

默认单位:

温度——K (开尔文)

时间——s (秒)

长度——Å(埃)

能量——eV(电子伏)

频率——Hz (赫兹)

通量——atom/m²/s(个原子每平方米每秒)

一、背景知识与算法

1.1 基本原理

蒙特卡洛(MC)方法,是指利用计算机产生的伪随机数来模拟随机过程的一类方法。这类方法起源于二战期间,曾部署在世界上第一台可编程电脑 ENIAC 中,用来研究核材料当中中子的随机运动,可以说是最早的一类计算机模拟方法。动力学蒙特卡洛(KMC)是 MC 方法的一种,与其他 MC 方法一样, KMC 方法利用伪随机数来模拟物理体系中的随机热扰动,从而推动体系的随机演化。与常见的 MC 方法不同之处在于, KMC 模拟不关心体系在势能的极小值附近的随机热振动,仅仅考虑那些引起体系构型变化(即从一个能量极小值跳到另一个极小值,例如缺陷在晶体内扩散一步)的热扰动。由于在绝大多数时间内体系都处于前一种状态,忽略掉这一部分无关紧要的热振动,能够极大的提高模拟效率。因此, KMC 通常被用于模拟体系的长时间演化。此外,软件采用的是更为简化的 OKMC 模拟方法,其原理与普通的 KMC 方法一致,但 OKMC 方法将无缺陷的基底晶格看作一个均匀的背景,并把不同的缺陷或缺陷团簇抽象成不同形状的对象。体系的微观结构演化就由这些缺陷对象在均匀背景中的离散化运动主导(例如空位的移动步长为其第一近邻距离)。得益于其更高的抽象程度, OKMC 的模拟时间尺度往往能与许多宏观实验进行直接对比(小时~天量级)。

在 OKMC 模拟中,体系的演化过程被认为是一个马尔可夫过程。换句话说,体系从构型 i 跃迁到构型 j 的概率,仅仅取决于初末态 i 和 j 的性质,与体系在到达 i 态之前的演化路径无关。在数学上,两个能量极小值之间必然存在一个极大值或鞍点,所以从 i 构型跃迁到 j 构型必然需要跨越一个能垒。如图 1 所示,假设这个能垒为 E_{ij}^b ,那么应用过渡态理论[1]可知相应的跃迁速率为:

$$r^{ij} = v_{ij}^0 \times \exp\left(\frac{-E_{ij}^b}{k_B T}\right), \quad (1)$$

其中 k_B 为玻尔兹曼常量, T 为体系温度, 尝试频率 v_{ij}^0 为:

$$v_{ij}^0 = \prod_{n=1}^{3N} v_i^n / \prod_{n=1}^{3N-1} v_{sad}^n, \quad (2)$$

其中 N 为体系自由度, v_i^n 和 v_{sad}^n 为体系在 i 处和 ij 之间鞍点处的实频简振模(简谐近似下的声子频率)。需要注意的是,虽然这里明确给出了获得 v_{ij}^0 的方法,但实际中求解方程 2 需要进行费时的声子频率计算。考虑到作为指前系数, v_{ij}^0 对跃迁速率的影响并不明显。因此,实际过程中往往将 v_{ij}^0 设置为一个常数,通常在 10^{12-13} Hz 左右。

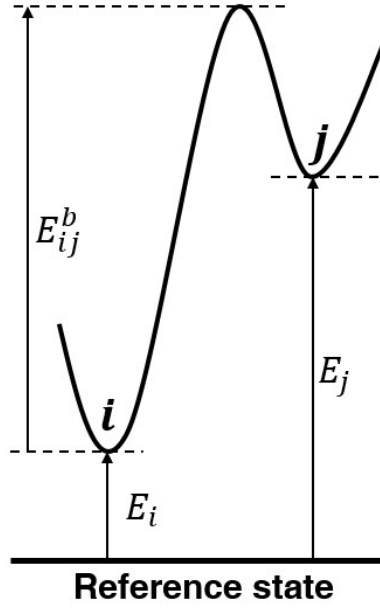


图 1 两个构型附近的能量示意图

由于体系自由度通常很高，处于 i 处的体系往往有许多可能的跃迁终点，即存在很多潜在的末态 j ，体系最终朝哪个末态跃迁是随机的。另外，除了热激活的跃迁会导致体系构型发生变化之外，本文所关心的离子辐照损伤也会引发构型变化。这类辐照损伤也是随机产生的，其发生速率为：

$$r^{ion} = f \times S, \quad (3)$$

其中 f 为辐照通量， S 为辐照方向上的表面积。体系最终选择发生 X 动力学事件（ X 可以是某个热激活的跃迁，也可以是引入一次辐照损伤）的概率正比于其速率：

$$p^X = r^X / \sum_x r^x. \quad (4)$$

相应的，体系每发生一次动力学事件所需的平均时间为：

$$\Delta t = -\ln(rd) / \sum_x r^x. \quad (5)$$

其中 rd 为 0-1 之间的随机数。注意，OKMC 模拟需要知道每一个构型跃迁的能垒。然而由于能垒数据需要进行非常耗时的 DFT 计算获得，几乎不可能预先穷举出所有可能的跃迁能垒。因此，常见的作法[2]是用第一性原理计算精确获得一部分能垒，如较小的缺陷团簇的结合能。然后分析能垒随团簇尺寸变化的物理规律，从而推测/拟合出较大团簇的相应能垒。有了这些能垒信息后，我们便可以按照方程 1-5 所述的方法随机的改变体系构型，并推动演化时间的前进。不断重复这一步骤，便可实现体系的长时间演化模拟。

1.2 基本过程与算法

本 OKMC 软件主要用于模拟离子辐照环境下材料中的缺陷长时间演化问题，考虑的物理过程主要包括辐照级联损伤和缺陷的产生，缺陷的扩散、结合、湮灭（空位与自间隙）、解体以及表面脱附。模拟的物理空间为任意形状的长方体。长方体的三个方向上的表面可设置为周期性边界，也可以设置为自由表面。当某个方向设置为周期性边界时，缺陷从该方向上的一

个面扩散出去后，会从另一个面的对应位置返回模拟盒子；当该方向被设置为自由表面时，缺陷扩散至表面之后则会被移出体系。在本 OKMC 软件中，所有的点缺陷以及缺陷团簇都被近似为具有一定作用半径的球体。当两个缺陷之间的距离小于其作用半径之和时，两个缺陷将结合成为一个新的缺陷，其成分为两个旧缺陷成分之和（空位与自间隙相互抵消）。

