

晶体生长计算机模拟中的一种省时的蒙特-卡罗方法

刘光照

(中国科学院上海硅酸盐研究所)

如果不考虑晶体生长单元在体相内的扩散,那么晶体生长动力学可以看作是由吸附、蒸发和表面迁移等随机过程组成的。蒙特-卡罗方法可以对这些过程直接进行模拟。目前,蒙特-卡罗方法已用于研究晶体表面的平衡性质和晶体生长(或其倒易过程)动态性质的许多方面^[1-4]。

由于蒙特-卡罗方法是一种统计试验方法,它是借助随机数进行一次次的尝试(抽样)来获得整个系统(总体)的某些数字特征的。一般来说,尝试的次数越多结果越可靠。而从统计物理角度看,这些尝试组成了相空间点的马尔可夫链,人们正是借助这条链来计算系统的某些平均性质。如果该链收敛很慢,那么需要的尝试次数便多,因此在蒙特-卡罗模拟中常常遇到计算量大、计算时间长的困难。本文介绍一种省时间的方法。采用这种方法可以提高计算效率,特别是当所模拟的各类“事件”发生的几率相差大时,效果更加显著。

将整个系统分隔成相同大小和相同形状的单胞。这些单胞要不是固态就是流体态。如果限制固态单胞只能排列在固态单胞之上,便得到所谓的 SOS (solid on solid) 模型。一般模拟的是 kossel 晶体即简单立方晶体的 {100} 面。

发生在晶面上的固态和流体态之间单胞的交换,决定于“产生”(单胞从流体相变为固相)和“湮没”(单胞从固相变为流体相)的动力学速度常数。

令 k^+ 为“产生”的速度常数(即该事件发生的频率或几率), k_n^- 为具有 n 个水平方向最近邻的固态单胞变为流体的“湮没”的速度常数。根据细致平衡原理,可以得到^[1,5-7]

$$k^+ = k_{0q}^+ \exp(\Delta\mu/kT), \quad (1)$$

$$k_n^- = k_{0q}^- \exp[(4-2n)\omega] \\ n = 0, 1, 2, 3, 4, \quad (2)$$

其中 $\Delta\mu = \mu_f - \mu_s$ 是单胞在流体相和固相中化学势的差; k_{0q}^+ 为 $\Delta\mu = 0$ 时的平衡“产生”频率; ω 为固-流体键的形成能:

$$\omega = \left(\phi_{sf} - \frac{1}{2} \phi_{ff} - \frac{1}{2} \phi_{ss} \right) / kT. \quad (3)$$

ϕ_{ss} , ϕ_{ff} 和 ϕ_{sf} 分别代表固-固、流-流和固-流体键的势能。

在进行模拟时,需要对上述几率进行归一化,归一化的方法可参阅文献[4]中的图(2)。

所模拟的界面用一个二维数组 $\{S_{i,j}\}$ 表示。其中 i, j 代表界面上某个座位的坐标, S 表示该座位上固态胞的高度。为了减小有限列阵的边界对模拟结果的影响,需要使用周期性边界条件^[4]。

蒙特-卡罗模拟借助在 0 与 1 之间均匀分布的随机数 R 进行。先用第一个随机数 R_1 决定事件发生的地点,然后借助第二个随机数 R_2 决定在该座位上何种事件将要发生。一般使用的二维列阵为 50×50 , 对每个座位大约要进行 400 次左右的尝试以达到平稳状态,所以需要一百万次的计算。

一、一种省时的蒙特-卡罗方法

要提高计算效率,一方面可以靠编制计算机程序的熟练技巧,另一方面则靠尽量减少无效的尝试。本文介绍的方法属于后一类。这种方法首先由 van der Eerden 等人^[8]在模拟小晶粒的表面迁移时采用,后来本文作者又与 van der Eerden 和 Bennema^[9]一起用于模拟应力对晶体生长和溶解的影响,效果十分显著。

这种方法的特点是将借助于在(0, 1)之间均匀分布的随机数平均抽样(称为离散型方法)改为把时间作为一个连续变量进行抽样(称为连续型抽样)。因此, 需要把事件发生的几率转换成事件发生所需要的平均“等待”时间, 然后借助于在(0, 1)之间均匀分布的随机数计算实际的“等待”时间, 并按时间长短排列构成一个“等待时间表”。等待时间短的事件首先被执行。下面以模拟应力对晶体生长和溶解的影响为例^[9], 来说明如何采用这种连续型方法。

