# 小组作业报告

小组成员：田鑫涛、俞泽榕、楚昊锟

选择题目 project 15.45

**摘要：**本次大作业旨在模拟二元合金在快速淬火时所发生的旋节分解过程。Lifshitz 和 Slyozov 预测在该过程中，线性域尺寸随时间的增长关系为 $R\sim t^{1/3}$, 且对于任意大于等于 2 的维度，该结论成立。本小组跟随题目的四小问，逐步建模并模拟旋节分解过程的二维及三维结果，通过两种方法分别求解了域尺寸随时间的增长规律，并给出温度等参量的影响。

**关键词：**旋节分解 合金淬火 蒙特卡洛模拟

## 1、简介

**元合金淬火**是一种金属热处理工艺。对于元合金（由单一金属元素组成的合金，不过通常更常见的是二元合金、三元合金等多元合金的表述），淬火过程通常包括将合金加热到特定温度并保持一段时间，使合金内部组织均匀化，然后迅速冷却。

在加热阶段，合金中的原子获得足够能量进行扩散，可能改变合金的相结构。快速冷却时，由于冷却速度极快，合金中的原子来不及扩散回到平衡态位置，从而形成非平衡态的组织，这种组织往往具有较高的硬度和强度，但韧性可能会有所下降。例如在钢铁淬火过程中，加热后快速冷却可以得到马氏体组织，具有高硬度的特点。不同的元合金根据其自身特性，在淬火温度、冷却介质选择等方面都有所不同。

**旋节分解**是过饱和固溶体中因扩散效应引发的相分离过程。它分两阶段，先由浓度波动产生新相核并生长，后以大晶粒吞并小晶粒的合并过程为主。其晶粒半径变化率与溶质浓度等相关，存在临界半径决定晶粒生长或溶解。随着时间，晶粒尺寸分布趋近特定函数，临界尺寸、过饱和度和晶粒数量均有其相应变化规律，且各向异性、晶体有序性和弹性应变等因素可通过有效参数考虑其影响，在空位过饱和晶体中有相关的应用实例。

## 2、问题详解

根据题目设问，可以将问题总结成以下几点：

（1） 如何设置 AB 两种原子及空穴随机分布的初态？为什么这对应的无限温度的情形？

（2） 如何通过关联函数和能量公式的两种方法来估算域的生长大小？两种方法给出的结果一致吗？二维域生长与时间之间的幂律关系是怎样的？

（3） 改变温度，对域生长的演化过程的影响是什么？

（4） 如果二维改为三维情形，幂律关系是否保持？

## 3、 模型算法

### （1）Ising-model

对于二元合金旋节分解过程的模拟，一般可以通过 Ising 自旋交换模型来实现。Ising 模型哈密顿量可以由

$$H = J\sum_{i,j} S_i S_j$$

给出，是否翻转自旋则是通过 Metropolis 算法来实现，具体思路为：

- 计算能量变化：随机选择一个格点，与空穴交换位置后，计算前后能量变化$\Delta E$；
- 计算接受概率：根据 Metropolis 算法，接受概率$P = \min(1, e^{\Delta E/(k_B T)})$，其中$k_B$是玻尔兹曼常数（在数值计算中可以取为 1，因为它只影响能量和温度的相对尺度）。
- 决定是否接受自旋翻转：生成一个在$[0,1]$区间内均匀分布的随机数$r$，如果$r < P$，则接受交换；否则，不进行交换。

与传统自旋翻转不同的是，本问题需要通过蒙特卡洛模拟来实现相邻原子的交换，处理思路与 Ising 自旋模型情形没有区别，但是实际模拟中域的增长是很缓慢的，同时我们在模拟过程中不可避免地需要浪费计算机资源来搜查所有可供交换的近邻原子同时再随机取点，这给模拟带来了很大的计算工作量。

**（2）空穴介导的动力学**

另一种模拟该过程的方案是空穴介导的动力学模型，思路为在 AB 原子中引入少量空穴，由于空穴交换相比随机交换原子而言从概率上讲是更优的交换，跟踪单个空穴来实现对全局的演化模拟是可行的。本模拟中按题设引入一个空穴，由于真实合金中的空位数量也很小，可以通过只包含一个空穴来进行模拟，如此模拟中只需要使用数组来记录每个晶格位置上的原子类型，同时跟踪这单个空穴的位置。模拟中也确实发现，此方案可以更早实现旋节分解的演化过程。

