针对 Logistic 映射的混沌现象的探索

数学科学学院

PB18010496 杨乐园

序言

在数学分析 A1 教学过程中，我们跳过了一节，即混沌现象，鉴于本人对该现象的好奇与未知，决定利用 Mathematica 探索并研究一些现象。

首先，什么是混沌现象？通过数学分析与微分方程等相关课程讲解，我们知道利用一些简单的迭代格式可以求解非线性方程，并且符号计算软件课程也展示了通过迭代产生一些非常复杂而漂亮的图形。而事实上，若迭代序列不收敛，则可能出现以下两种情况：

（1）当迭代次数足够充分大时，迭代序列出现周期性重复，即存在自然数 N，$k > 0$，使得$x_{N+k} = x_N$，迭代序列为：

$$x_0，x_1，\ \ldots，x_N，x_{N+1}，\ \ldots，x_{N+k-1}，x_N，x_{N+1}，\ \ldots，x_{N+k-1}，\ \ldots$$

此时，$x_N，x_{N+1}，\ \ldots，x_{N+k-1}$称为周期为 k 的循环，而初始点$x_0$称为预周期点。

（2）迭代序列“似乎”没有任何可寻的规律，显得杂乱无章，出现了“看似”随机的行为，此时称为混沌。

通过查阅文献和历史了解到，在混沌现象发现之前，人们在观念上认为事物的发展是服从确定论的，但在具体研究某些问题的时候，总还是会遇到无法预测的事实。法国数学家庞加莱在 19 世纪末研究天体力学时，特别是三体问题时发现了混沌。他在《科学与方法》一书中以绕太阳运动的小行星为例，指出如果它们平均每天相差超出千分之一秒，那么三年相差将超出 1 秒，一万年相差 1 度，三四百万年将超出一个圆周！这是令人惊叹的事实。这种初始条件的微小误差促成最后现象的极大误差所描述的正是通常成语所说的“差之毫厘，谬以千里”。

1963 年美国气象学家洛伦兹发表了他对大气流的研究工作，揭示出混沌现象具有不可预言性和对初始条件的极端敏感性这两个基本特点，同时，他还发现表面上看起来杂乱无章的混沌现象，仍然具有某种条理性。

基于本人求知欲，本实验从简单的迭代出发，直观认识混沌现象并观察其蕴含的规律。

实验内容

First Part：对函数$f(x)=-2+\sin 1.5x$，分别取初值$x_0=0$和$x_0=-0.7$进行迭代。利用蛛网图直观观察迭代序列

$$x_0,\ f(x_0),\ f(f(x_0)),\ f(f(f(x_0))),\ \cdots\cdots$$

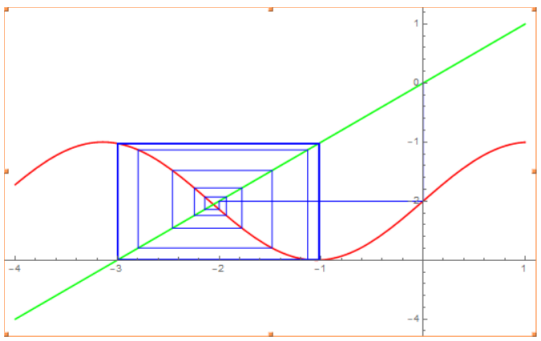
的循环与混沌现象。

程序代码如下：

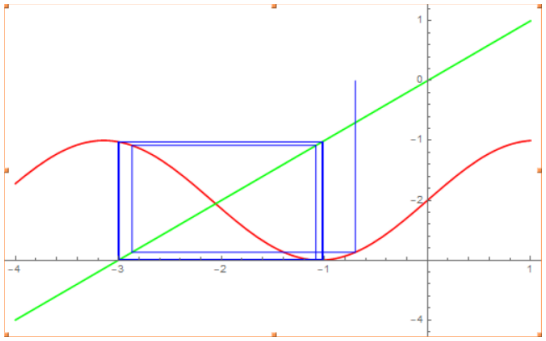
```
f[x_]:=-2+Sin[1.5 x];
f1=Plot[f[x], {x,-4,1},PlotStyle->RGBColor[1,0,0]];
f2=Plot[x, {x,-4,1},PlotStyle->RGBColor[0,1,0]];
x0=0;r={};
r0=Graphics[{RGBColor[0,0,1],Line[{{x0,0}, {x0,x0}}]}];
For[i=1, i<100, i++,r=Append[r,Graphics[{RGBColor[0,0,1],Line[{{x0,
x0}, {x0,f[x0]}, {f[x0],f[x0]}}]}]];x0=f[x0]];
Show[f1,f2,r,r0,PlotRange->{-4,1}]

x0=-0.7;r={};
r0=Graphics[{RGBColor[0,0,1],Line[{{x0,0}, {x0,x0}}]}];
For[i=1, i<100, i++,r=Append[r,Graphics[{RGBColor[0,0,1],Line[{{x0,
x0}, {x0,f[x0]}, {f[x0],f[x0]}}]}]];x0=f[x0]];
Show[f1,f2,r,r0,PlotRange->{-4,1}]
```

