数学实验第三次实验报告

计算机系　计 43　2014011330　黄家晖

2017 年 4 月 2 日

1　实验目的

- 学会使用 MATLAB 求解线性代数方程组，会分析迭代法的收敛性和解的稳定性；
- 掌握用 MATLAB 软件求解非线性方程和方程组的方法，分析收敛、分岔与混沌现象。

2　计算题

2.1　CH5-T9 种群繁殖与收获

2.1.1　数学建模

根据题目意义，如果需要稳定收获（即 $\tilde{x_k} = x_k$），则针对 $x_k(k = 1, 2, \cdots, n)$ 可列出下列线性方程组：

$$\begin{cases} \Sigma_{k=1}^{n} b_k x_k = x_1 \\ s_k x_k - h_k = x_{k+1}, k = 1, 2, \cdots, n-1 \end{cases}$$

记：

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$$

$$\mathbf{h} = (0, h_1, h_2, \cdots, h_{n-1})^T$$

$$\mathbf{S} = \begin{bmatrix} b_1 & b_2 & \cdots & \cdots & b_n \\ s_1 & 0 & \cdots & \cdots & 0 \\ 0 & s_2 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & s_{n-1} & 0 \end{bmatrix} \tag{1}$$

因此，上述线性方程组模型即可化为所求个年龄的稳定种群数量模型：

$$\mathbf{Sx} - \mathbf{h} = \mathbf{x}$$

2.1.2 数值求解

算法设计 上一节中的数学模型可以表示为：

$$(\mathbf{S}-\mathbf{I})\mathbf{x}=\mathbf{h}$$

记 $\mathbf{A}=\mathbf{S}-\mathbf{I}$，通过 MATLAB 中的 `cond`函数计算该矩阵的条件数，判断矩阵是否为病态矩阵。本题中的 $n=5$，可以采用直接法进行求解。

MALTAB 程序 如下：

```
1 %% Math Exp Homework 3 5-T9
2 % Calculate age distribution under the restriction of stability
3
4 %% Input data
5 b = [0 0 5 3 0];
6 s = [0.4 0.6 0.6 0.4];
7 h = [0; 500; 400; 200; 100];
8 %h = [0; 500; 500; 500; 500];
9
10 %% Generate Linear equation model
11 S = [b; [diag(s) zeros(length(s), 1)]];
12 A = S - eye(length(b));
13 fprintf('Cond(A) = %f\n', cond(A));
14
15 %% Get the result x
16 x = A \ h
17 % calculate the error
18 err = sum(sum(A * x - h))
```

如果需要更改 $\mathbf{h}$ 的定义，仅需要更改源程序第 7 行即可。

计算结果 矩阵 $\mathbf{A}$ 的条件数为 87.19，大于 1，因此稍显病态。

当 $h_1,\cdots,h_5$ 分别为 500, 400, 200, 100, 100 时，计算得到：

$$\mathbf{x}=(8481.0,2892.4,1335.4,601.3,140.5)^T$$

以该结果计算得到的真实 $\mathbf{h}$ 与要求的 $\mathbf{h}$ 误差为 -3.7×10^{-13}，可以接受。

当 $h_1,\cdots,h_5$ 均为 500 时，计算得到：

$$\mathbf{x}=(10981.0,3892.4,1835.4,601.3,-259.5)^T$$

以该结果计算得到的真实 $\mathbf{h}$ 与要求的 $\mathbf{h}$ 误差为 -7.4×10^{-13}，该误差可以接受。但是考虑到 $x_5<0$，在实际情况中不可能使得种群数量为负数。因此题目中要求的 $h_1,\cdots,h_5$ 均为 500 不可能达到。但是可以考虑通过人工控制的方式改变物种的死亡率（例如提高生活环境质量）达到预期的收获率：假设我们将 s_4 提高为 0.95，可以得到：

$$\mathbf{x}=(10981.0,3892.4,1835.4,601.3,71.2)^T$$

该结果相比上一结果较为合理。

通过该题的数值计算能够看出来，对于稍显病态的矩阵 $\mathbf{A}$，稍稍更改了 $\mathbf{h}$，计算出的结果就有了较大的差距。

2.2 CH6-T8 价格方程求解中的混沌现象

算法设计 根据商品供求关系，可以得出期望价格 q 的递推公式为:

$$q_{t+1} = \frac{r}{d}[c - \arctan(\mu q_t)] - (r-1)q_t$$

为了观察混沌现象并计算分岔点，可以采用课本中提供的 `chaos`函数画出分岔图进行观察分岔点的大致位置。再通过枚举的方式确定分岔点的精确位置，只需要计算出现两倍于原来个收敛序列的起始点即可。例如，需要计算由 $2^n \to 2^{n+1}$ 个收敛子序列的分岔点，对于参数 c，循环执行函数 $q(t)$ 指定轮数等待收敛模式出现的时候，再采样连续的 2^{n+1} 个点，如果这些点所包含的收敛序列相比参数 $c - \delta c$ 有所减少，则将 c 作为分岔点的估计输出。