## 4、 算法代码

**（1）主函数**

```
function binary_alloys_simulation()
% 参数设置
L = 128; % 网格边长
N = L * L;
Tc = 2.26; % 临界温度
factor = 0.5; % 设置温度与Tc之比
T = Tc * factor; % 当前温度
num_A = floor(N / 2);
num_B = num_A;
num_mc = 33000; % Monte Carlo 步数(2^15=32768)
num_steps = L * L; % 每次模拟的步数
J = 1; % 能量交互系数

% 生成格子和初始状态
lattice = createRandomLattice(L, num_A, num_B);
start_energy = getEnergy(lattice, L, J);
disp(['初始总能量: ', num2str(start_energy)]);

% 数据存储
energies_in_time = [];
times = [];
pair_correlation = {};
R_cross = [];
time_correlation = [];

% Monte Carlo 模拟
[lattice, energies_in_time, times, pair_correlation, R_cross, time_correlation] = ...
    move(lattice, start_energy, J, T, L, num_mc, num_steps);

% 绘图
plotCorrelation(pair_correlation, time_correlation);
plotRFromPairCorrelation(time_correlation,R_cross);

plotRFromEnergy(times, energies_in_time, L , time_correlation);
end
```

主要组成部分为基本的参数设置，调用初始化函数、MC 模拟函数和绘图可视化函数。实现模拟全过程所调用的函数包括：

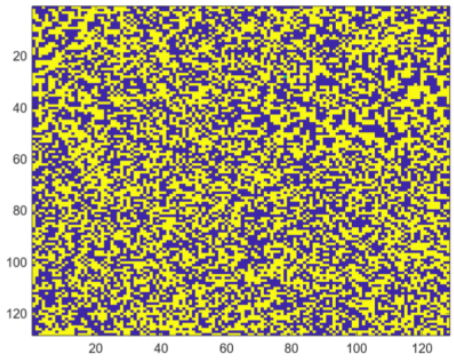
- createRandomLattice()：用于创建随机分布的初始晶格状态；
- getEnergy( )：用于计算当前晶格状态下的总能量；
- move ( )：实现 MC 单次模拟的主要函数，实现 Metropolis 算法过程和提取帧生成 gif 动图的功能，内部亦调用了 calculatePairCorrelation()函数、calculateLocalEnergy()函数等来及时计算各时间点处的统计物理量；
- plot 类别函数：包含 plotCorrelation()、plotRFromePairCorrelation()、plotRFromEnergy()等函数，实现绘制关联函数、$R—t$变化关系图等功能。

**（2）具体函数实现：**

- createRandomLattice()：

```
% 创建值数组并随机打乱
function lattice = createRandomLattice(L, num_A, num_B)
    if(mod(L,2)==0)
        values = [ones(1, num_A), -ones(1, num_B)];
        values(1,randi(L*L))=0;
    elseif(mod(L,2)==1)
        values = [ones(1, num_A), -ones(1, num_B), 0];
    end
    values = values(randperm(numel(values))); % 随机打乱顺序
    lattice = reshape(values, [L, L]); % 转换为 LxL 矩阵
end
```

基本实现思路是先根据$L$奇偶性分类，选择对应的 AB 原子和空穴数创建有序数组，使用内置的 randperm 函数随机打乱数组顺序后，reshape 函数将其重组为$L \times L$矩阵。选取$L = 128$，生成初始的晶格状态参考如下图：




同时我们也可以回答题目设问中的(a)问题：无限温度对应于系统处于完全无序的状态，A 和 B 原子随机分布，没有任何有序性，这类似于在无限温度下系统没有形成任何有序结构的状态。因此，这种随机分布的 A 和 B 原子加上一个空位的配置对应于无限温度。