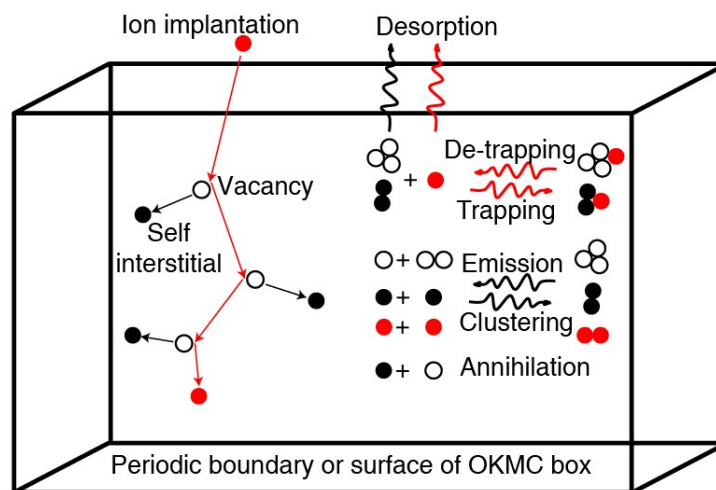


图 2 离子辐照环境下，OKMC 模拟中关键物理过程示意图

图 2 给出了 OKMC 程序的具体实现算法。具体而言，程序开始运行后，首先读取上级多尺度模拟给出的缺陷参数数据（一般由 DFT 计算给出）和辐照损伤数据（一般由 BCA 模拟给出），并根据模拟需求给出体系中的缺陷初始分布，建立元胞链表和速率二分查找树数据结构用于加速空间搜索和随机事件选择效率。随后，按照需求设定模拟的宏观条件，包括温度、时间、离子辐照通量。按照方程 1-5 所述的方法随机选择物理过程，执行并更新模拟时间、缺陷构型。重复以上过程直至完成所需的模拟时间为止。

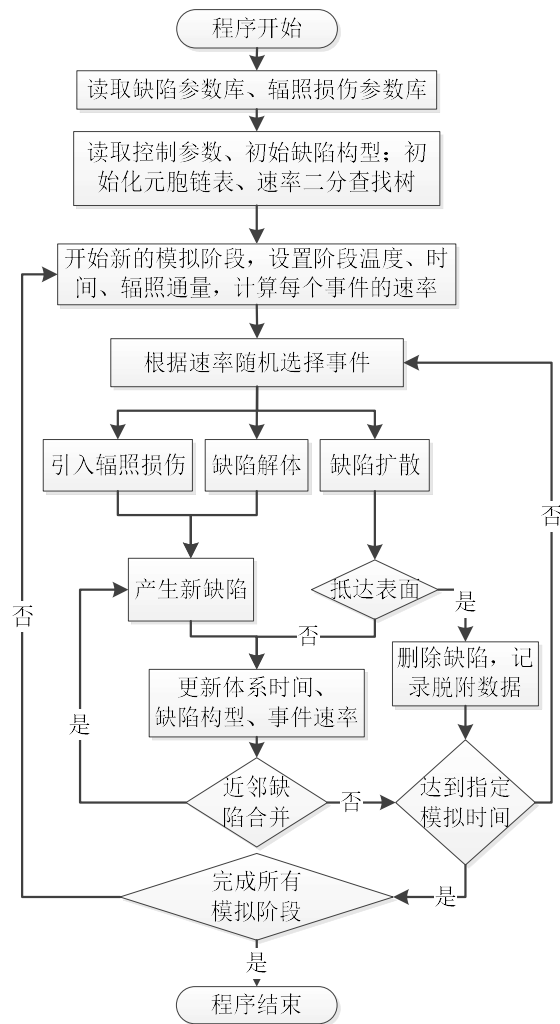


图 3 OKMC 程序算法示意图

1.3 元胞链表加速算法

在 OKMC 模拟中，当体系内某个缺陷构型发生变化时（例如扩散一步，或增删缺陷），需要对搜索该缺陷附近的其他缺陷，计算缺陷间的距离是否足够近，以至于发生结合反应。如图 4a 所示，实现这一目的最简单算法，便是逐一计算目标缺陷到体系中其他所有缺陷的距离。显然，这种算法的计算量近似的正比于总缺陷对象数目 N ，时间复杂度为 $O(N)$ 。当 N 值很大时（部分模拟中 N 可达百万量级），基于以上算法的缺陷近邻搜索会消耗大量的时间，极大的降低 OKMC 模拟效率。

针对以上问题，常见的解决方案便是元胞链表（cell lists）算法。如图 4b 所示，我们首先将模拟盒子分割为若干个小方块，这些小方块称为元胞。然后根据每个缺陷的坐标位置，建立起缺陷与元胞网格之间的空间映射关系（即每个元胞包含哪几个缺陷）。当需要对目标缺陷进行近邻搜索时，我们只需要遍历目标缺陷所在元胞和其近邻元胞内其他缺陷即可，无需对整个体系进行搜索。除了程序启动时，需要初始化所有缺陷与元胞之间的空间映射关系之外，此后某个缺陷构型发生改变时（增删缺陷，或缺陷跨越元胞边界），只需要更新该缺陷的映射关系即可。因此，元胞链表算法进行一次近邻搜索的计算量与缺陷总数无关，其时间复杂度为 $O(1)$ 。在缺陷对象数 N 较大时，元胞链表算法能极大提高 OKMC 的近邻搜索效率。

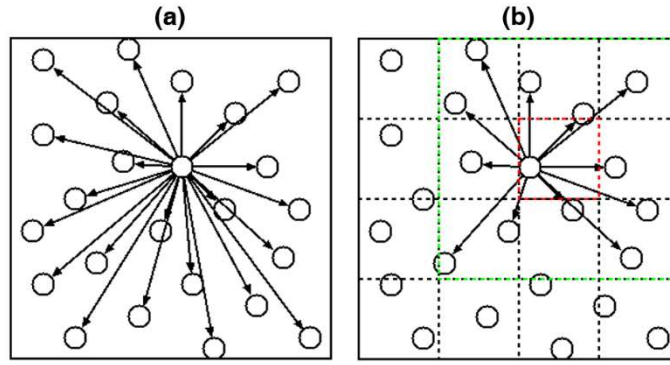


图 4 近邻缺陷搜索示意图：(a) 计算所有缺陷到目标缺陷的距离；(b) 仅仅计算目标缺陷所在元胞（红色格子）和近邻元胞（绿色格子）内缺陷到目标缺陷的距离。

此外，在元胞链表法中，网格划分越细，需要遍历的近邻缺陷就越少，效率就越高。但是网格边长的必须大于体系中最大的俘获半径 $\times 2$ ，否则在进行近邻搜索时，可能会漏判原本可能发生结合反应的两个缺陷。例如图 5a，左边的大黑圆为新加缺陷，红色代表会被遍历的缺陷，网格太细时，近邻 9 个网格不足以覆盖到右边大缺陷的中心，两个大缺陷之间发生跨网格重叠，因此遍历时便漏判了。因此，在类似空位不断聚合长大成大孔洞这样的模拟中，需要设置很粗的网格，这样的效率是较低的（图 5b）。这种情况下，可以划分一粗一细两套并行的元胞网格，并将细网格边长设为临界直径。小于该直径的圆仅被细网格索引，大于该直径的圆仅仅被粗网格索引。在添加新圆时，需要遍历两套网格中的近邻圆。尽管遍历的格子数增加了，但实际需要遍历的圆的数目是降低了（图 5c）。