由于题目中所给的递推式无法精确求出分岔点的位置，因此在使用穷举法计算分岔点的时候参数的选取（例如收敛判据、收敛轮数）对于最终的计算结果影响较大。本题中仅仅给出我使用参数的计算结果。

MATLAB 程序 主程序如下:

```
%% Math Exp Homework 3 6-T8
% Chaos and forking phenomena

%% Params
init_val = 0.5;

%% Call the chaos function.
chaos(@qt, init_val, [0, 2, 0.001], [1000, 1200]);
%chaos(@qt, init_val, [0.885, 0.9, 0.00001], [1000, 1200]);

%% Find fork point
find_fork(@qt, init_val, 1000, 0.8945:0.0001:0.895, 4, 1e-6)

%% Draw iteration graph.
iter_count = 1000;
c = 0.896;
iter_vec = zeros(iter_count, 1);
iter_vec(1) = init_val;
for i = 2:iter_count
    iter_vec(i) = feval(@qt, iter_vec(i - 1), c);
end
plot(iter_vec);
```

其中 `chaos`函数与课本中的一致:

```
function chaos( iter_fun, x0, r, n )
    kr = 0;
    for rr = r(1):r(3):r(2)
        kr = kr + 1;
        y(kr, 1) = feval(iter_fun, x0, rr);
        for i = 2:n(2)
            y(kr,i) = feval(iter_fun, y(kr, i - 1), rr);
        end
    end
    plot([r(1):r(3):r(2)], y(:, n(1) + 1: n(2)), 'k.');
end
```

`find_fork`函数用于计算分岔点:

```
function [ fp ] = find_fork( iter_fun, init_val, converge_iter, search_range, level, tol )
```

```
% Find the forkpoint of iter_fun.
    fp = -1;
    for c = search_range
        % First iterate.
        iter_val = init_val;
        for i = 1:converge_iter
            iter_val = feval(iter_fun, iter_val, c);
        end
        % Check if this is the fp level.
        % Get consecutive 2 ** level numbers.
        sample_count = 2 ^ (level + 1);
        sample_vector = zeros(sample_count, 1);
        for i = 1:sample_count
            sample_vector(i) = feval(iter_fun, iter_val, c);
            iter_val = sample_vector(i);
        end
        % Check if can converge in Level.
        converge_mat = reshape(sample_vector, [], 2);
        converge_mat = abs(converge_mat(:, 1) - converge_mat(:,2));

        if (sum(converge_mat < tol) == length(converge_mat))
            fp = c;
            break;
        end
    end

end
```

计算结果 使用 `chaos`函数画出的分岔图如图 1所示。该函数的分岔图关于 $c=0$ 对称，而考虑到当 $c<0$ 的时候，会存在 $q<0$ 的情况，不符合价格的相关定义，因此仅仅考虑参数 $c>0, q_t>0$ 时的情况。

由图 2观察得到第一个分岔点区间在 $[1, 1.12]$ 之间，精确计算的结果为 $b_1 = 1.0795$。

由图 3观察得到第二个分岔点区间在 $[0.92, 0.98]$ 之间，精确计算的结果为 $b_2 = 0.9488$。

由图 4观察得到第三个分岔点区间在 $[0.90, 0.91]$ 之间，精确计算的结果为 $b_3 = 0.9073$，以此类推，可以得到 $b_4 = 0.8972, b_5 = 0.8949$。当参数 c 继续减小的时候（0.88 左右），已经不能观察出几个独立的明显的收敛序列了，出现了混沌现象。

为了进一步验证所得分岔点的正确性，选取了 $c = 2.0 > b_1, 1.0 \in (b_2, b_1), 0.92 \in (b_3, b_2), 0.90 \in (b_4, b_3), 0.896 \in (b_5, b_4)$ 分别进行 $q(t)$ 的绘制，所得图像分别如图 5, 6, 7, 8, 9所示。