运行程序代码，所得结果如下图所示：



(a) $x_0=0$时迭代出现循环



(b) $x_0=-0.7$时迭代出现混沌

我们可以从中看出$x_0=0$时该迭代得到了一个周期为 2 的循环，0 是该循环的一个预周期点；而$x_0=-0.7$时，迭代不具有任何可预测的模式，即出现混沌现象。

Second Part：二次函数$f(x)=\alpha x(1-x)\ (0\le x\le 1)$称为 Logistic 映射。单单从数学上面看这个映射（函数），这是一个十分简单的二次函数，但它却是产生混沌现象的一个范例，并且这个简单的映射孕育着数学中可能有的最复杂、最优美的性态。它说明简单的系统可能产生复杂的行为，即激烈的变化不一定有激烈的原因。

对于 Logistic 映射$f(x)=\alpha x(1-x)\ (0\le x\le 1)$，取初值$x_0=0.2$，分别对

$$\alpha=2.9,\ 3.4,\ 3.5,\ 3.7$$

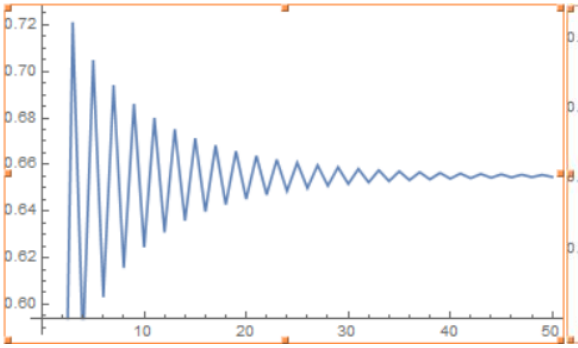
观察其迭代序列的收敛情况。

程序代码如下：

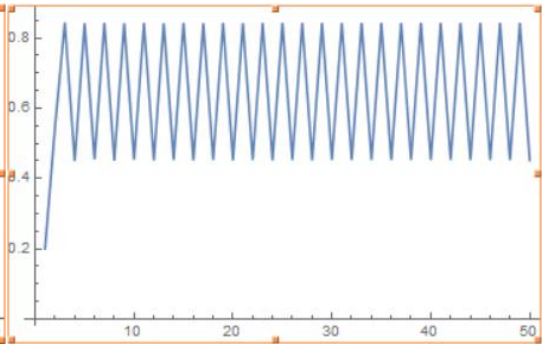
（修改不同的 r 即可）

```
xn={0.2};r=2.9;
For[i=1, i<50, i++, xn=Append[xn, r*xn[[i]]*(1-xn[[i]])];]
xn
ListPlot[xn,PlotJoined->True]
```

