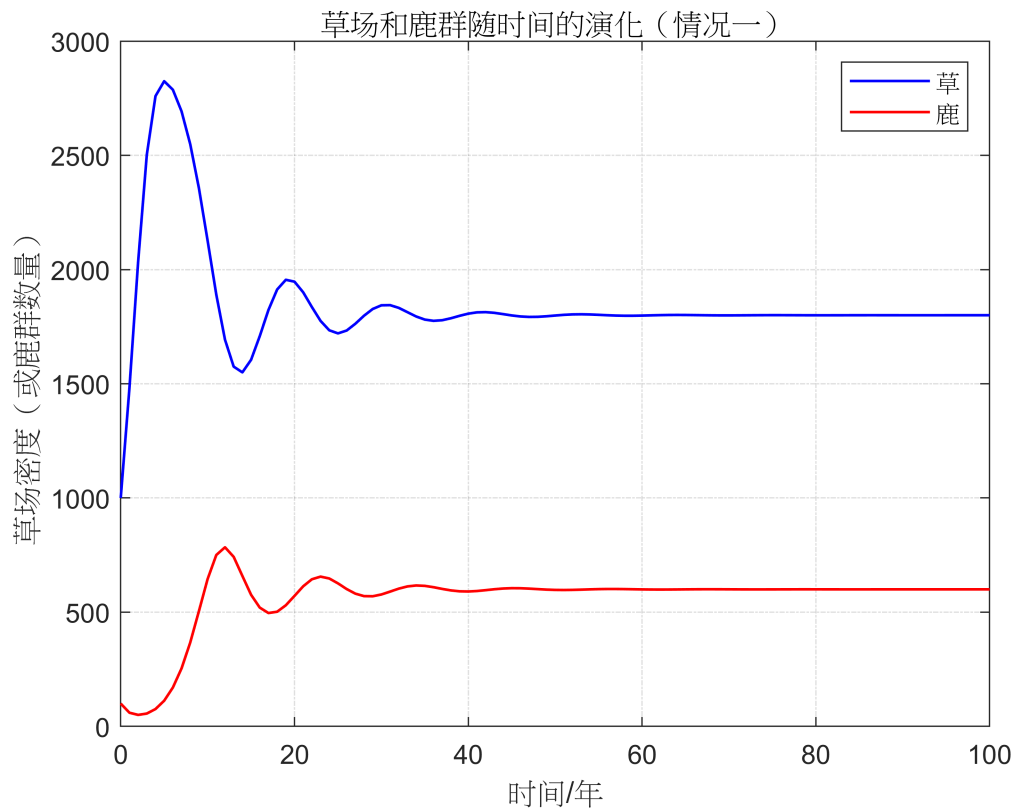
```

- c) 对比图像：
- 情况一：将 100 只鹿放入密度为 1000 的草场：

```

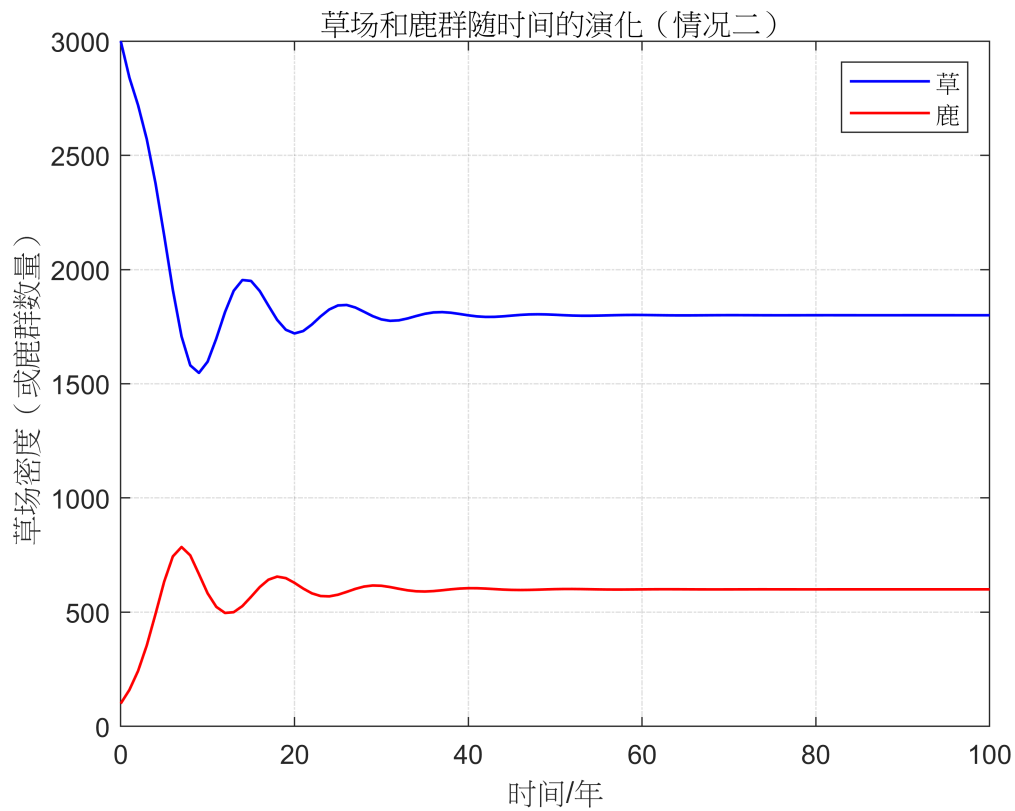
figure;
fig=plot(0:1:T, X, 'b', 0:1:T, Y, 'r');
%设置线宽
fig(1).LineWidth =1.0;fig(2).LineWidth =1.0;
%打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
%设置坐标轴和标题
xlabel('时间/年');
ylabel('草场密度（或鹿群数量）');
legend('草', '鹿');