最后，计算：

$$\frac{b_2 - b_1}{b_3 - b_2} = 3.15$$

$$\frac{b_3 - b_2}{b_4 - b_3} = 4.11$$

$$\frac{b_4 - b_3}{b_5 - b_4} = 4.39$$

可以看到，$\frac{b_n - b_{n-1}}{b_{n+1} - b_n}$ 趋于 4.67（虽然 n 不大），该趋势符合 Feigenbaum 常数定律。

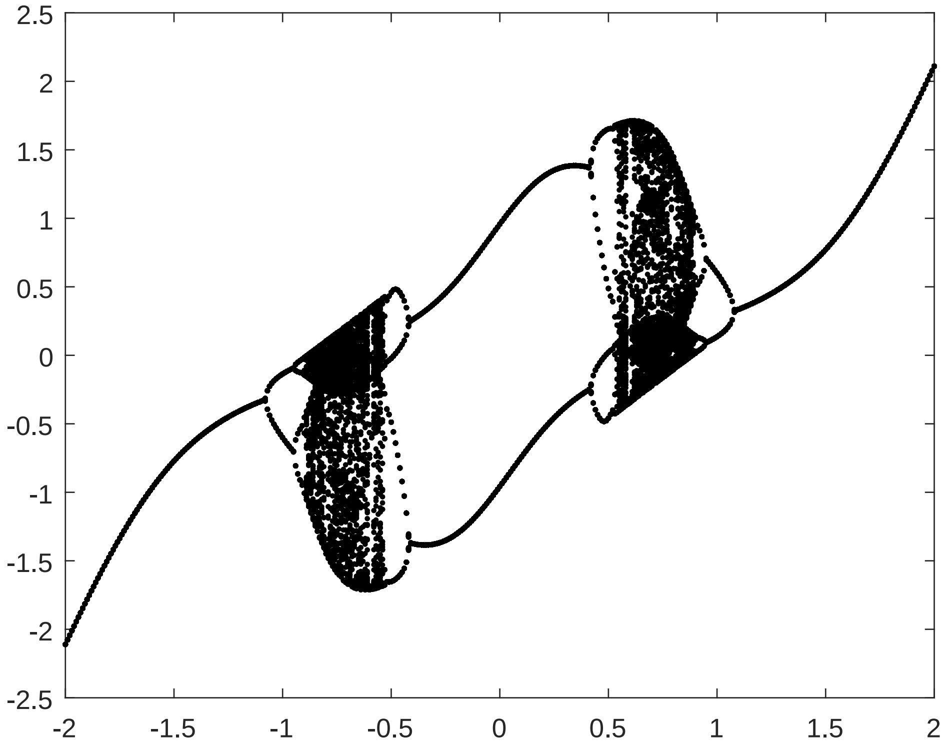

图 1: 使用 chaos函数画出的分岔图

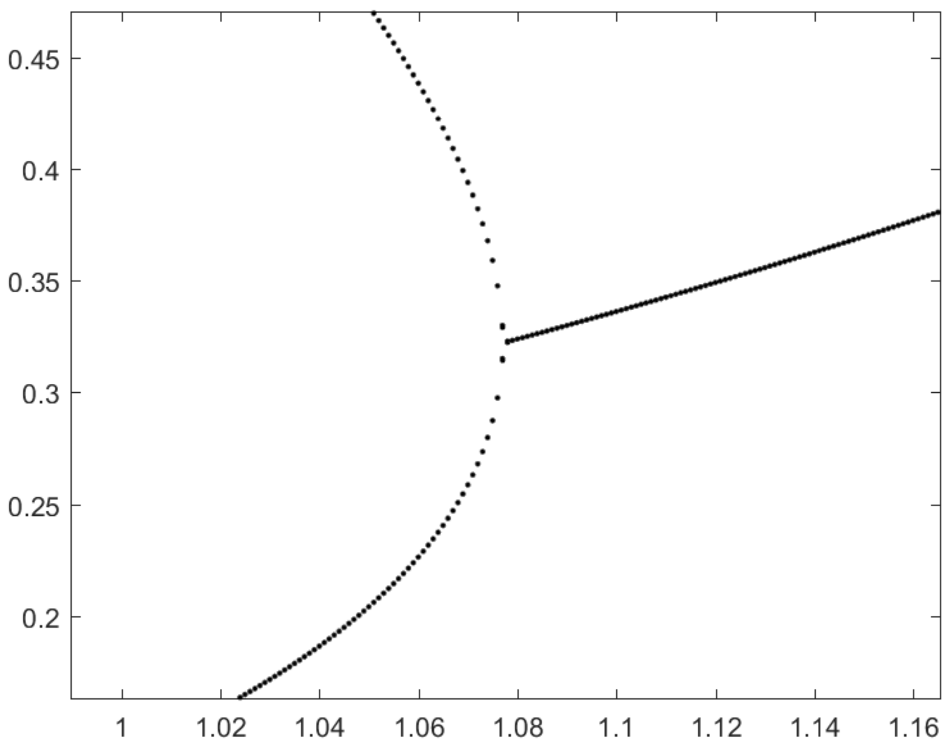

图 2: 寻找第一个分岔点

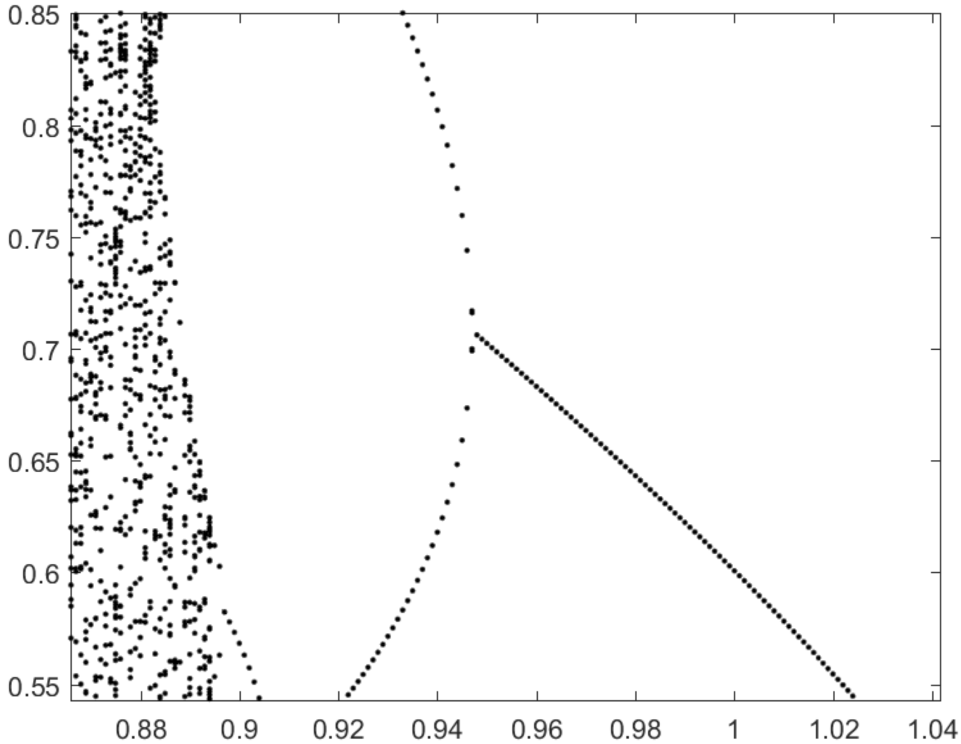

图 3: 寻找第二个分岔点

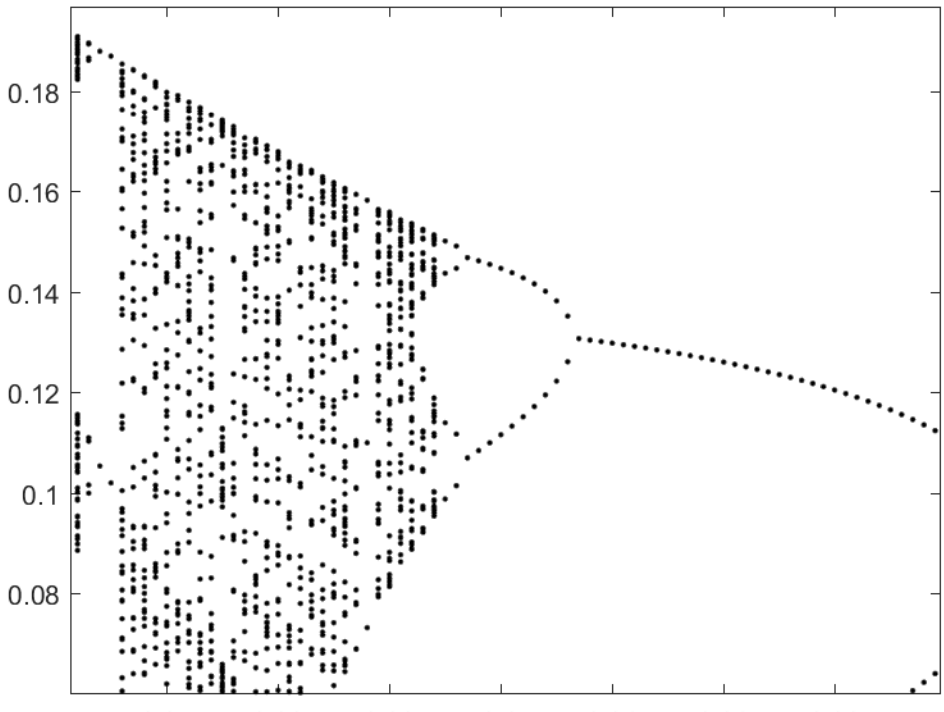

图 4: 之后的分岔点与混沌现象

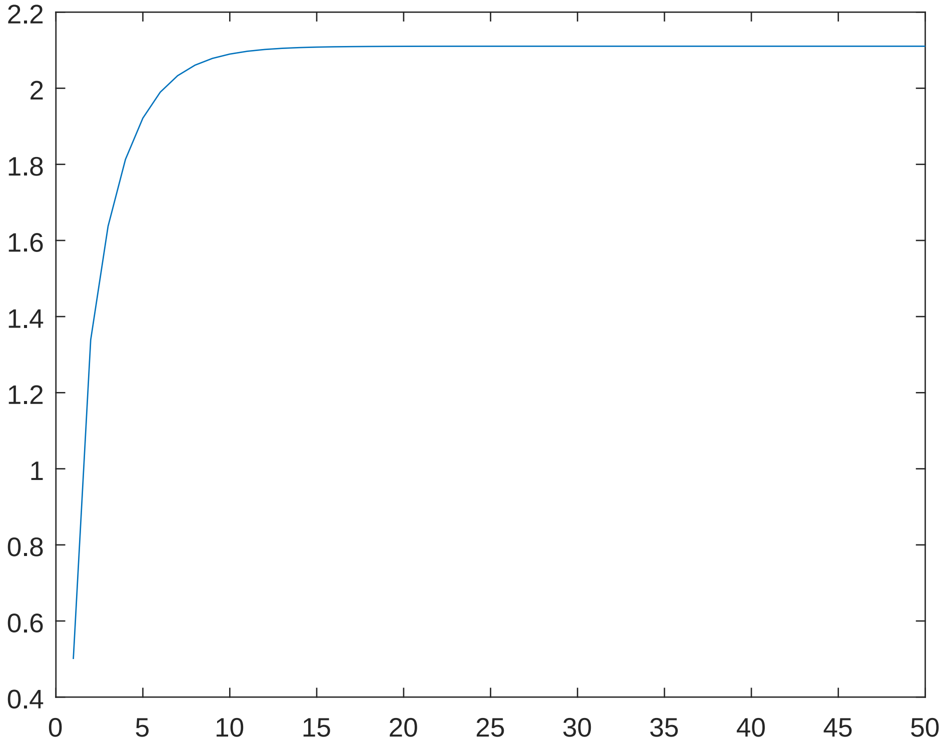