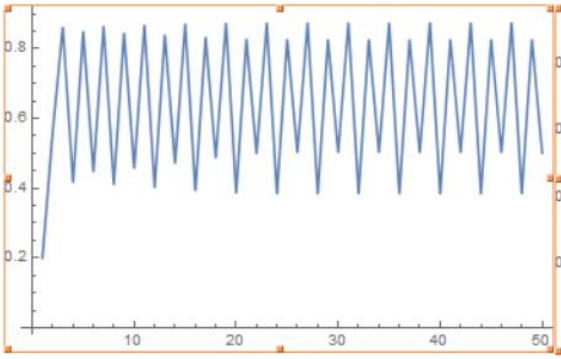
取不同的α时，运行程序所得结果如下图所示：



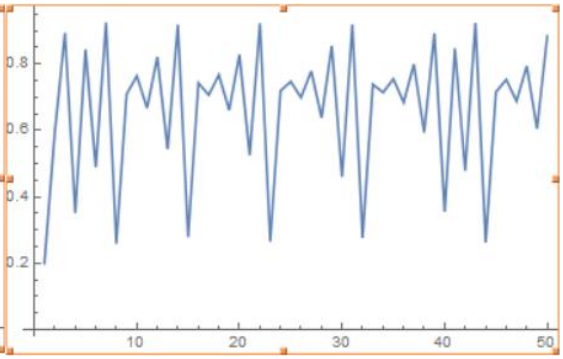
(a) $\alpha = 2.9$



(b) $\alpha = 3.4$



(c) $\alpha = 3.5$



(d) $\alpha = 3.7$

观察不同的α值，可以看出，当$\alpha = 2.9$，$x_0 = 0.2$时，迭代出现上下振荡并趋向于一个不动点。当$\alpha = 3.4$，$x_0 = 0.2$时，迭代序列经过一段时间振荡后开始在两个近似值 0.42 和 0.82 值之间上下振荡，且出现周期性的重复现象，简称 2-周期振荡。当$\alpha = 3.5$，$x_0 = 0.2$时，迭代序列经过一段时间振荡后开始在四个值之间上下振荡，且出现周期性的重复，简称 4-周期振荡。注意到当参数α变化时，迭代序列从 1-周期即不动点振荡变化到 2-周期振荡，再从 2-周期振荡变化到 4-周期振荡，这种有规律的分裂现象称为分叉现象（参考数学分析教程上册 2.12 混沌现象）。

但是当$\alpha = 3.7$，$x_0 = 0.2$时，此时迭代序列不再出现稳定的周期性，也不具有任何可预测性的模式，即出现混沌现象。

Third Part：由于蛛网迭代可以非常直观形象的描述迭代序列的变化，故对上述四个不同α取值的 Logistic 映射，画出其蛛网图，并观察其迭代行为。

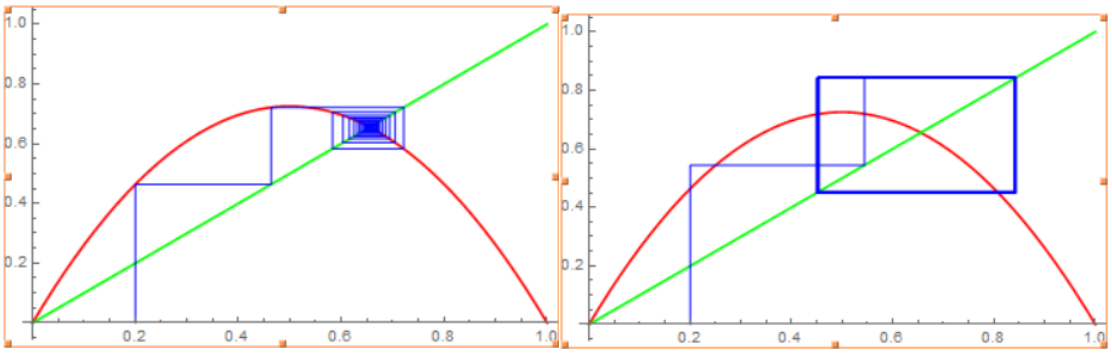
程序代码如下：

（修改不同的 u 即可）

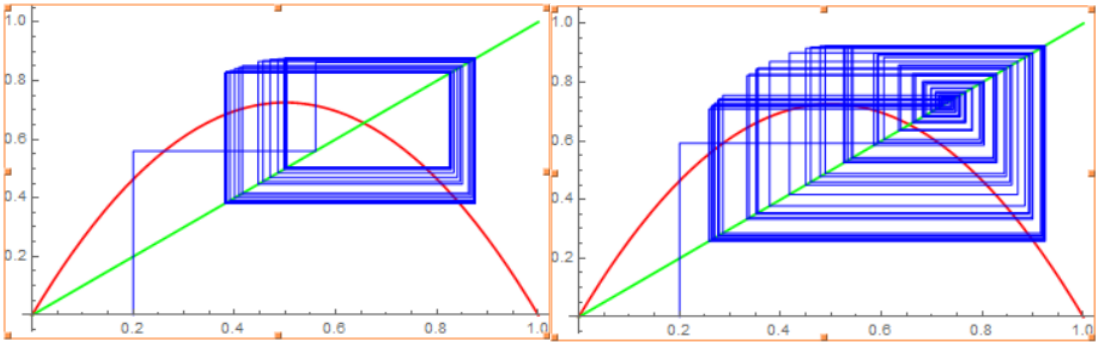
```
u=2.9;
```

```
g[x_]:=u x(1-x);
g1=Plot[g[x], {x,0,1},PlotStyle->RGBColor[1,0,0]];
g2=Plot[x, {x,0,1},PlotStyle->RGBColor[0,1,0]];
x0=0.2;r={};
r0=Graphics[{RGBColor[0,0,1],Line[{{x0,0}, {x0,x0}}]}];
For[i=1, i<100, i++,r=Append[r,Graphics[{RGBColor[0,0,1],Line[{{x0,
x0}, {x0,g[x0]}, {g[x0],g[x0]}}]}]];x0=g[x0]];
Show[g1,g2,r,r0,PlotRange->{0,1}]
```