- move ( )：
  使用 Metropolis 算法来实现空穴和相邻原子之间的交换过程，计算交换能量变化时使用 calculateLocalEnergy()函数计算局部的能量；依照题意，在$t = 2^n, n = 1,2,\ldots$ 处记录晶格状态并计算总能量和关联函数。至于生成 gif 动图方面，则使用 getframe()、frame2im()、imwrite()函数来实现。

```
    % Monte Carlo 迭代
    function [lattice, energies_in_time, times, pair_correlation, R_cross ,time_correlation] =
    ...
        move(lattice, start_energy, J, T, L, num_mc, num_steps)
        [vacancy_row, vacancy_col] = find(lattice == 0, 1); % 找到空位
        if (isempty(vacancy_row) || isempty(vacancy_col))
            error('未找到空穴，请检查初始配置');
        end
        vacancy = [vacancy_row, vacancy_col];
        sum_energy = start_energy;

        % 初始化数据存储
        energies_in_time = [];
        times = [];
        pair_correlation = {};
        R_cross = [];
        time_correlation = [];
        picture_num=1;

        for i = 1:num_mc
            for step = 1:num_steps
                % 随机选择一个移动方向
                dir = randi(4);
                moves = [-1, 0; 1, 0; 0, -1; 0, 1];
                move = moves(dir, :);
                new_vacancy = mod(vacancy + move - 1, L) + 1;

                old_energy = calculateLocalEnergy(lattice, new_vacancy, L, J)*lattice(new_vacancy
    (1),new_vacancy(2));
                new_energy = calculateLocalEnergy(lattice, vacancy, L, J)*lattice(new_vacancy(1),
    new_vacancy(2))+J*lattice(new_vacancy(1),new_vacancy(2))^2;
                dE = new_energy - old_energy;

                if (dE <= 0 || rand < exp(-dE / T))
                    lattice(vacancy(1), vacancy(2)) = lattice(new_vacancy(1), new_vacancy(2));
                    lattice(new_vacancy(1), new_vacancy(2)) = 0;
                    vacancy = new_vacancy;
                    start_energy = start_energy + dE;
                end
            end
```

```
    sum_energy = sum_energy + start_energy;
    energies_in_time(end+1) = sum_energy / (i + 1);
    times(end+1) = i;

    if mod(log2(i), 1) == 0
        disp(['Monte Carlo 步: ', num2str(i)]);
        pair_correlation{end+1} = calculatePairCorrelation(lattice, L);
        temp = calculatePairCorrelation(lattice, L);
        R_cross(end+1) = find(temp(1:end-1) .* temp(2:end) < 0, 1);
        if isempty(R_cross(end))
            error('配对相关函数法未找到R');
        end
        time_correlation(end+1) = i;

        figure;
        plotLattice(lattice);

    end

    fig=figure('Visible', 'off');
    if floor(i/100)==i/100
        imagesc(lattice);
        C{picture_num}=frame2im(getframe(fig));
        picture_num=picture_num+1;
        close(fig);
    end
end

for idx=1:picture_num-1
    [B,map]=rgb2ind(C{idx},256);
    if idx == 1
        imwrite(B,map,'SD_2D.gif','gif', 'Loopcount',inf,'DelayTime',0.05);
    else
        imwrite(B,map,'SD_2D.gif','gif','WriteMode','append','DelayTime',0.05);
    end
end
end
```

➢ calculatePairCorrelation()

```
% 计算配对相关函数
function pair_correlation = calculatePairCorrelation(lattice, L)
    pair_correlation = zeros(1, L/2);
    for r = 1:L/2
        corr = 0;
        for i = 1:L
            for j = 1:L
                corr = corr + lattice(i, j) * (lattice(mod(i+r-1, L)+1, j)+lattice(i, mod(j+r-1, L)+1)+lattice(mod(i-r-1, L)+1, j)+lattice(i, mod(j-r-1, L)+1));
            end
        end
        pair_correlation(r) = corr/L^2;
    end
    pair_correlation = pair_correlation / pair_correlation(1); % 归一化
end
```

计算关联函数时，由于二维晶格的各向异性，为方便起见我们仅计算中心原子和上下左右对应距离的原子之间的关联，系统演化到一定程度时，该近似已可以较好地反映域的尺度特征，当然其中周期性边界条件是默认保证的，具体是通过 mod 函数简单实现。