图 5: $c > b_1$ 的情况，此时稳定收敛

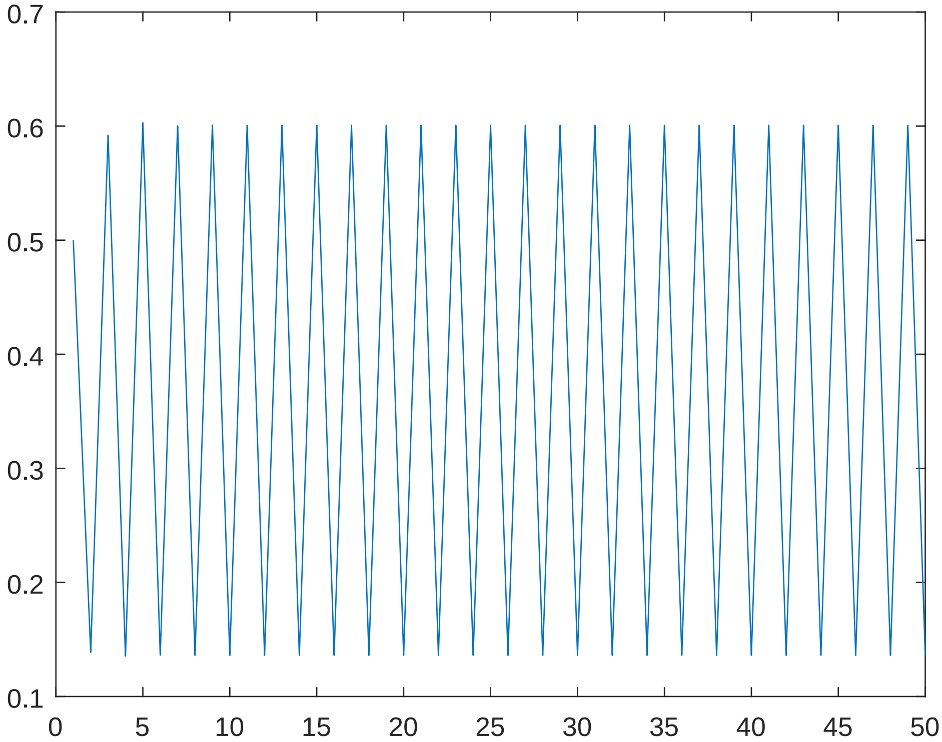

图 6: $b_2 < c < b_1$ 的情况，可见有两个独立的收敛序列

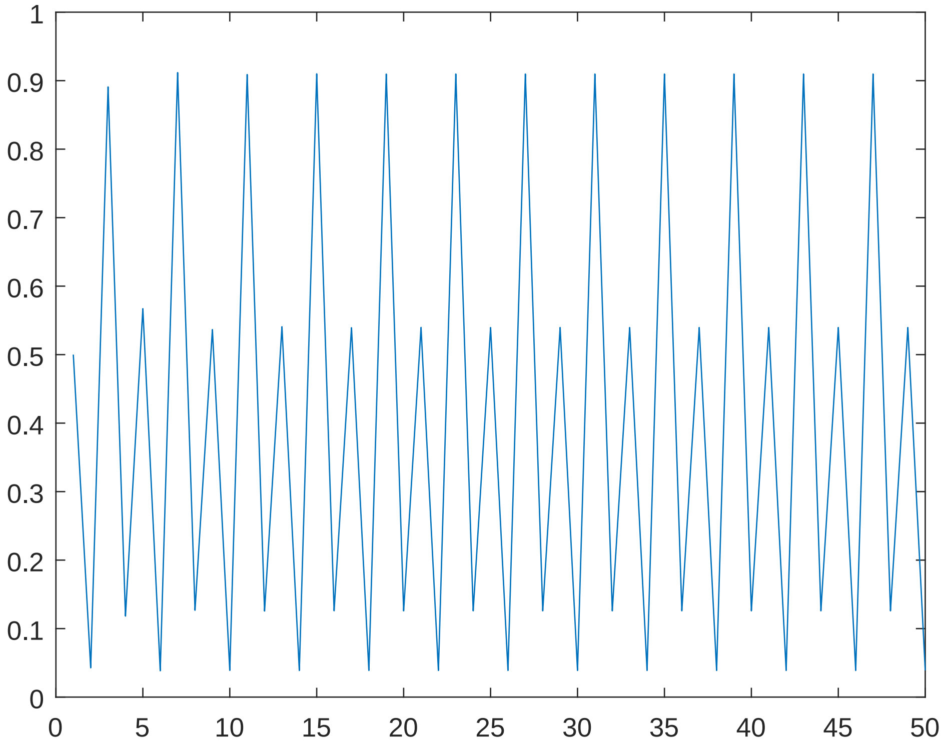

图 7: $b_3 < c < b_2$ 的情况，可见有四个独立的收敛序列

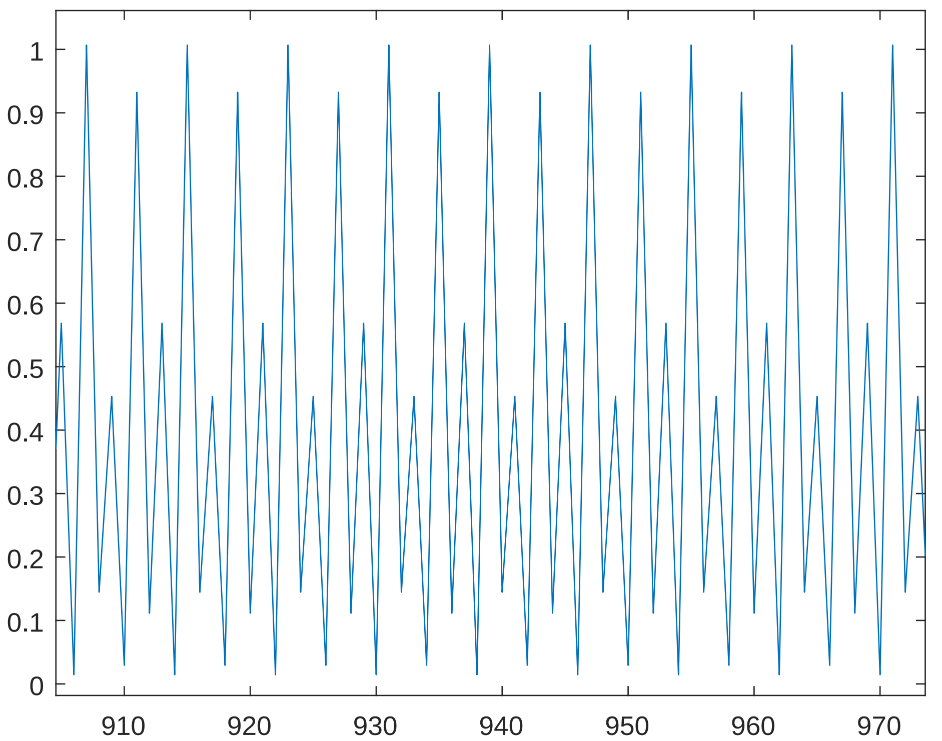

图 8: $b_4 < c < b_3$ 的情况，可见有八个独立的收敛序列

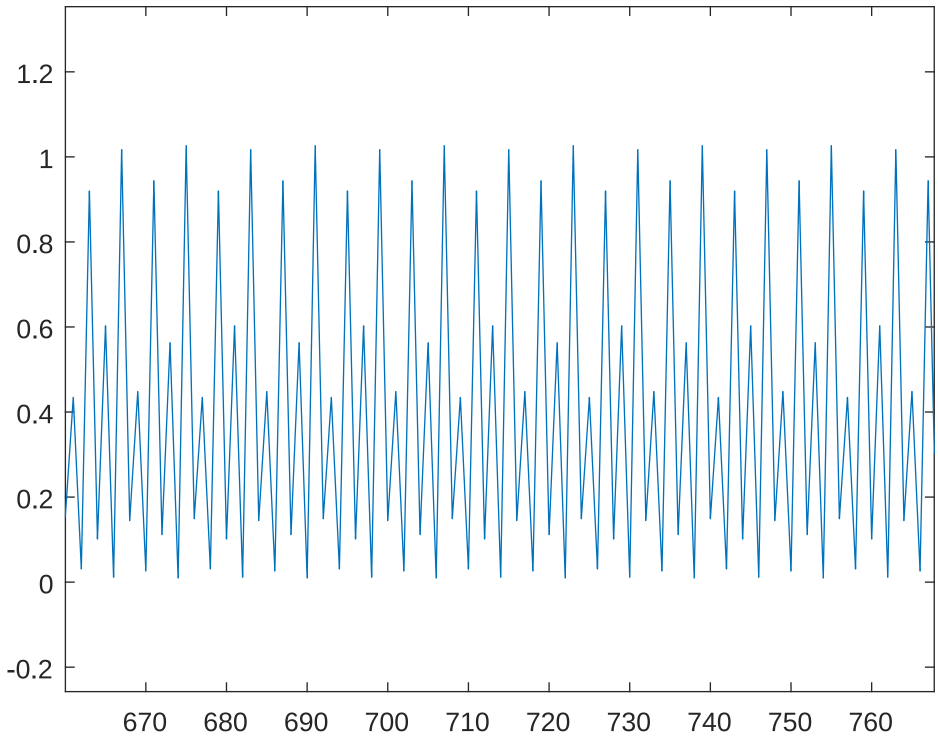

图 9: $b_5 < c < b_4$ 的情况，可见有 16 个独立的收敛序列，注意相邻的两个峰实际上从属于两个不同的收敛序列

3 应用题

3.1 CH6-6 均相共沸混合物求解

3.1.1 模型建立与算法设计

基本建立模型同例题，设 x_i 为第 i 种组分的占比，T 为温度，则对于该系统，有下列方程成立：