title('草场和鹿群随时间的演化（情况一）');

```



• 情况二：将 100 只鹿放入密度为 3000 的草场：

```
figure;
fig=plot(0:1:T, XX, 'b', 0:1:T, YY, 'r');
%设置线宽
fig(1).LineWidth =1.0;fig(2).LineWidth =1.0;
%打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
%设置坐标轴和标题
xlabel('时间/年');
ylabel('草场密度（或鹿群数量）');
legend('草', '鹿');
title('草场和鹿群随时间的演化（情况二）');
```



(2)适当改变参数，观察变化趋势。

- 调整参数：

```
clear;
xm = 3000;
a=0.8;
bb = 1.6;
b=bb/xm;
c = 0.9;
dd = 1.5;
x0=2000;
d=dd/xm;
T = 100; % 研究时间（年）
```

- 仿真各参数下的运行结果：

```
%初始条件
y0 = 100;
% 初始化每年的草场密度和鹿群数量
X = zeros(T+1, 1); % Grass population
Y = zeros(T+1, 1); % Deer population
X(1) = x0;
Y(1) = y0;
```

```

% 差分方程模型
for k = 1:T
    %草场增长密度
    dx = X(k) * (a * (1 - X(k)/xm) - b*Y(k));
    % 鹿群增长数量