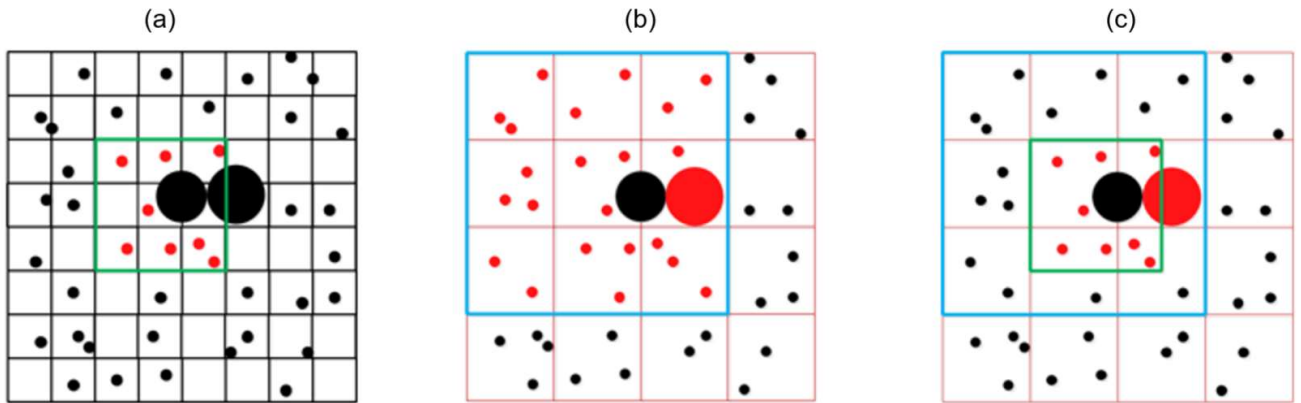


图 5 双重元胞链表算法示意图

1.4 二分查找加速算法

除了近邻搜索外，OKMC 中另一个十分耗时的操作便是动力学事件的随机选择。按照方程 4 所述，每个事件被选中的概率等于事件速率除以总速率。在计算机中实现这个过程，可以将所有的事件排列成一个区间列表（见图 5），并使其区间长度正比于事件的速率大小。然后，用一个随机数乘以总速率，得到一个随机飞镖。将这个飞镖随机射向事件区间列表，飞镖落在哪个区间内，该区间对应的事件就被选中。通常，需要从事件区间列表的头部开始，逐一判断事件是否被飞镖击中（图 5 下半部分）。这种线性查找算法的计算量正比于事件数目（正比于缺陷对象数 N ），算法时间复杂度为 $O(N)$ 。当 N 值很大

时，该算法效率非常低。

在本项研究中，我们通过建立二分查找树来加速以上事件选择过程。如图 5 上半部分所示，我们将每个事件区间视为一片树叶，并将两片树叶速率之和作为上级树枝节点的速率，依此类推至根节点。若事件总数不足 2 的整数次幂，不足的树叶由 0 速率事件填补。建立好这样一株二分查找树之后，我们仍然用一个随机数乘以总速率，得到一个随机飞镖。将这个随机飞镖与根节点下的两个节点对比，判断飞镖是击中左边还是右边的树枝，并依此类推直至击中树叶节点，然后选中相应的事件。当事件区间发生变化时（例如增删缺陷导致事件数的增减），只需要更新发生变化的树叶节点及其所有上级节点即可，无需更新整个二分树。通过二分查找树随机选择事件所需的计算量正比于二分查找树的高度，算法时间复杂度为 $O(\log_2 N)$ 。因此，当 N 较大时，二分查找算法能极大提高 OKMC 模拟中的事件选择效率。

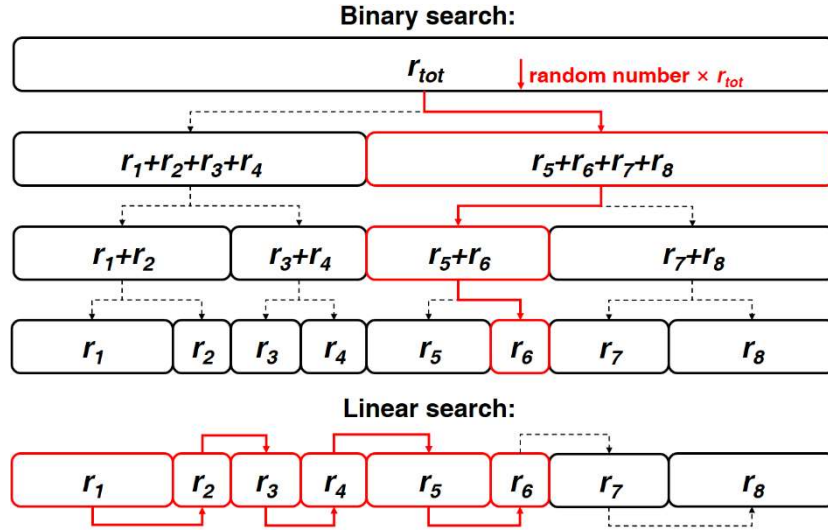


图 5 OKMC 事件随机选择的查找路径示意图。上半部分为二分查找算法，下半部分为线性查找法。

1.5 等效晶界算法

晶界是缺陷的重要吸收阱。本软件虽然可以直接将盒子表面设为吸收阱以模拟长方体晶界，但由于缺陷总量与体积成正比，因此很难处理比较大的晶界。为此，可以在模拟中添加一个隐性的晶界，该晶界在模拟空间中并不存在一个显性的实体，而是隐性作用于所有缺陷上，每次缺陷进行扩散时，都有一定概率被晶界吸收。该算法基于缺陷扩散的吸收强度理论[3]，吸收强度的定义为：

$$k^2 = \frac{2n}{d^2 \langle m \rangle} \quad (6)$$

其中 n 为扩散的空间维度， d 为缺陷扩散步长， $\langle m \rangle$ 为缺陷在被吸收阱捕获前的平均扩散次数。而球形晶界对一维、三维扩散缺陷的吸收强度为[4]：

$$k_{1,g}^2 = \frac{5}{R^2}; \quad k_{3,g}^2 = \frac{14.4}{R^2} \quad (7)$$

其中 R 为晶粒半径。联合公式 6-7，可以给出

$$\langle m \rangle_1 = \frac{6}{15} \left(\frac{R}{d} \right)^2 ; \langle m \rangle_3 = \frac{6}{14.4} \left(\frac{R}{d} \right)^2 \quad (8)$$

