

5.3 北京拉面的数学模型

我们大都品尝过拉面面条,这种风味食品我国各地都有,例如北京拉面、兰州拉面、乐亭拉面等。厨师把一根一定长度的精制面粉的面团均匀拉长成原来的两倍,再把右手的一端交给左手,右手抓住中点,再拉长成两倍长,如此反复进行下去,直到把面条拉得十分细软,煮熟后再加入鸡汤或牛肉汤等,吃起来十分可口。1998年,中国中央

电视台曲艺杂坛节目播出一位高级厨师的拉面绝活儿，这位师傅拉伸了 50 多个回合，把一根面柱拉伸成原来粗细的 $\frac{1}{2^{50}}$ 以下，比头发还要细，创立了吉尼斯世界纪录。

自然界物质的运动，人类社会人群的活动，如果局限在一个有限的范围内发展变化，免不了扩张拉伸，同时由于有一个限制，也免不了发生折叠，下面我们以北京拉面这种实际模型来模拟拉伸折叠现象，建立它的数学模型，且分析其中的混沌表现。值得提醒的是北京拉面并不是随机过程，而是一种简单规则支配之下的确定性过程。它的数学模型是 $[0, 1]$ 到自身的映射 Δ 的迭代：

$$\Delta: y = \begin{cases} 2x, & 0 \leq x < \frac{1}{2} \\ 2 - 2x, & \frac{1}{2} \leq x \leq 1 \end{cases} \quad (5.3)$$

(5.3) 式的图像像一个三角帐篷，见图 5-1，故称 Δ 为三角帐篷映射。

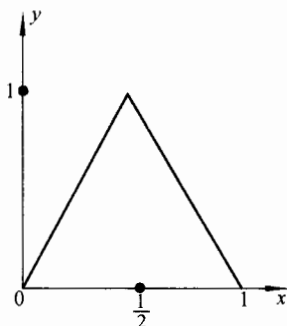


图 5-1

在二进制表达式中，(5.3) 式的迭代为

$$x_{n+1} = \begin{cases} 2x_n, & 0 \leq x_n < \frac{1}{2} \\ 2 - 2x_n, & \frac{1}{2} \leq x_n \leq 1 \end{cases}$$

$$n = 0, 1, 2, \dots$$

我们看到, $0 \leq x_n < \frac{1}{2}$ 时, 这就是砍头映射 σ 的迭代。当 $\frac{1}{2} \leq x_n \leq 1$ 时, 在二进制当中

$$x_n = 0.1a_2a_3 \cdots a_n \cdots$$

$$x_{n+1} = \Lambda(x_n) = 0.\bar{a}_2\bar{a}_3 \cdots \bar{a}_n \cdots \quad (5.4)$$

其中 \bar{a}_i 是 a_i 的“补”, 即 $a_i + \bar{a}_i = 1$, $a_i, \bar{a}_i \in \{0, 1\}$, 就是 $a_i = 0$ 时, $\bar{a}_i = 1$, $a_i = 1$ 时, $\bar{a}_i = 0$ 。

事实上, 由于 $x_n \in \left[\frac{1}{2}, 1\right]$, 所以

$$\begin{aligned} x_n &= \frac{1}{2} + a_2 \frac{1}{2^2} + a_3 \frac{1}{2^3} + \cdots + a_n \frac{1}{2^n} + \cdots \\ 2 - 2x_n &= 2 - 2 \left(\frac{1}{2} + a_2 \frac{1}{2^2} + a_3 \frac{1}{2^3} + \cdots + a_n \frac{1}{2^n} + \cdots \right) \\ &= 1 - \left(a_2 \frac{1}{2} + a_3 \frac{1}{2^2} + \cdots + a_n \frac{1}{2^{n-1}} + \cdots \right) \\ &= \left(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots + \frac{1}{2^{n-1}} + \cdots \right) \\ &\quad - \left(a_2 \frac{1}{2} + a_3 \frac{1}{2^2} + \cdots + a_n \frac{1}{2^{n-1}} + \cdots \right) \\ &= (1 - a_2) \frac{1}{2} + (1 - a_3) \frac{1}{2^2} + \cdots \\ &\quad + (1 - a_n) \frac{1}{2^{n-1}} + \cdots \\ &= 0.\bar{a}_2\bar{a}_3 \cdots \bar{a}_n \cdots \end{aligned}$$

上面的推导告知, 当 $x_n \in \left[\frac{1}{2}, 1\right]$ 时, $\Lambda(x_n)$ 的动作是下列两个步骤的实施:

- ①把小数点后第一个数字删去(砍头)。
- ②把小数点后的数字都变成其补数, 即 0 变成 1, 1 变成 0。