    if X(k) > 0
        dy=Y(k)*(-c+d*X(k));
    else
        dy=Y(k)*(-c);
    end
    % 更新数值
    X(k+1) = X(k) + dx ;
    Y(k+1) = Y(k) + dy;
end

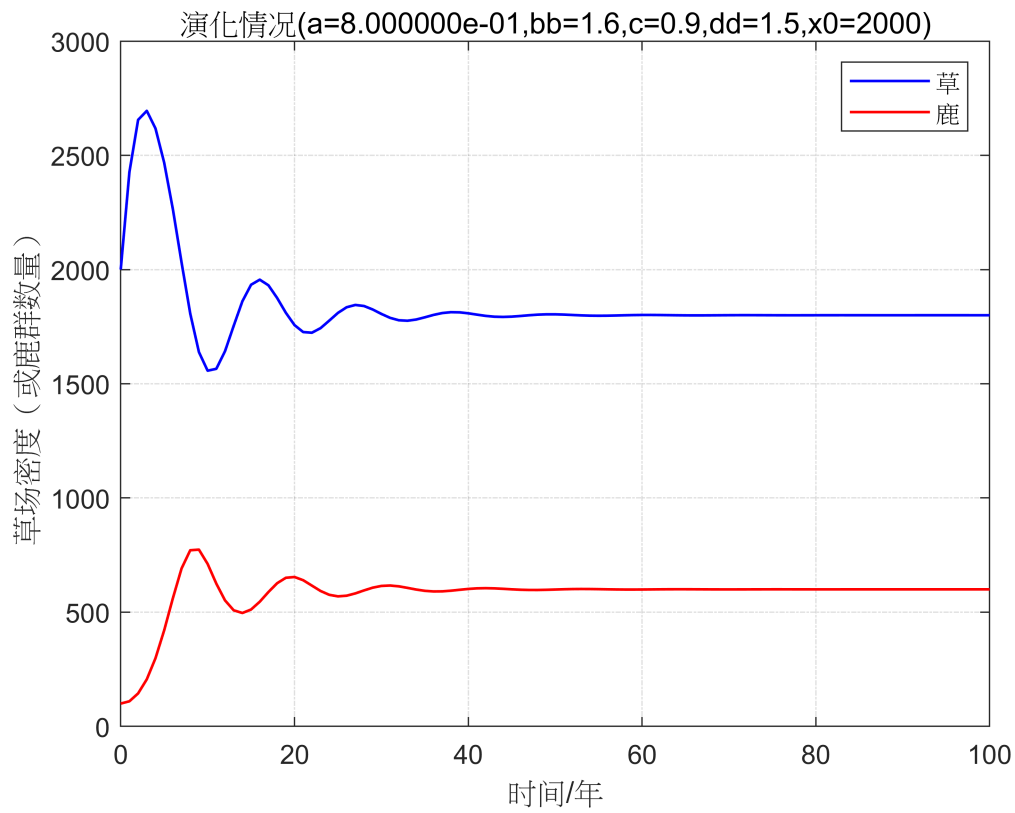
```

• 分析图像：

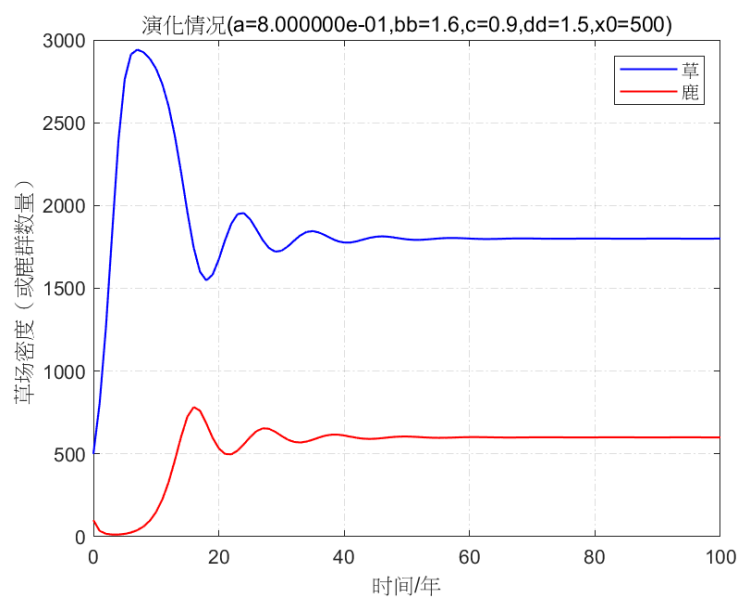
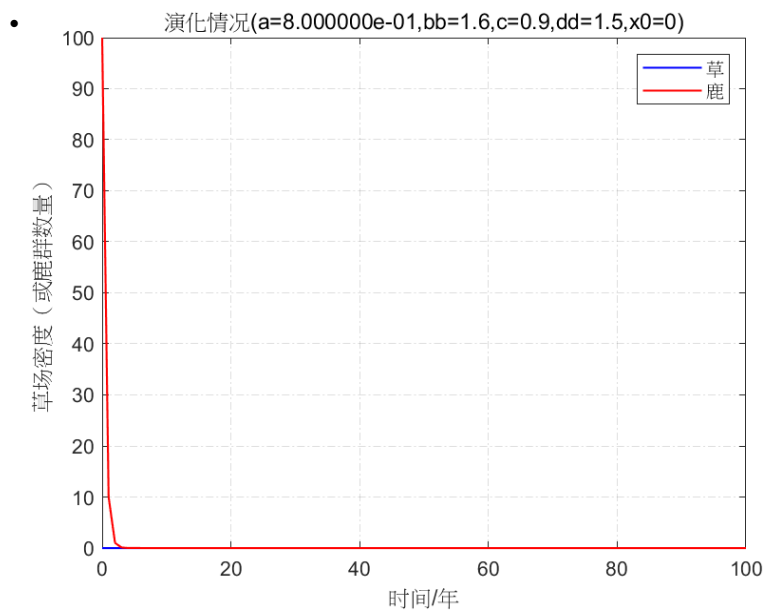
```

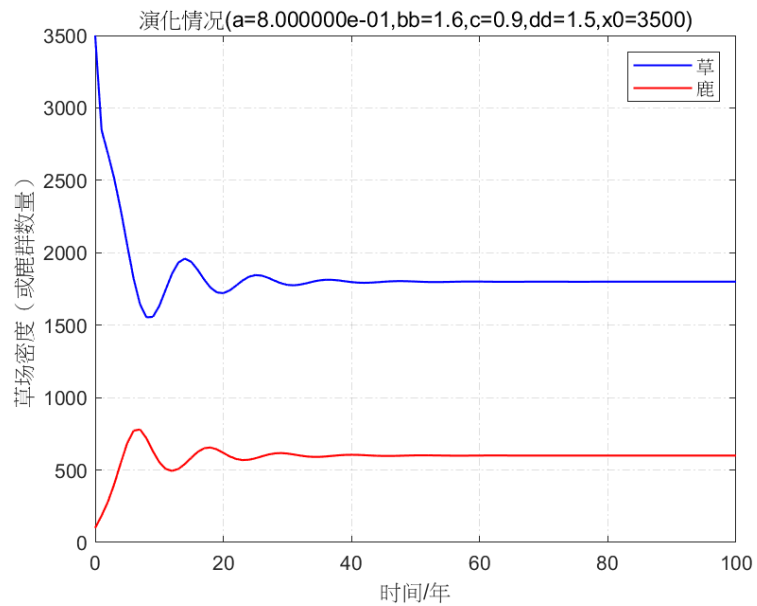
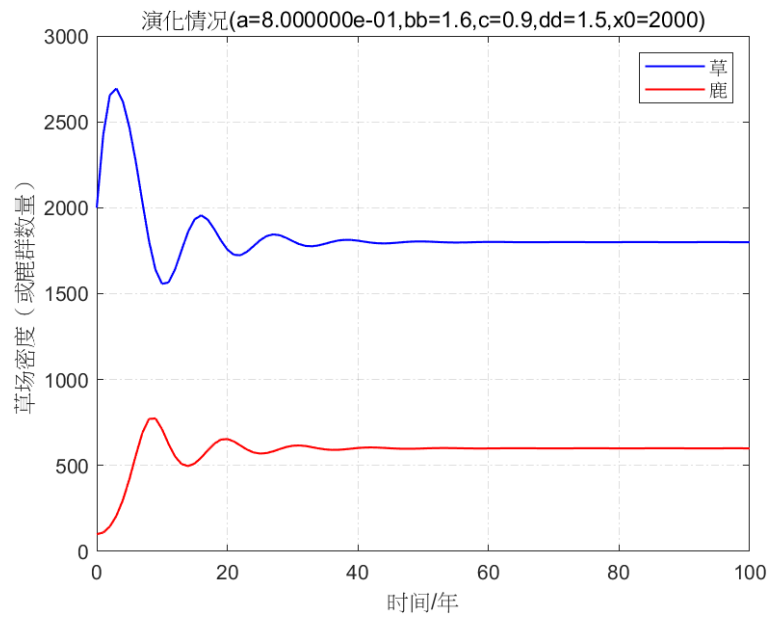