考虑到二者差距不大，在本软件中，我们近似采用晶界一维扩散缺陷的吸收强度，并根据上式给出每次扩散时缺陷被晶界吸收的概率：

$$p = \frac{1}{\langle m \rangle_1} = \frac{15}{6} \left(\frac{d}{R} \right)^2 \quad (9)$$

二、程序安装

本程序为串行程序，可在 windows 和 linux 下运行。请尽量在以下建议的配置环境中编译程序，其他编译环境尚未做过仔细测试。

Windows 环境：

使用 visual studio + intel fortran compiler。并将实型变量的默认值设置为双精度,步骤为：Project-> 'Project_name' properties -> Fortran -> Data -> Default real kind -> 8

选择 Release 以及 x64 位模式编译链接产生 exe 文件。将产生的可执行文件与输入文件（见下文介绍）置于同一目录下，双击 exe 文件可运行程序

若在 Windows 下运行时提示缺失 dll 错误,可以在 Project > Properties > Config Properties > Fortran > Libraries > Runtime Library, 设置为 Multithread, 然后重新生成 exe 文件，并尝试更新 Intel® Fortran Compiler Runtime for Windows:

<https://www.intel.com/content/www/us/en/developer/articles/tool/compilers-redistributable-libraries-by-version.html>

Linux 环境

使用 ifort 编译，先编译全局变量模块（default_setting.f90 和 global.f90），再编译其他源代码。参考 makefile:

fc = ifort

target = okmc.x

source1 = *.f90

source0 = default_setting.f90 global.f90

object = *.o

\$(target): \$(source1)

\$(fc) -c \$(source0)

\$(fc) -c \$(source1)

\$(fc) -O2 -o \$(target) \$(object)

clean:

rm *.o *.mod *.x

三、程序使用

3.1 基本流程

- 1、 准备 INPUT 文件，设置盒子尺寸、元胞网格、初始缺陷浓度等
- 2、 准备 CONTROL 文件，设置温度、时间、辐照通量、IO 参数
- 3、 准备 PARA 文件，定义缺陷的动力学参数
- 4、 准备 POSITION 文件，给出初始缺陷构型（可选）
- 5、 准备 aic.xyz.cfg 或 ION/SIA/VAC 文件，给出辐照初级损伤数据（可选）

3.2 输入文件

输入文件包括 INPUT.txt、CONTROL.txt、POSITION.txt(或 POSITION.Imp)、PARA.txt、aic.xyz.cfg（或 ION.txt、VAC.txt、SIA.txt）。文件名大写敏感。

注意，POSITION 文件、aic.xyz.cfg（或 ION.txt、VAC.txt、SIA.txt）这几个输入文件为可选项，若用不上可以不提供。程序未找到 POSITION 文件时，会跳过读取初始构型的步骤。未找到 aic.xyz.cfg 以及 ION.txt、VAC.txt、SIA.txt 时，辐照功能将会关闭，此时即便 CONTROL 文件中设置了不为 0 的辐照通量，也不会产生任何辐照。

在输入文件中，一般以#号作为注释符号，每一行#号后面的内容不会被程序读取，空行视为无效行。

3.2.1 INPUT.txt 文件

INPUT 文件包含了程序运行所需的基本参数，包括周期性边界条件的设置、球形晶界边界条件的设置、元胞列表的划分方式、初始缺陷的浓度和类型等。

INPUT 文件中的参数应按以下格式给出：

variable name=variable value

Variable name 为输入的参数名称（参数列表见下表），大小写均可，但不可大小写混合。variable value 为参数的输入值，当 variable value 为数组类型的输入值时，不同的值之间应以空格或制表符分开。INPUT 中的部分参数都有默认值，若不进行设置，程序会自动取默认值。

Variable 列表

| variable name | 类型 | 默认值 | 说明 |
|---------------|-------|-------|---|
| PBC | 逻辑型*3 | 1 1 0 | 三个方向的周期性边界条件，1 表示该方向采用周期性边界条件，0 表示该方向采用自由表面边界条件 |
| grain_radius | 正值*1 | 1e15 | 等效晶界模型中的晶粒半径，缺陷在每次扩散中都有一定几率被 grain 吸收，跟据 sink strength 计算吸收概率，每次扩散的吸收概率为 $2.5(d/\text{grain_radi})^2$ |
| box_length | 正值*3 | 0 0 0 | 设置盒子三个维度的边长。 注意盒子尺寸有多种设置方式，按照优先级依次为： (1) 根据 box_length 参数设置盒子尺寸。 (2) 读取 POSITION.Imp 文件中的盒子尺寸。 (3) 根据 cell_size*cell_number 参数设置盒子尺寸。 当 (1) 或者 (2) 设置生效时，程序会忽略 INPUT 中设置的 cell_size、 |