$$x_i[\frac{b_i}{T+c_i}+\ln(\Sigma_{j=1}^n x_j q_{ij})+\Sigma_{j=1}^n(\frac{x_j q_{ji}}{\Sigma_{k=1}^n x_k q_{jk}})-1-a_i+\ln P]=0, i=1,2,\cdots,n$$

在实际进行求解的时候，利用共沸物组分之和为 1 的条件有：

$$x_n=1-\Sigma_{i=1}^{n-1}x_i$$

可以列出关于 $x_1,\cdots,x_{n-1}$ 和 T 的 n 个非线性方程组，该方程组难以求出解析解，因此使用 MATLAB 中的 `fsolve`函数进行数值求解。

3.1.2 MATLAB 程序

```
%% Math Exp Homework 3 6—T6
% Component of homogeneous azeotrope

%% Input data
n = 4;
P = 760;
a = [18.607, 15.841, 20.443, 19.293]';
b = [2643.31, 2755.64, 4628.96, 4117.07]';
c = [239.73, 219.16, 252.64, 227.44]';
Q = [1.0 0.192 2.169 1.611;
     0.316 1.0 0.477 0.524;
     0.377 0.360 1.0 0.296;
     0.524 0.282 2.065 1.0];
% Initial
XT0 = [0.0, 0.0, 0.0, 100]';

%% Start solver
[XT, Y, conv, opt] = fsolve(@azeofun, XT0, [], n, P, a, b, c, Q)
```

其中 `azeofun`函数实现如下，相对于书上了进行了一定的优化：

```
function f = azeofun( XT, n, P, a, b, c, Q )
    x = [XT(1:n—1); 1 — sum(XT(1:n—1))];
    sigmai = Q * x;
    f = x .* (b ./ (XT(n) + c) + log(sigmai) + Q' * (x ./ sigmai) — ...
        a — 1 + log(P));
end
```