➢ getEnergy( )& calculateLocalEnergy()

```
% 计算当前网格的总能量
function energy = getEnergy(lattice, L, J)
    energy = 0;
    for i = 1:L
        for j = 1:L
            if lattice(i, j) ~= 0
                neighbors = [lattice(mod(i-2, L)+1, j), lattice(mod(i, L)+1, j), lattice(i, mod(j-2, L)+1), lattice(i, mod(j, L)+1)];
                energy = energy - J * lattice(i, j) * sum(neighbors);
            end
        end
    end
    energy = energy / 2; % 防止双计
end
```

```
% 计算局部能量
function local_energy = calculateLocalEnergy(lattice, pos, L, J)
    neighbors = [lattice(mod(pos(1)-2, L)+1, pos(2)), lattice(mod(pos(1), L)+1, pos(2)), ...
                 lattice(pos(1), mod(pos(2)-2, L)+1), lattice(pos(1), mod(pos(2), L)+1)];
    local_energy = -J *sum(neighbors);
end
```

用于计算晶格全域的能量和局域的能量，注意周期性边界条件的使用和避免能量计算中双计即可。

## 5、 结果与讨论

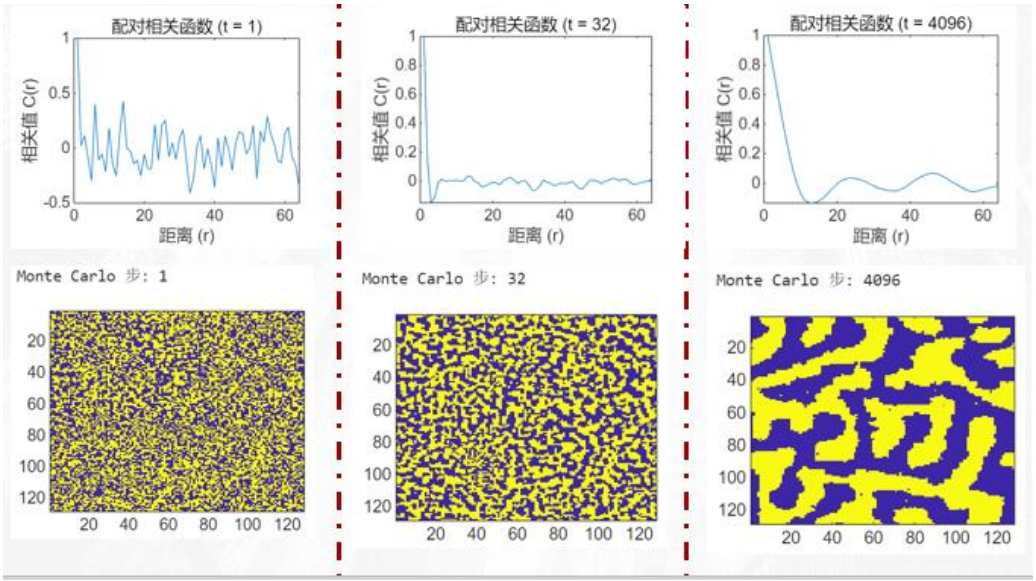
**(a)** 问题已在之前 createRandomLattice()函数的讲解中作出回答。

**(b)** ***R(t)−t* 关系 ($t = 2^n, n = 1, 2, ...$)**

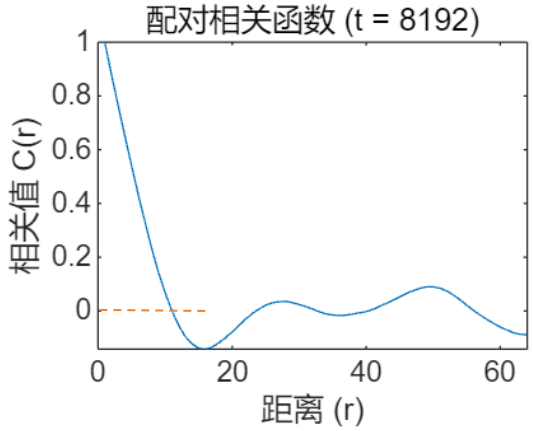
在给出具体结果之前，有必要对两种方法求解域尺度的原理加以阐述：

➢ 关联函数法（**pair correlation function**）：$C(r) =< s_i s_j >$

绘制不同时刻的关联函数$C(r) - r$关系图以及对应的晶格状态，可以看到如下图所示的结果，可见随着蒙特卡洛时间演化，域尺度在不断增长，相关函数的图样也逐渐具有规律性，一个重要的特点是其关联性下降的速度越来越慢，表现在域增长阶段关联性的逐渐扩大。