| | | | |
|-------------------|-------|-----------------------------|---|
| | | | cell_number、large_cell_size、large_cell_number 四个参数, 并自动生成大小元胞网格。自动生成的大元胞边长在 20 左右, 数量为 box_length/20, 向下取整。对特别大的盒子尺寸、程序会进一步粗化上述格子, 确保内存不溢出。 |
| cell_size | 正值*3 | 10 10 10 | 单个元胞的长宽高 |
| cell_number | 正整数*3 | 0 0 0 | 三个维度上的元胞个数, 分别乘上 cell_size 后得到模拟盒子的长宽高。 |
| large_cell_size | 正值*3 | 20 20 20 | 单个大大元胞的长宽高, 注意该元胞的边长的必须大于体系中最大的俘获半径 x2, 否则可能发生近邻缺陷的漏判。 |
| large_cell_number | 正整数*3 | 0 0 0 | 三个维度上的大元胞个数, 分别乘上 large_cell_size 后得到模拟盒子的长宽高。注意这个长宽高应与小元胞计算出来的盒子长宽高一致。 |
| surface_depth | 正值*1 | 3.1652 | <p>将盒子表面到 surface_depth 深度处的空间视为表面吸附阱, 用于阻断周期性边界条件, 实现部分维度的周期性。例如, 当 PBC=1 1 0 时, 在盒子的第 3 个维度的两个内表面上, 会出现两个厚度为 surface_depth 的表面吸附阱。缺陷扩散至这两个表面时便会被吸附阱吸收。而不是像正常周期性边界条件下一样, 越过盒子边界后从另一边出来。</p> <p>surface_depth 的值可取任意正实数, 该值应大于体系中最大的缺陷迁移步长。</p> <p>注意, 应用部分周期性边界条件时, 在不采用周期性的维度上, 盒子的有效长度会比实际长度小 2*surface_depth, 若对盒子的大小要求比较严格, 在设置盒子长宽高时应考虑到这一点并增加补偿量</p> |
| critical_radius | 正值 | $\min(\text{cell_size})/2$ | 判断缺陷索引于小元胞还是大元胞的判据。小于 critical_radius 的缺陷团簇仅会被小元胞列表索引到, 大于 critical_radius 的则仅会被大元胞列表索引到 |
| uniform_damage | 逻辑型*3 | 1 1 0 | <p>辐照级联在三个维度均匀分布情况, 取值为三个 0 或 1 的整数, 均匀分布为 1, 此时会在该辐照级联的基础上添加一个随机数, 实现辐照点的随机分布, 不影响级联内部的空间关联。否则为 0, 按照辐照数据库的入射点设置辐照位置。</p> <p>例如对 Z 方向注入的离子辐照, 建议设置为 1 1 0。</p> |
| ion_type | 正整数*1 | 2 | 注入离子类型, 取值范围 1-4, 默认为 2, 即注入离子设为第 2 类缺陷 |
| iso_eff | 逻辑型*1 | 0 | 同位素效应, 0/1 表示关闭/开启。当开启时, 会将第 2、3 类缺陷视为同位素 |
| implant_direction | 正整数*1 | 3 | 辐照注入方向, 取值范围 1-3, 对应 3 个维度的方向。注意, 对离子辐照而言, 这个方向必须与你的辐照损伤数据库的注入方向一致, 具体内容见 3.2.5 小节。 |
| initial_defect | 正值*4 | 0 0 0 0 | 初始添加的缺陷浓度列表, 单位为 appm, 其中第一个代表本征缺陷 (VAC/SIA), 其他代表异质缺陷 |
| intrinsic_type | 整数*1 | 0 | 初始添加本征缺陷的类型, 取值范围为 -1/0/1。-1 代表仅添加指定浓度 |

| | | | |
|--------------|-------|-----------|--|
| | | | 的 SIA, 1 代表仅添加指定浓度的 VAC, 0 代表同时添加指定浓度的 VAC 和 SIA |
| lattice_type | 字符串 | bcc | 支持 bcc 与 fcc 两种结构, 该变量会影响: 1) initial_defect 生成的数量。假如 a0 相同, 由于 bcc 和 fcc 单胞内原子数量相差 1 倍, 根据原子浓度计算得到的缺陷数量也会相差一倍。 2) 团簇的迁移方向, bcc 中迁移方向为 8 个<111>方向之一, fcc 中为 12 个<110>方向之一。 |
| a0 | 正值*1 | 3.1652 | 晶体的晶格常数 |
| cfg_type | 字符串 | txt | 输入/输出结构文件的格式, 可取 txt 或 Imp 两种格式。程序的输入 (POSITION 文件, 见下文) 和输出会跟据该变量自动调整格式 |
| damage_type | 字符串 | txt | 离子辐照数据库文件的格式, 可取 txt 或 cfg 两种格式。取值为 txt 时, 离子辐照数据库由 ION.txt、VAC.txt、SIA.txt 三个文件提供。取值为 cfg 时, 由 aiv.xyz.cfg 文件提供, 具体内容见 3.2.5 小节。 |
| rd_seed | 正整数*4 | 与时间有关的随机值 | 随机数种子, 手动设置该参数可使模拟结果具有可重复性, 部分编译器只需要 2 个种子, 此时只有前两个数生效 |
| output_rate | 逻辑型*1 | 0 | 缺陷构型输出时是否添加速率信息, 默认为 0, 若设为 1, 则会额外输出团簇的单向迁移、转向迁移、发射点缺陷的动力学速率。 |

3.2.2 CONTROL.txt 文件

CONTROL 文件为程序输入模拟的流程信息, 包括各阶段的温度、时间、辐照速率等。

该文件中, 每一个有效行都是一个模拟单元。每个单元由 温度、时间、辐照速率、单元名称、输出类型 这五个参数描述。通过多个模拟单元的拼接实现复杂的模拟流程。

温度: 即该单元的模拟温度, 单位为 K, 在同一单元内温度保持不变。

时间: 单元的模拟时间, 单位为 s

辐照速率: 表示辐照的通量, 单位为 ION/m²/s。

单元名称: 用以方便区分不同的模拟单元, 并用于命名构型输出文件。

输出类型: 取值范围为 0-5。用以判定在单元模拟结束时是否输出详细的构型文件。取 0 时不输出, 取 1 时输出所有缺陷团簇, 取值为 2/3/4/5 时, 仅输出 f1/f2/f3/f4 组分不等于 0 的缺陷团簇。例如, 当你只关心第 2 类缺陷的演化时, 可以将该行的对应参数设为 3, 此时程序只会输出包含第 2 类缺陷的团簇。

以下是一个 CONTROL 文件的示意:

| CONTROL.txt | | | | | |
|-------------|-----|--|--------|-------------|--------|
| 1 | #模拟 | 的流程控制矩阵, 用若干参数描述一个模块的特征, 多个模块拼接实现复杂流程的模拟 | | | |
| 2 | #温度 | 时间 | 通量 | 模块名称 | 是否输出构型 |
| 3 | 0 | 0 | 0 | initiate | 1 |
| 4 | 800 | 1000 | 1.0e18 | irradiation | 1 |
| 5 | 795 | 60 | 0 | cool_795 | 0 |
| 6 | 790 | 60 | 0 | cool_790 | 0 |
| 7 | 785 | 60 | 0 | cool_785 | 0 |
| 8 | 780 | 60 | 0 | cool_780 | 0 |
| 9 | 775 | 60 | 0 | cool_775 | 0 |
| 10 | 770 | 60 | 0 | cool_770 | 0 |