取不同的α时，运行程序所得结果如下图所示：



(a) $\alpha = 2.9$　　(b) $\alpha = 3.4$



(c) $\alpha = 3.5$　　(d) $\alpha = 3.7$

对照 Second Part 的图像，可以看到α取不同值时，Logistic 映射迭代序列或趋于一个不动点，或逐渐周期性重复，或走向无规则的混沌现象。

Forth Part：为了更清楚的认识到混沌现象对初始条件的极端敏感性，取四个差别非常小的初值，计算$\alpha = 4$时的 Logistic 映射的迭代序列的前 50 项数值。

程序代码参照 Second Part 即可。

Mathematica 运算结果如下：

当$\alpha = 4$，$x_0 = 0.1$时计算结果为：

0.1, 0.36, 0.9216, 0.289014, 0.821939, 0.585421, 0.970813, 0.113339, 0.401974, 0.961563, 0.147837, 0.503924, 0.999938, 0.000246305, 0.000984976, 0.00393603, 0.0156821, 0.0617448, 0.23173, 0.712124, 0.820014, 0.590364, 0.967337, 0.126384, 0.441645, 0.986379, 0.053742, 0.203415, 0.64815, 0.912207, 0.320342, 0.870893, 0.449754, 0.989902, 0.039986, 0.153548, 0.519885, 0.998418, 0.00631654, 0.0251066, 0.0979049, 0.353278, 0.913891, 0.314778, 0.862771, 0.473588, 0.99721, 0.0111304, 0.0440261, 0.168351, 0.560037

当$\alpha = 4$，$x_0 = 0.1000001$时计算结果为：

0. 1, 0. 36, 0. 9216, 0. 289013, 0. 821937, 0. 585426, 0. 97081, 0. 113353, 0. 402016, 0. 961596, 0. 147715, 0. 503582, 0. 999949, 0. 000205318, 0. 000821103, 0. 00328172, 0. 0130838, 0. 0516504, 0. 195931, 0. 630167, 0. 932226, 0. 252722, 0. 755415, 0. 739052, 0. 771416, 0. 705333, 0. 831353, 0. 560822, 0. 985203, 0. 0583125, 0. 219649, 0. 685612, 0. 862192, 0. 475267, 0. 997553, 0. 00976397, 0. 0386745, 0. 148715, 0. 506396, 0. 999836, 0. 000654488, 0. 00261624, 0. 0104376, 0. 0413145, 0. 158431, 0. 533321, 0. 995559, 0. 0176859, 0. 0694925, 0. 258653, 0. 767007