在系统得到充分演化后，我们可以得到类似下图的$C(r)-r$关系图：



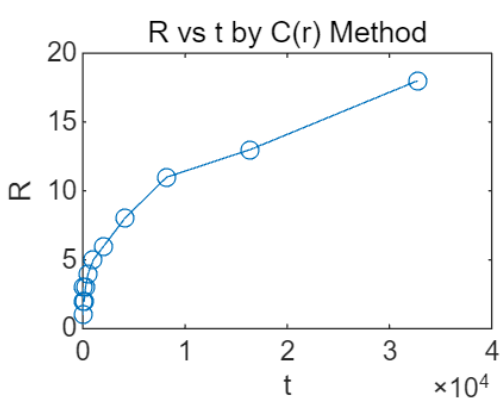

我们选取$C(r)$随 $r$ 首次降为 0 的$r_{min}$作为此时域尺度的衡量标准，意味着此时关联性减弱到很小，可近似认为域生长到该尺度下。在代码实现中则是简单的零点判断程序。

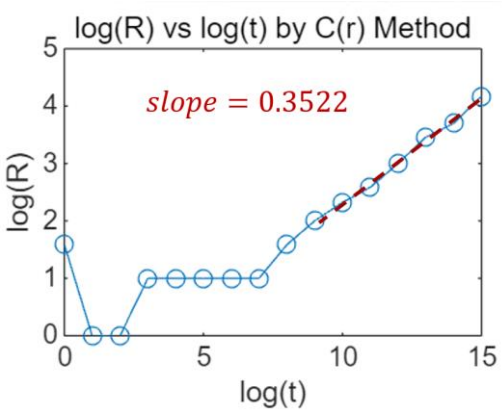
- 能量法(**energy formula** )：$R=\frac{2}{2+<E>/N}$

此公式适用于二维情形，更高维情形可以类推得到具体公式，公式由自旋情形类比而来，具体思想为：一个简单的畴大小测量方法是计算畴的周长长度，这可以通过每自旋的能量计算得出，假设一个由$N$个自旋组成的区域，该区域由向上自旋的畴组成，其周长为$R$，周围是向下自旋的区域。该区域的总能量为$-2N+2R$，因为对于每个位于周长上的自旋，其能量增加 2，因为其中一个相邻自旋将是相反方向的。因此，每自旋的能量$<E>/N=-2+2R/N$。由于$N$与$R^2$成正比，我们得到了公式结果。

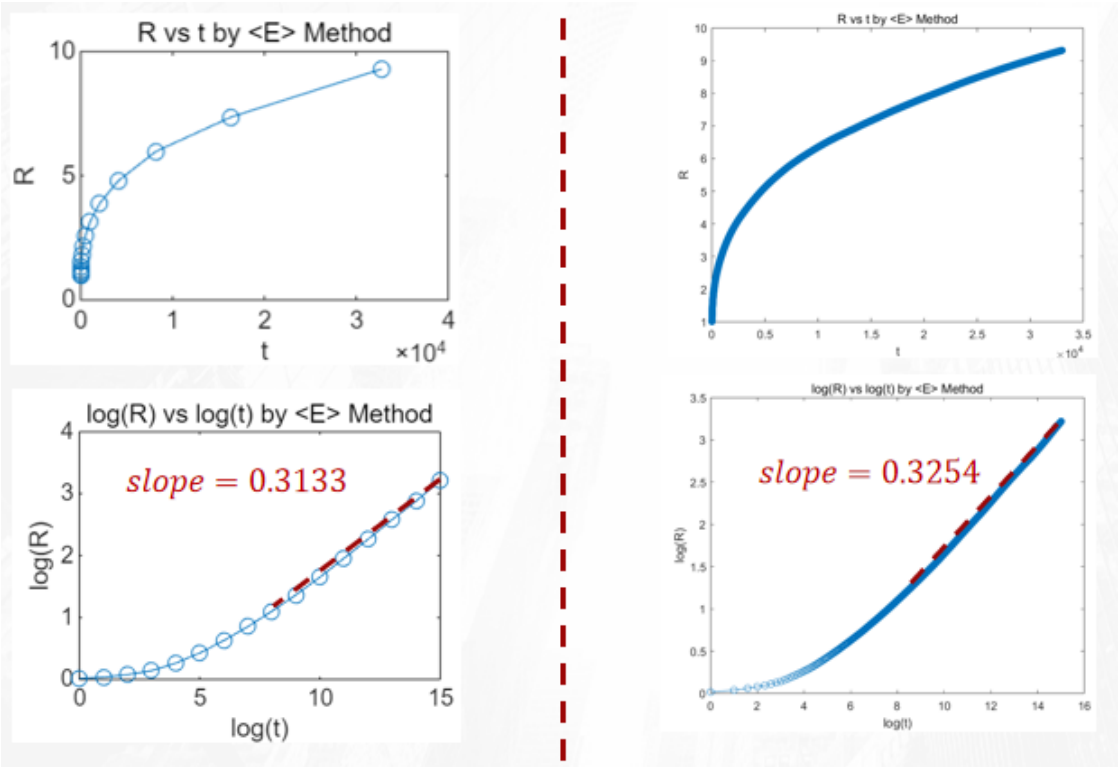
以下我们选定参数，分别就两种方法给出其$R(t)-t$关系图
$L=128, T_c=2.26, T=0.5T_c, (J=1, k_B=1), 33000(\sim 2^{15}\ MCS)$;