其中 2-10 行定义了 8 个模拟单元, 第一个模拟单元 initiate 时间为 0, 一般用于输出程序的初始构型。第二个单元

irradiation 模拟 800K 温度、 1.0×10^{18} 通量的离子辐照下，体系在 1000 秒内的演化过程。第 3-8 个模拟单元以固定的降温速率降低温度，用于模拟辐照后的降温过程。程序会自上而下执行这 8 个模拟单元，并在每个模拟单元结束时，根据最后一列的取值输出当时的体系构型。

3.2.3 POSITION.txt 或 POSITION.lmp 文件

用于在模拟最开始时载入的自定义缺陷构型文件，包含每个缺陷的坐标和组分信息。若无需自定义初始构型，则该文件内容留空。模拟单元的输出详细构型文件重命名后，可直接用做 POSITION 文件。实现断点续算或多个任务的拼接。

跟据 cfg_type 的设置，POSITION 文件有 txt 和 lmp 两种格式。

Txt 格式不包含盒子信息，只给出每个缺陷的信息，示例如下：

| POSITION.lmp | POSITION.txt |
|--------------|--|
| 1 | 987.339401 439.862157 636.101435 0 0 1 0 5 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 2 | 201.837714 89.910013 109.217649 0 1 0 0 8 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 3 | 721.311900 240.772993 582.918474 0 1 0 0 1 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 4 | 555.632295 857.488392 475.978266 0 1 0 0 1 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 5 | 641.688680 916.753296 449.782143 0 1 0 0 1 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 6 | 280.450803 315.891761 372.845669 0 1 0 0 6 1.560 0.697E+02 0.139E+03 0.000E+00 |

其中每行代表一个缺陷团簇的信息，每个有效行包含信息为：

X Y Z f(1) f(2) f(3) f(4) orientation radius rate_mig rate_rot rate_emit

其中 XYZ 为缺陷缺陷团簇的坐标，f(1:4)为缺陷缺陷团四类点缺陷的个数，orientation 为缺陷当前的迁移取向（某些缺陷，如钨中的 SIA，只能 1D 或 1D/3D 混合迁移），radius 为缺陷捕获半径，rate_mig、rate_rot、rate_emit 为缺陷的单向迁移、转向迁移、发射点缺陷的动力学速率。

在读取，radius rate_mig rate_rot rate_emit 四个参数会被忽略，可以不设置。

lmp 格式参考 lammps 的 custom 输出格式，例如：

| POSITION.lmp | POSITION.txt |
|--------------|--|
| 1 | ITEM: TIMESTEP |
| 2 | 0 |
| 3 | ITEM: NUMBER OF ATOMS |
| 4 | 631 |
| 5 | ITEM: BOX BOUNDS pp pp pp |
| 6 | 0.000000E+00 1000.0000000000 |
| 7 | 0.000000E+00 1000.0000000000 |
| 8 | 0.000000E+00 1000.0000000000 |
| 9 | ITEM: ATOMS x y z f1 f2 f3 f4 orient radius rate_mig rate_rot rate_emit |
| 10 | 987.339401 439.862157 636.101435 0 0 1 0 5 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 11 | 201.837714 89.910013 109.217649 0 1 0 0 8 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 12 | 721.311900 240.772993 582.918474 0 1 0 0 1 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 13 | 555.632295 857.488392 475.978266 0 1 0 0 1 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 14 | 641.688680 916.753296 449.782143 0 1 0 0 1 1.560 0.697E+02 0.139E+03 0.000E+00 |
| 15 | 280.450803 315.891761 372.845669 0 1 0 0 6 1.560 0.697E+02 0.139E+03 0.000E+00 |

注意，lmp 格式下可以设置盒子大小，但程序会优先根据 INPUT 文件中的 box_length 参数设置盒子大小。只有当 INPUT 文件中未对 box_length 参数进行设置时，才会根据 POSITION.lmp 文件中 6-8 行的盒子参数进行设置。

3.2.4 PARA.txt 文件

对 OKMC 程序而言，各个缺陷的参数化是一个麻烦的问题。一方面，由于 OKMC 程序并不具有计算参数的能力，这些参数往往需要依赖其他方法（DFT/MS 等）获得。另一方面，程序运行过程中可能会涉及到成千上万种不同的缺陷，完全依靠手动输入这些数据是很繁琐的，更何况依赖 DFT 等方法获得如此巨量的参数也并不现实。因此，常见的做法是：利用 DFT 等方法精确计算点缺陷及小型缺陷团簇的参数，并在此基础上，拟合推测大型缺陷团簇的参数[5-7]。

在本程序中，缺陷团簇的所有参数都记录在 PARA.txt 文件中，例如：

```

1  -1  2
2  0  2
3  0  1
4  0  0
5  #本征缺陷-杂质-H体系参数库
6
7  #formula      Em      Em+Er      Em+Eb      radius  ve      vm      emit      sto
8  #I/V
9  -1  0  0  0  0  0.02  0.40  100.00  1.5762  6.10E+12  2.44E+12  -1  2.7411
10  0  0  0  0  0  100.00  100.00  100.00  0.0000  0.00E+00  0.00E+00  1  0.0000
11  1  0  0  0  0  1.66  1.66  100.00  1.3706  6.10E+12  6.10E+12  1  2.7411
12  2  0  0  0  0  1.66  1.66  1.72  1.7756  7.69E+12  6.10E+09  1  1.3706
13  #H
14  0  1  1  0  0  0.28  0.25  100.00  3.6371  2.58E+13  8.06E+11  2  6.3304
15  0  2  0  0  0  0.22  0.25  0.00  2.5820  3.24E+13  0.00E+00  2  3.1652
16  #I/V-H
17  1  1  1  0  0  100.00  100.00  1.44  1.3706  5.05E+13  0.00E+00  2  0.0000
18  1  2  0  0  0  100.00  100.00  1.44  1.3706  6.36E+13  0.00E+00  2  0.0000
19  2  1  0  0  0  100.00  100.00  1.64  1.7756  5.05E+13  0.00E+00  2  0.0000
20  2  2  0  0  0  100.00  100.00  1.67  1.7756  6.36E+13  0.00E+00  2  0.0000
21  -2  1  0  0  0  100.00  100.00  0.38  2.0420  5.05E+13  0.00E+00  2  0.0000
22  -2  2  0  0  0  100.00  100.00  0.38  2.0420  6.36E+13  0.00E+00  2  0.0000
23  #I/V-IMP
24  -1  0  1  1  0  100.00  100.00  0.00  1.5762  6.10E+12  0.00E+00  3  0.0000
25  0  0  1  0  0  100.00  100.00  100.00  2.6000  0.00E+00  0.00E+00  3  2.7411
26  1  0  1  1  0  100.00  100.00  0.00  1.3706  6.10E+12  0.00E+00  3  0.0000
27  2  0  1  0  0  100.00  100.00  0.00  1.7756  6.10E+12  0.00E+00  3  0.0000
28  #H-IMP
29  0  1  1  1  0  100.00  100.00  0.89  1.3706  2.58E+13  0.00E+00  2  0.0000
30  0  2  1  1  0  100.00  100.00  0.80  1.3706  2.58E+13  0.00E+00  2  0.0000
31  #I/V-H-IMP
32  1  1  1  1  0  100.00  100.00  0.00  1.3706  5.05E+13  0.00E+00  3  0.0000
33  1  2  1  1  0  100.00  100.00  0.00  1.3706  6.36E+13  0.00E+00  3  0.0000
34  2  1  1  1  0  100.00  100.00  0.00  1.7756  5.05E+13  0.00E+00  3  0.0000
35  2  2  1  1  0  100.00  100.00  0.00  1.7756  6.36E+13  0.00E+00  3  0.0000
36  -1  1  1  1  0  100.00  100.00  0.00  1.5762  5.05E+13  0.00E+00  3  0.0000
37  -1  2  1  1  0  100.00  100.00  0.00  1.5762  6.36E+13  0.00E+00  3  0.0000