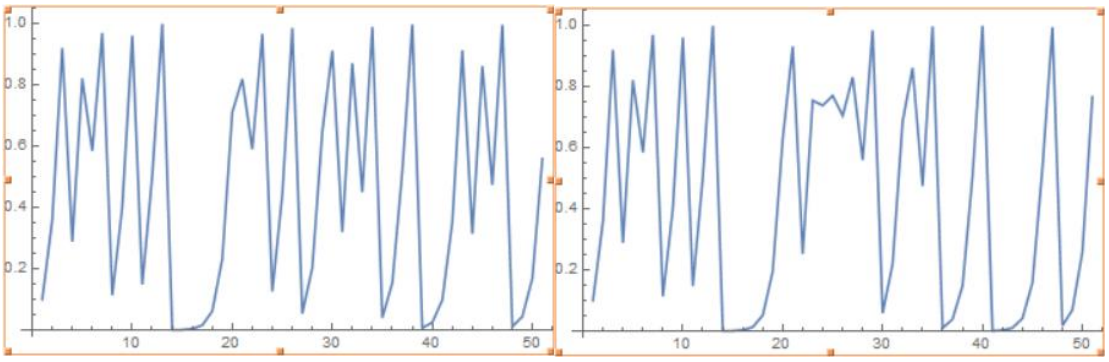
当$\alpha = 4$，$x_0 = 0.10000001$时计算结果为：

0. 1, 0. 36, 0. 9216, 0. 289014, 0. 821939, 0. 585421, 0. 970813, 0. 113341, 0. 401978, 0. 961567, 0. 147824, 0. 50389, 0. 999939, 0. 000242038, 0. 000967919, 0. 00386793, 0. 0154119, 0. 0606974, 0. 228053, 0. 704179, 0. 833244, 0. 555795, 0. 987548, 0. 0491887, 0. 187077, 0. 608316, 0. 95307, 0. 178909, 0. 587603, 0. 969303, 0. 119018, 0. 419411, 0. 974022, 0. 101214, 0. 363879, 0. 925885, 0. 274489, 0. 79658, 0. 648161, 0. 912193, 0. 320388, 0. 870958, 0. 44956, 0. 989823, 0. 040293, 0. 154678, 0. 523011, 0. 997882, 0. 00845385, 0. 0335295, 0. 129621

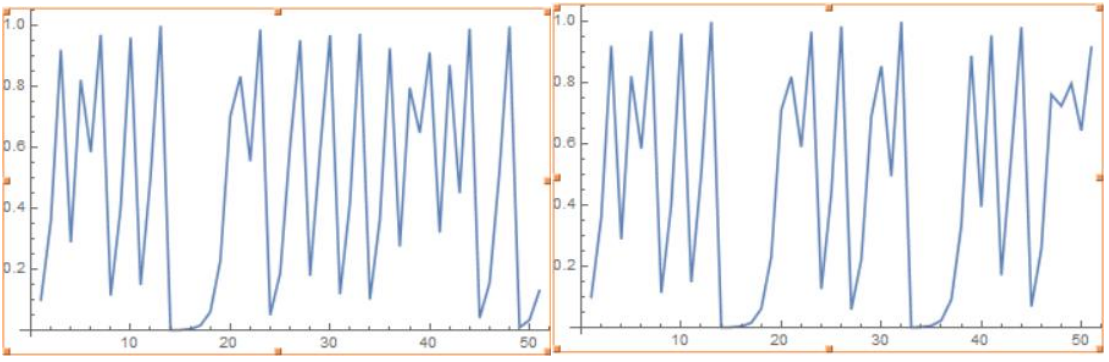
当$\alpha = 4$，$x_0 = 0.1000000001$时计算结果为：

0. 1, 0. 36, 0. 9216, 0. 289014, 0. 821939, 0. 585421, 0. 970813, 0. 113339, 0. 401974, 0. 961564, 0. 147836, 0. 503923, 0. 999938, 0. 000246262, 0. 000984805, 0. 00393534, 0. 0156794, 0. 0617343, 0. 231693, 0. 712045, 0. 820148, 0. 590021, 0. 967585, 0. 125457, 0. 438869, 0. 985052, 0. 0588977, 0. 221715, 0. 69023, 0. 85525, 0. 49519, 0. 999907, 0. 000370115, 0. 00147991, 0. 00591089, 0. 0235038, 0. 0918054, 0. 333509, 0. 889123, 0. 394334, 0. 955339, 0. 170666, 0. 566157, 0. 982493, 0. 0688011, 0. 25627, 0. 762383, 0. 724621, 0. 798182, 0. 644349, 0. 916654