- 关联函数法（**pair correlation function**）：$C(r)=<s_i s_j>$




➢ 能量法(**energy formula** )：$R = \frac{2}{2+<E>/N}$

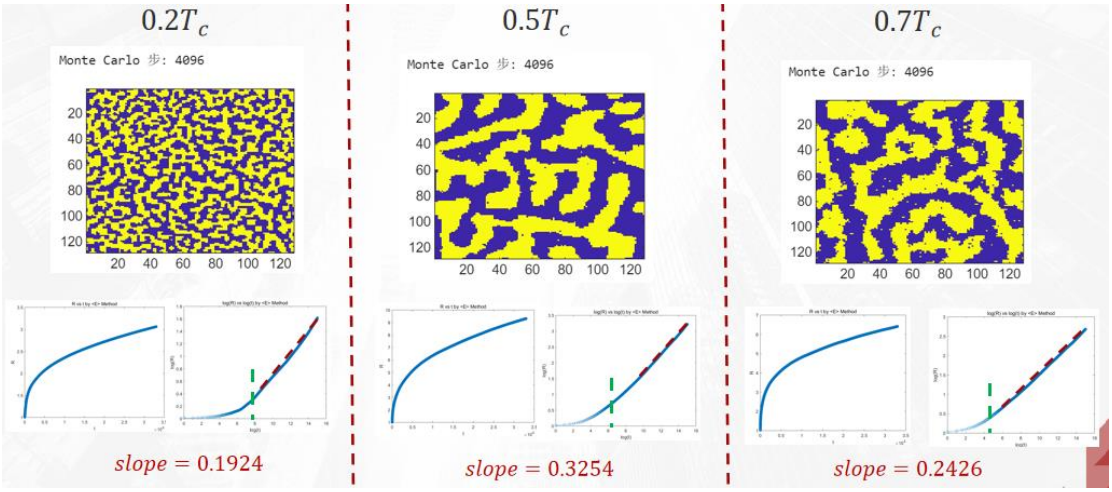



由 loglog 图像可以看出，在演化到一定阶段后，logR 随 logt 开始线性增长，编写代码计算其斜率，得到的斜率值大致在$1/3$附近，这与论文给出的结论相一致，可见$R(t)\sim t^{1/3}$规律得到验证。对比发现，关联函数法求得的结果误差相对较大，这与我们在求解关联函数过程中的近似处理有关，同时蒙特卡洛模拟的演化应当更加充分才可体现更为精准的规律性，在现设参数下已可得到符合预期的结果，考虑后续增加模拟步数来进一步精确结果。