```

PARA 文件的格式为:

第 1-4 行为声明行，分别声明本文件所描述缺陷的四个组分取值范围，每行内容为两个整数。这四行中不允许有注释。例如上图的示意中声明了一共 24 种不同组分的缺陷 object，其组分列表为 (-1:2,0:2,0:1,0:0) 的一个 4x3x2x1 四维矩阵。

剩余有效行为缺陷参数定义行，允许注释。每一行定义一种缺陷的参数信息，每一有效行的格式为：

f(1) f(2) f(3) f(4) Em Er+Em Edetrar ve vm emit step

其中, $f(1):f(4)$ 为该缺陷团簇包含的四种点缺陷个数, $f(1)$ 为本征缺陷, 负值代表自间隙, 正值代表空位, 其余三种点缺陷可自行定义。 E_m 为团簇迁移能垒, E_r+E_m 为团簇的转向迁移能垒, E_{detrap} 为点缺陷从团簇中解离出去的最小能垒, r 为团簇的俘获半径, v_e 为解离的尝试频率, v_m 为迁移尝试频率, e_{mit} 为最小解离能对应的解离组分 (取值范围为 $1\sim 4$), step 为团簇迁移步长。

当 isoeff 设置为 1 时，开启同位素效应功能，此时会将 2、3 类缺陷视为同位素。缺陷团簇的参数按照 $f(2)=f(2)+f(3)$ ， $f(3)=0$ 设置，无序设置 $f(3)$ 对应的参数。发射缺陷时，随机在 2、3 类缺陷中抽选一个发射，概率正比于 2、3 类缺陷的数量。

注意， PARA 文件中一般需要包括模拟中涉及所有的缺陷。如果模拟中生成了超出前 4 行声明范围的缺陷，或者在缺陷尺寸在声明范围内，但没有被定义，该缺陷会被设置成不稳定缺陷自动解体，并发射团簇内含量最多的点缺陷。

3.2.5 ION/VAC/SIA.txt 或 aiv.xyz.cfg 文件

离子辐照数据库文件，根据 damage_type 参数的取值可选择 txt 或 cfg 两种格式。取值为 txt 时，离子辐照数据库由 ION.txt、VAC.txt、SIA.txt 三个文件提供。取值为 cfg 时，由 aiv.xyz.cfg 文件提供。

aiv.xyz.cfg 文件由 IM3D 软件直接输出，记录了 BCA 离子辐照模拟给出的离子停留坐标，以及产生的自间隙原子、空位原子坐标。文件格式为 AtomEye 软件的 cfg 格式，以下是一个 aiv.xyz.cfg 文件的示例：

```

aiv.xyz.cfg
1 Number of particles = 306173
2 A = 10 Angstrom (basic length-scale)
3 H0(1,1) = 10000 A
4 H0(1,2) = 0.0 A
5 H0(1,3) = 0.0 A
6 H0(2,1) = 0.0 A
7 H0(2,2) = 10000 A
8 H0(2,3) = 0.0 A
9 H0(3,1) = 0.0 A
10 H0(3,2) = 0.0 A
11 H0(3,3) = 10000 A
12 .NO_VELOCITY.
13 entry_count = 7
14 auxiliary[0] = type (A-I-V)
15 auxiliary[1] = tab (A-ion)
16 auxiliary[0] = mater
17 auxiliary[1] = element
18 4
19 aiv_pos
20 0.501472 0.499672 0.009447 1 0 0 0
21 0.501539 0.499630 0.009407 2 0 0 0
22 0.507223 0.497102 0.012853 1 0 0 0
23 0.507212 0.497069 0.012827 2 0 0 0
24 0.510954 0.490892 0.015272 1 0 0 0
25 0.510982 0.491045 0.015176 2 0 0 0
26 0.510996 0.491016 0.015138 1 0 0 0
27 0.510989 0.491089 0.015153 2 0 0 0
28 0.516579 0.493441 0.014202 1 0 0 0
29 0.516554 0.493433 0.014228 2 0 0 0
30 0.514886 0.493270 0.014028 0 0 -1 -1
31 0.499695 0.499924 0.003126 1 1 0 0
32 0.499667 0.499920 0.003159 2 1 0 0

```

其中，第 1 行给出了缺陷总数，第 2 行为盒子边长单位，3-11 行为盒子三个基矢的坐标（注意这里只支持正交盒子），12-18 行为 auxiliary 信息。

第 20 行开始每行代表一个点缺陷，其中前三列为缺陷的分数坐标，第四列为缺陷类型（0/1/2 分别表示注入离子/自间隙/空位），第五列为产生该缺陷的注入离子编号（一个离子编号产生一个离子，可能不产生自间隙和空位，也可能产生多个自间隙和空位）。第 6-7 列信息不会被 OKMC 程序读取。