迭代图示为：



(a) $x_0 = 0.1$　　(b) $x_0 = 0.1000001$



(c) $x_0 = 0.10000001$　　(d) $x_0 = 0.1000000001$

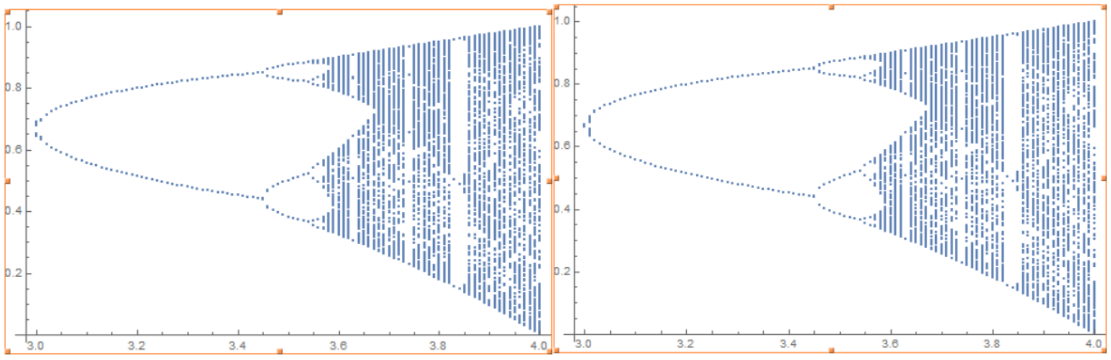
注意到上述四个初值的差别非常小，仅在小数点后七、八、十位上有些许差别。观察数值结果以及四幅图像，会发现经过大概 10 次迭代后所得结果差别不大，大约都为 0.9615，但经过 50 次迭代后所的结果差别就很大了。这就是混沌现象的极端敏感性。

Fifth Part：对于 Logistic 映射$f(x) = \alpha x(1-x)\ (0 \le x \le 1)$，取$\alpha = 3$时，在（0,1）中随机取一数作为初值$x_0$进行迭代，共迭代 300 次左右，丢弃起始的 100 次迭代数据，在图上绘制出所有的点（α，x_n）$(n > 100)$，然后取步长为 0.01 慢慢地增加α值，每增加一次都重复之前的步骤，一直增加到$\alpha = 4$为止，这样得到的图形称为 Feignbaum 图。接下来绘制出 Logistic 映射的 Feignbaum 图并观察其迭代行为。

程序代码如下：

```
h[t_,x_]:=t*x*(1-x);
x0=0.5;r={};
Do[For[i=1,i<=300,i++,x0=h[t,x0];If[i>100,r=Append[r,{t,x0}]]],{t,3.0,4.0,0.01}];
ListPlot[r]
```

运行结果如下图：



(a) $x_n = 0.5$　　(b) $x_n = 0.354367$

为了得到更细致的图形，可将α的步长进一步缩小。从上图可以看出，较左的部分是一些清晰的曲线段，这说明对该范围内的任一值α而言，当进行到 100 次以后，迭代所得的x_n只取有限的几个值，表明该迭代序列构成了一个循环，其周期等于竖直的直线与图形交点个数；从左到右，每一段曲线到一定位置同时一分为二，表明迭代序列周期在这些位置处增长了一倍，因而曲线的种种分叉称为倍周期分支，有时也称 Pitch-Fork 分支（参见数学分析教程上册 2.12 混沌现象）；随着α的增长，出现分支位置的间隔越来越少，大约在$\alpha = 3.57$左右，分支已经看不清楚，这时便出现了混沌现象，由此向右的区域被称为混沌区域。但需注意，并不是所有大于 3.57 的α，函数都出现混沌，在图中可以看到混沌区域中有一些空白带，说明仍出现了周期循环。

总结

之所以选择 Logistic 映射进行研究，是因为其具有广泛应用，如对人口变化规律的研究。简单介绍如下

世界人口的增长，受到自然资源的约束，设地球允许承载的总人数为 M，以 x(t)表示时刻为 t 时人口总数，t 到 t+Δt 的人口平均增长速度，既与 t 时人口总数成正比，也与容许增长的人口数成正比，故有

$$\frac{x(t+\Delta t)-x(t)}{\Delta t}=kx(t)(M-x(t))$$

其中 k 为比例系数。设以年为时间单位，当 Δt=1 时，上式变为

$$x(t+1)-x(t)=kx(t)(M-x(t))$$

若以 n 代替 t，并令$x_n=\frac{k}{1+kMx(n)}$，则有

$$x_{n+1}=(1+\mathrm{kM})x_n(1-x_n)$$

令$\alpha=1+\mathrm{kM}$，可得

$$x_{n+1}=\alpha x_n(1-x_n)$$

此式正是 Logistic 构成的迭代式。

因此，足见 Logistic 映射所发挥的巨大作用。

曾经的数学研究，对于极限$\lim_{n\to\infty} f^{(n)}(x)$，一旦确定其极限不存在之后，就将其“搁置一边，不予理睬”，但有时这个迭代所带来的周期性与混沌性，以及各种各样的复杂变化，广泛应用于物理、化学、生物、以及社会科学等众多领域。