figure;
fig=plot(0:1:T, X, 'b',0:1:T, Y, 'r');
%设置线宽
fig(1).LineWidth =1.0;fig(2).LineWidth =1.0;
%打开网格
grid on;
set(gca, 'XGrid', 'on', 'YGrid', 'on', 'GridLineStyle', '-.');
%设置坐标轴和标题
xlabel('时间/年');
ylabel('草场密度（或鹿群数量）');
legend('草', '鹿');
str = sprintf('演化情况(a=%d,bb=%.1f,c=%.1f,dd=%.1f,x0=%d)', a,bb,c,dd,x0);
title(str);

```

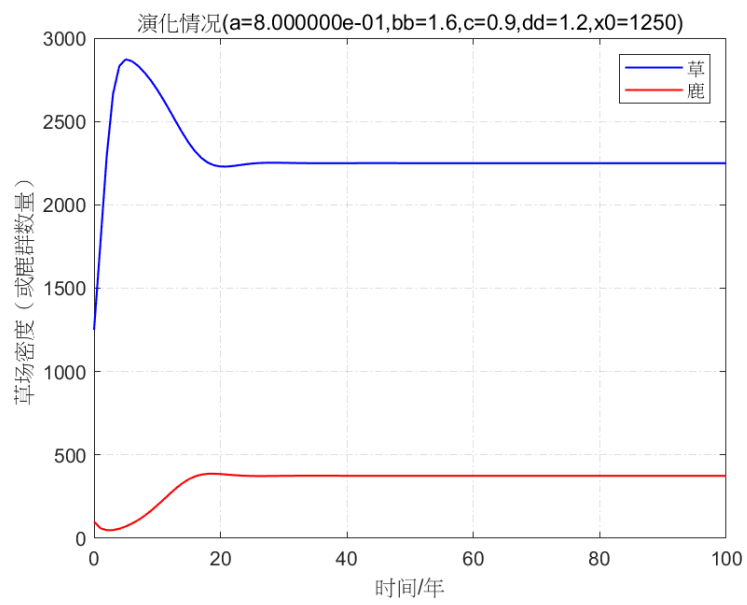
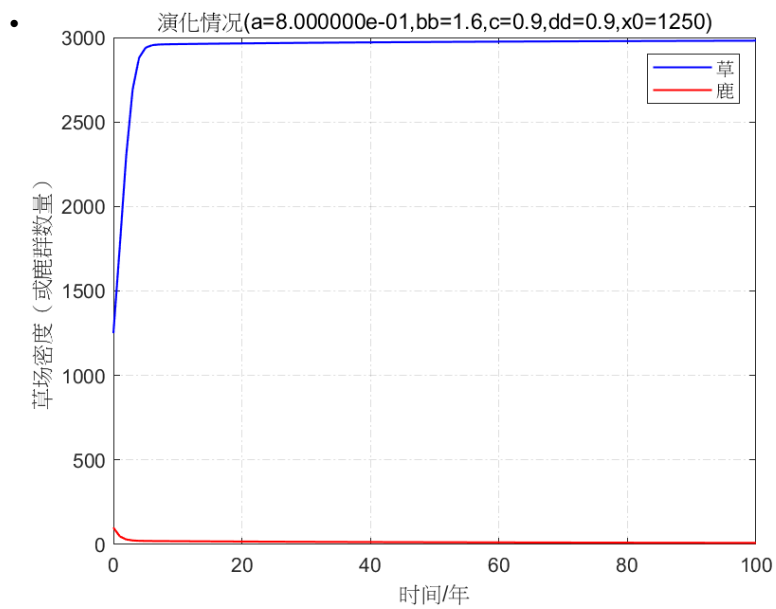


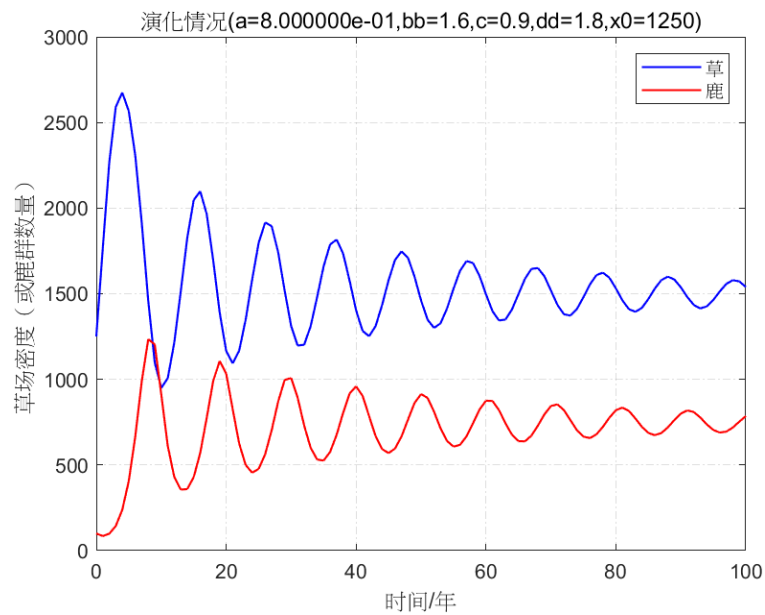
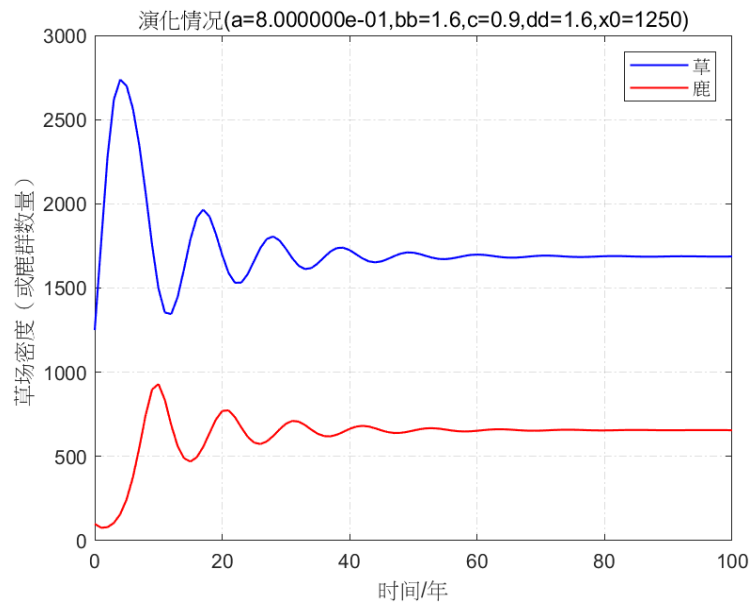
- 经过对比，可以得到以下的结论：
- 调整草场初值，若草场初值为 0，则鹿群在数年内死亡，草场也不会生长；若刚开始草场初值大于 0，则无论其是否大于最大密度，最终草场和鹿群数量都趋于稳定，且稳定值与初值无关。



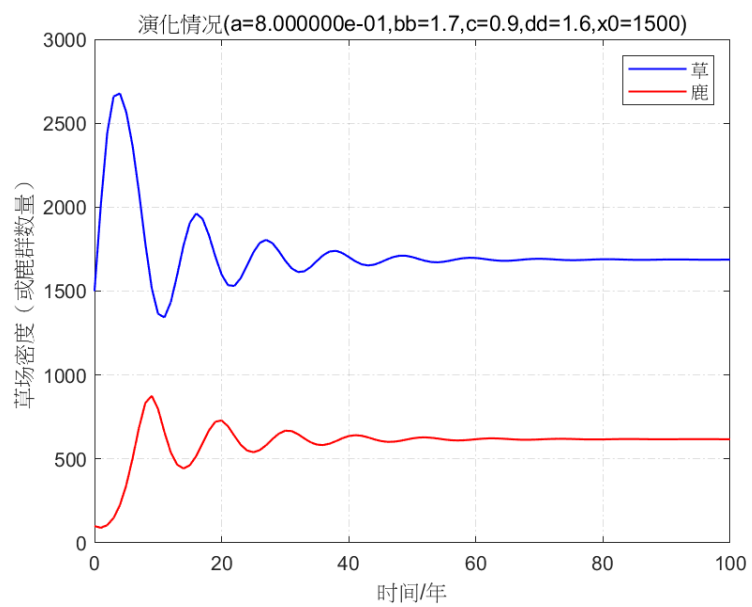
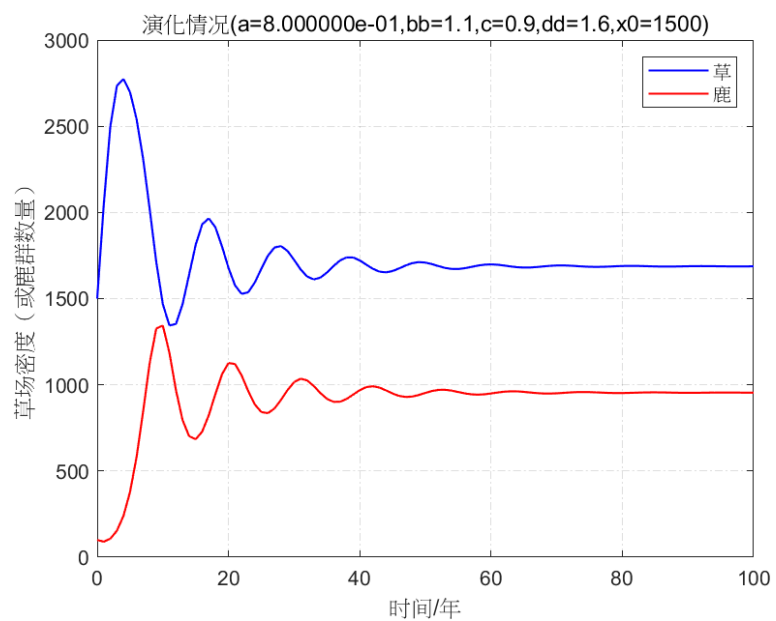