若 damage_type 参数的取值为 cfg，则需要按照上述所示的格式提供一个 aiv.xyz.cfg 文件（可以用 IM3D 软件直接输出）。程序会将 aiv.xyz.cfg 文件转换为 ION.txt、VAC.txt、SIA.txt 三个文件分别记录离子、自间隙、空位的坐标。若 damage_type 参数的取值为 txt，则需要预先提供上述三个 txt 文件（如果模拟不涉及辐照损伤，文件可以留空）。三个 txt 文件的具体格式如下。

ION.txt：离子注入后离子的坐标信息，格式如下图：

```

ION.txt
1 1 832.4000 50420.20 49649.40
2 2 717.3000 49683.30 51002.00
3 3 1341.800 50328.80 50031.70
4 4 1130.700 50813.60 49480.50
5 5 -10000 -10000 -10000
6 6 1333.700 50488.80 50115.70
7 7 832.9999 50711.70 49651.40
8 8 988.5000 50261.50 49929.60

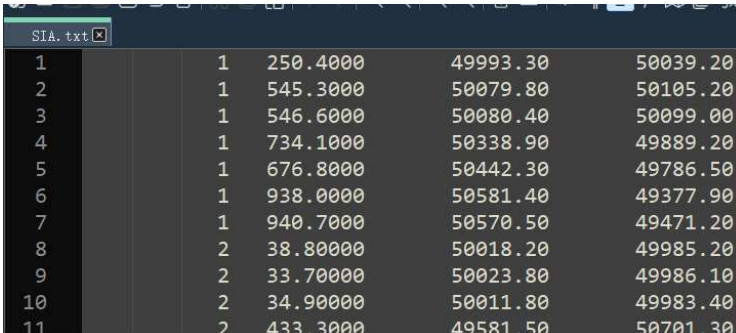
```

每个有效行的格式为：

Ion_number X Y Z

Ion_number 为离子编号，XYZ 为离子停留在靶材内的绝对坐标，若离子反弹离开靶材，XYZ 的值取一个较大的负值（如 -10000）

SIA.txt: 离子注入后 SIA 的坐标信息, 格式如下图:



| | | | | |
|----|---|----------|----------|----------|
| 1 | 1 | 250.4000 | 49993.30 | 50039.20 |
| 2 | 1 | 545.3000 | 50079.80 | 50105.20 |
| 3 | 1 | 546.6000 | 50080.40 | 50099.00 |
| 4 | 1 | 734.1000 | 50338.90 | 49889.20 |
| 5 | 1 | 676.8000 | 50442.30 | 49786.50 |
| 6 | 1 | 938.0000 | 50581.40 | 49377.90 |
| 7 | 1 | 940.7000 | 50570.50 | 49471.20 |
| 8 | 2 | 38.80000 | 50018.20 | 49985.20 |
| 9 | 2 | 33.70000 | 50023.80 | 49986.10 |
| 10 | 2 | 34.90000 | 50011.80 | 49983.40 |
| 11 | 2 | 433.3000 | 49581.50 | 50701.30 |

每个有效行的格式为:

lon_number X Y Z

lon_number 为产生该 SIA 的离子编号, XYZ 为 SIA 停留在靶材内的坐标, 若第 lon_number 号离子注入并未产生 SIA, 则 XYZ 的值取一个较大的负值 (如-100000)

VAC.txt: 与 SIA.txt 规则一致, 不再赘述。

这三个文件可以通过 IM3D 软件模拟给出 (当然, 需要进行格式转换), 也可以基于 MD 的 cascade 模拟给出, 甚至可以手动构建, 从而实现离子、中子、电子等不同类型的辐照。

请注意, 当进行离子辐照时, 一般需要手动设置 implant_direction、PBC、uniform_damage 三个参数。其中 implant_direction 需要与辐照损伤数据的实际注入方向一致, 并将注入方向的 PBC 和 uniform_damage 都设置为 0, 以模拟表面均匀的离子注入。例如, 假设通过 IM3D 软件模拟沿 Z 轴注入, 获得相应的 aiv.xyz.cfg 文件, 则需要设置 implant_direction=3, PBC=1 1 0, uniform_damage=1 1 0。

3.3 输出文件

输出文件包括 monitor.txt、defect_statistic.txt、以及按各个模拟单元名称命名的构型输出文件 unit_name.txt。

3.3.1 monitor.txt

monitor 文件为程序的监视文件和报警信息输出文件, 用于在程序运行时查看 INPUT、CONTROL 文件是否正确设置与读入。并且将显示模拟的单元进度与每个单元的 CPU 耗时。

3.3.2 defect_statistic.txt

每一行记录了每个模拟单元结束时团簇的:

温度、总团簇数、4 类点缺陷总数、上表面 (离子注入的表面) 释放的 4 类点缺陷总数、下表面释放的 4 类点缺陷总数、被晶界吸收的 4 类点缺陷总数、unit_name。

3.3.3 unit_name.txt/lmp

记录了该模拟单元结束时, 体系内所有缺陷的构型信息。其格式与 POSITION.txt/lmp 一致。此外, 若 INPUT 中 output_rate 为 1, 程序会额外输出缺陷的动力学速率信息。

参考文献:

- [1] G.H. Vineyard, Frequency Factors and Isotope Effects in Solid State Rate Processes, Journal of Physics and Chemistry of Solids 3(1-2) (1957) 121-127.
- [2] C.S. Becquart, C. Domain, U. Sarkar, A. DeBacker, M. Hou, Microstructural evolution of irradiated tungsten: Ab initio

parameterisation of an OKMC model, *Journal of Nuclear Materials* 403(1-3) (2010) 75-88.

[3] J. Hou, X.-S. Kong, X.-Y. Li, X. Wu, C.S. Liu, J.-L. Chen, G.N. Luo, Modification on theory of sink strength: An Object Kinetic Monte Carlo study, *Computational Materials Science* 123 (2016) 148-157.

[4] L. Malerba, C.S. Becquart, C. Domain, Object kinetic Monte Carlo study of sink strengths, *Journal of Nuclear Materials* 360(2) (2007) 159-169.

[5] J. Hou, X. Kong, W. Hu, H. Deng, D. Nguyen-Manh, J. Song, Deuterium trapping and desorption by vacancy clusters in irradiated Mo from object kinetic Monte Carlo simulations, *Acta Materialia* 274 (2024).

[6] J. Hou, Y.-W. You, X.-S. Kong, J. Song, C.S. Liu, Accurate prediction of vacancy cluster structures and energetics in bcc transition metals, *Acta Materialia* 211 (2021) 116860.

[7] J. Hou, X.-S. Kong, X. Wu, J. Song, C.S. Liu, Predictive model of hydrogen trapping and bubbling in nanovoids in bcc metals, *Nature Materials* 18 (2019) 833–839.