3.1.3 计算结果

当初值为 $[0.25, 0.25, 0.25, 50]$ 时，能解出：

$$x_1=0.0\%,\quad x_2=58.6\%,\quad x_3=41.4\%,\quad x_4=0.0\%,\quad T=71.97$$

当初值为 $[0.0, 0.50, 0.50, 90]$ 时，能解出：

$$x_1 = 0.0\%, \quad x_2 = 78.3\%, \quad x_3 = 0.0\%, \quad x_4 = 21.97\%, \quad T = 76.96$$

当初值为 $[1.0, 0.0, 0.0, 90]$ 时，能解出：

$$x_1 = 0.0\%, \quad x_2 = 0.0\%, \quad x_3 = 0.0\%, \quad x_4 = 100.0\%, \quad T = 97.77$$

当初值为 $[0.0, 0.0, 0.0, 150]$ 时，能解出：

$$x_1 = 0.0\%, \quad x_2 = 0.0\%, \quad x_3 = 100.0\%, \quad x_4 = 0.0\%, \quad T = 82.56$$

可以看出，在找出的几组解中，没有出现所有的 x_i 均大于 0 的情况，这四种物质可能不能共同稳定存在。又由共沸物的定义，必须由一种以上的物质，上述几种情况中有些仅仅有一种物质存在（其他物质的成分占比均为 0.0%），不符合定义。综上所述，最终符合条件的物质配比和温度为：

$$x_1 = 0.0\%, \quad x_2 = 58.6\%, \quad x_3 = 41.4\%, \quad x_4 = 0.0\%, \quad T = 71.97$$

或

$$x_1 = 0.0\%, \quad x_2 = 78.3\%, \quad x_3 = 0.0\%, \quad x_4 = 21.97\%, \quad T = 76.96$$

4 收获与建议

通过这次的实验，我对 MATLAB 中提供（非）线性方程组求解函数理解更加深刻，通过实际编程、画图的方式观察了方程求解的结果，这是书本上无法学到的知识。同时，在做上机实验的过程中，我对 MATLAB 这款软件的使用也更加熟练了。希望在之后的课堂上老师能够当堂进行相关的技巧演示并给出题目的分步解答。