**(c) 温度的影响 (Temperature effects)：**

按照题意条件，更改温度为$T = 0.2T_c$和$T = 0.7T_c$，可以得到如下图的结果。对比发现，在较低温度下，系统仍处于畴增长演化的初级阶段，形成了较多小核但晶粒间组成形成大晶粒的过程较慢，从 loglog 图可见系统在较长时间后才开始显现一定的标度行为，出现线性规律。当提高温度时，演化到后期阶段的速度得以加快，若选取相同的时间节点观察晶格状态，可见温度较高时畴大致已经形成，但再升高温度时畴边界处有些许模糊，可见此处受到

热波动的效应更加显著。至于幂律关系方面，从已模拟的结果来看，其与温度之间的关系不甚明显，当然这也与 MC 模拟的次数有限、演化不充分、所选温度区间过小等因素有关，该方面仍存在大量可探究的内容。




通过查阅相关资料文献，我们可对温度的影响有大致的概念，后续我们将继续深入研究，并尝试复现论文结论，以下仅对文献资料相关内容作简要介绍：

淬火温度的深浅直接影响体系的相分离动力学，包括形态演化、标度行为和域增长动力学等。

（1）深淬火：热波动较弱

初始阶段，由于深度不稳定，体系迅速形成大量小的种子相（初始扰动），产生具有高曲率的小域。域增长主要依赖于界面的驱动力（界面能量的降低）。热波动较弱，体系难以克服能垒，小域的合并较慢。过渡增长阶段较长，形态为分散的孤岛状。

（2）中等淬火（介于深浅之间）

形态演化介于深淬火和浅淬火之间，域增长速率适中，域形态也较规则和稳定，热噪声在初期有助于域的演化，但对界面动力学的影响有限。

（3）浅淬火：接近临界温度

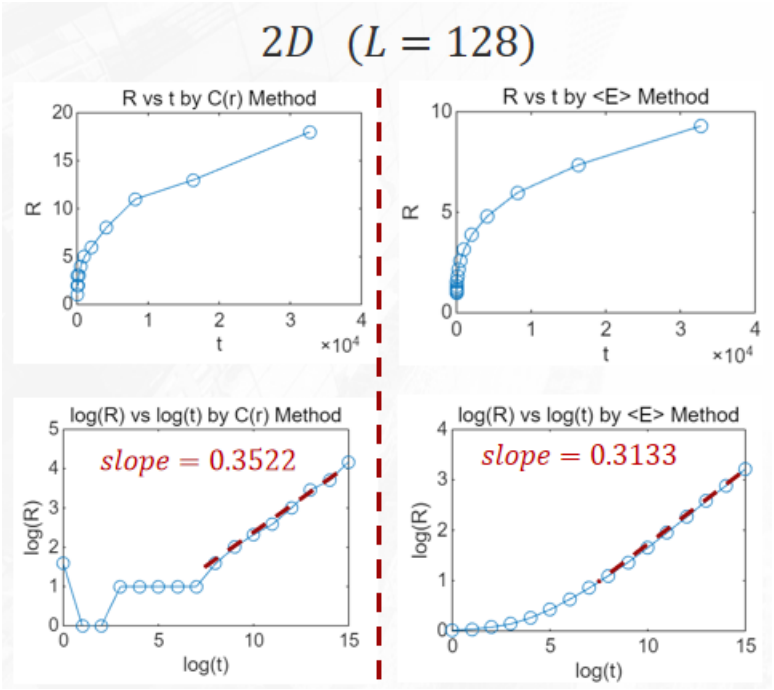
初始形成域块数量较少。热波动显著，界面能垒更容易被克服。体系中的域增长更容易实现，通过扩散驱动域合并，迅速进入标度增长阶段。域的形态更加连续，双连续网络结构（bicontinuous morphology）较为常见。