假定在一个各向同性的介质中有一个位错, 在位错线的周围必定联系着一个圆柱形的应力场, 依然根据细致平衡原理^[4, 5-7, 9], 在这种情况下事件发生的几率为

$$\begin{aligned} k^+ &= k_{eq}^+ \exp(\Delta\mu/kT), \\ k_n^- &= k_{eq}^- \exp[(4 - 2n)\omega \\ &\quad + \Omega U(r)/kT], \end{aligned} \quad (4)$$

式中 Ω 为单胞的体积, $U(r)$ 是离位错线 r 处的应变能密度, 它随 r 增大而减小^[10]。

我们采用上节叙述的常规离散型方法进行模拟, 结果表明, 在一个 43×43 的列阵上进行一百万次以上的尝试后, 界面几乎没有什么变化, 显然无效尝试太多, 因此我们改用省时的连续型方法。

由(1)和(4)式可以得到在界面上一给定位置处两次“产生”和两次“湮没”事件之间的平均相隔(或等待)时间, τ^+ 和 τ^- , 它们分别为

$$\begin{aligned} \tau^+ &= \tau_{eq}^+ \exp(-\Delta\mu/kT), \quad (5) \\ \tau_n^- &= \tau_{eq}^- \exp[(2n - 4)\omega \\ &\quad - \Omega U(r)/kT], \quad (6) \end{aligned}$$

这里 $k_{eq}^+ \tau_{eq}^+ = 1$ 。

由于这些事件可以看作是相互独立的, 因而连续两次事件发生的实际相隔时间 t 应围绕平均相隔时间 τ 呈泊松分布:

$$P(t) = (1/\tau) \exp(-t/\tau). \quad (7)$$

借助于随机数 $R \in (0, 1)$, 可以得到相应的具有泊松分布的实际等待时间^[9]:

$$t^+ = -\tau^+ \ln R, \quad (8)$$

$$t_n^- = -\tau_n^- \ln R. \quad (9)$$

在实际工作时, 首先设 $\tau_{eq} = 1$, 并根据(5)

和(6)式分别计算两次“产生”事件的平均等待时间和第一近邻数 n 分别为0, 1, 2, 3, 4的两次湮没之间的平均等待时间。然后对所模拟的列阵(设其大小为 $M \times N$)上的每个组元(即界面上的每个座位)逐一计算其实际等待时间。对于每个这样的座位, 利用两套随机数, 由公式(7)和(8)可以计算出两个数值。一个是在该座位上发生“产生”事件时需要的实际等待时间 t^+ , 另一个是在同一座位上发生湮没事件时所需要的实际等待时间 t_n^- 。将这两个时间进行比较, 如果 $t^+ < t_n^-$, 那么在该座位上发生的事件应为“产生”, 并以 t^+ 作为其真正的等待时间。令其为正值, 而 t_n^- 将废弃不用; 反过来, 如果 $t_n^- < t^+$, 那么在该座位上发生的事件应为湮没, 并以 t_n^- 作为其真正的等待时间。令其值为负, t^+ 将弃之不用。在对每个座位逐一计算后, 我们得到了 $M \times N$ 个正负数, 将这些数字按绝对值进行排列后, 便组成了等待时间表(表1), 绝对值最小者位于表首, “正”表示“产生”, “负”表示“湮没”。

表 1

-20	-31	+32	+37	+40	-41	-48	...
(10,9)	(15,14)	(11,13)	(12,16)	(6,8)	(1,1)	(5,10)	...

(共 $M \times N$ 对数字)

计算机执行位于表首的事件, 例如表1中发生在 $i = 10, j = 9$ 处的事件为湮没, 所以 $S_{10,9} \rightarrow S_{10,9} - 1$ 。

上述事件发生后, 除去 $S_{10,9}$ 发生变化外, 它还可能影响到它的四个近邻的配位数。对于受该事件影响的那些位置, 需要重新决定事件的性质及其等待时间, 并且重新排列, 构成一个新的等待时间表, 然后再一次执行位于表首的事件。因此, 归纳起来, 计算机的工作步骤为: (1) 执行第一事件; (2) 对受该事件影响的单胞重新计算并进行排队; (3) 回到步骤(1), 直至执行了足够的尝试; (4) 给出输出。

结果表明, 在同样进行百万次的尝试后, 界

(下转第 691 页)