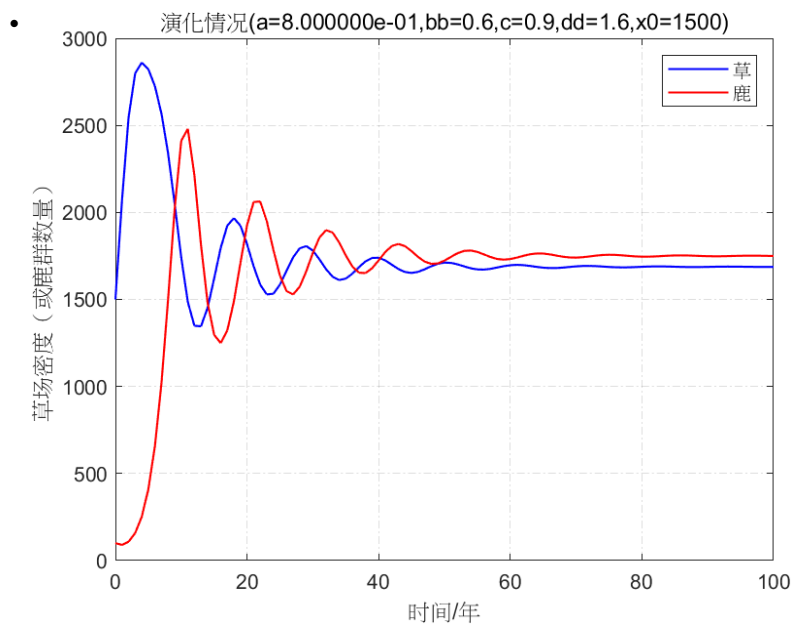


- 调整参数 dd (草最茂盛时对鹿群的供养能力), 发现 dd 越大, 则草和鹿群达到稳定数目的时间越长, 且草的稳定解越小, 鹿的稳定解越大。当 $x_m \cdot dd \leq c$ 时, 发现最终鹿群会走向灭绝, 符合模型假设。





- 调整参数 bb (草最茂盛时鹿群掠取草的能力), 发现 bb 越小, 则草和鹿群达到稳定数目的时间越长, 且草的稳定解越小, 鹿的稳定解越大。在 $bb < 1$ 时, 甚至会出现鹿群数量大于草的情况。



- 调整参数 a (草独立存在时的增长率) 和 c (鹿群独立存在时的死亡率)，发现当 c 较小或 a 较大时，则草和鹿群达成稳定所需时间非常长，所以近似出现周期现象，近似于 Volterra 模型。