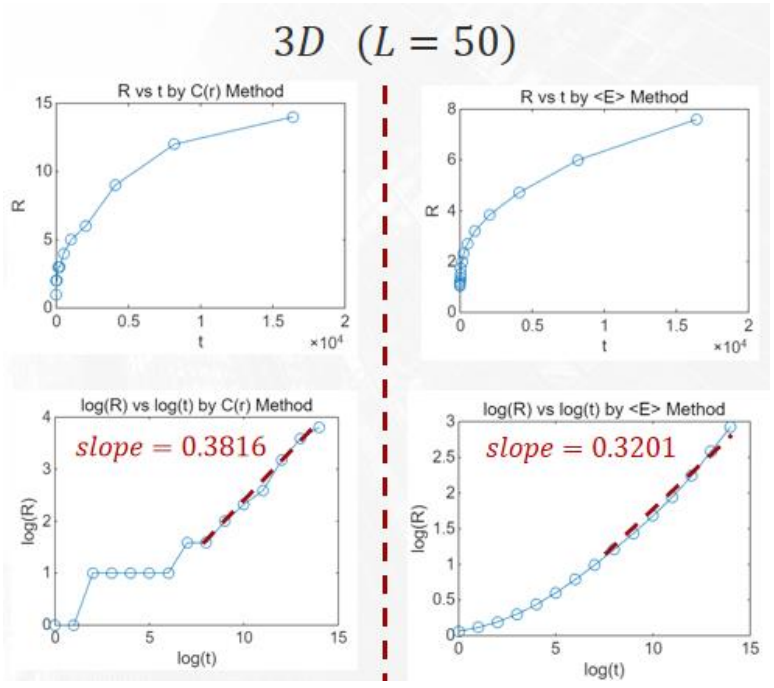
**(d) 维数的影响 (Dimension effects)：**

考虑到 3D 情形模拟的复杂度上升，在尽量兼顾结果准确性的基础上，我们选取模拟参数如下：

$T = 0.5T_c, 17000MCS;$

以下我们分别展示二维和三维条件下，两种方法求得的$R(t) - t$关系，并将其计算的斜率值标注在图上，受限于模拟的算力以及考虑到一定的模拟误差，结合论文给出的结果，我们可以大致认为三维情形下$R(t) - t$间演化的幂律关系仍满足 LSW 理论，即$1/3$的幂次关系。当然这需要更多模拟结果来进一步验证，实验中所取得 MCS 步数依旧较少，考虑未来进一步优化代码细节，获得更精准的结果。




## 6、 总结与展望

**（1）总结：**

- 空穴介导的动力学和自旋交换动力学都可以实现对二元合金淬火的旋节分解过程的模拟，相比之下，空穴介导的动力学方法更为高效，演化速度更快。
- 域生长过程中可以用关联函数$C(r) =< s_i s_j >$和能量法$R = \frac{2}{2+<E>/N}$来近似估计域的尺度，两种方法都可以给出特征性的一致结果，但是为方便起见关联函数法计算过程中存在近似，题设所取的数据点仍较少，给出的求解结果相比能量法仍具有一定误差。

- 温度主要改变的是系统演化的速率，在温度较低时系统演化速率较慢，反之则较快；同时温度较高时域边界处的热波动会更加明显，其他温度的影响结果仍待进一步探究和验证。
- 2D 和 3D 给出的$R(t)-t$演化的幂次关系是近似相同的，都符合 LSW 理论。

**（2）展望：**

- 为了进一步验证论文结果并深入探究各个因素的影响，有必要对现有代码进行优化，同时借助更好的算力设备来进行模拟。
- 参考一些论文中的研究内容，可以向系统中引入更多的参数，例如键无序程度 (bond disorder) 使得系统各向异性的作用更加明显，从而会对域生长的动力学过程产生影响，得到的幂次关系也可能发生改变，是值得探究的方向。
- 对于更高的维度，是否具有相同的幂次关系规律，其理论基础是否是通用的？可以从理论和模拟两方面同时加以分析和验证，并总结其中的物理本质规律，而不是仅仅停留在浅显规律性的总结上。

参考文献：

[1] I.M. Lifshitz, V.V. Slyozov,The kinetics of precipitation from supersaturated solid solutions,Journal of Physics and Chemistry of Solids,Volume 19, Issues 1–2,1961,Pages 35-50, ISSN 0022-3697

[2] Shrivastava, S., & Singh, A. (2023). Phase separation kinetics of binary mixture in the influence of bond disorder: sensitivity to quench temperature. Phase Transitions, 96(5), 311–327.

[3] PhysRevLett.46.1581, Domain Growth of Degenerate Phases, Safran, S. A.}, Phys. Rev. Lett. volume46, issue 24, pages1581--1584, numpages 0, year 1981, month Jun, publisher AmericanPhysicalSociety.doi10.1103/PhysRevLett.46.1581.