

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра прикладної математики та моделювання складних систем

Допущено до захисту

Завідувач кафедри ПМ та МСС

_____ Коплик І.В.

(підпис)

«__» _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»
спеціальність 113 «Прикладна математика»

освітньо-професійна програма

«Наука про дані та моделювання складних систем»

тема роботи «Моделювання методом фазового поля
росту зерен: динаміка та статистичні властивості »

Виконавець

студентка факультету ЕЛІТ

Яковенко Анастасія Юріївна _____

(підпис)

Науковий керівник

кандидат фізико-математичних наук, доцент

Дворниченко Аліна Василівна _____

(підпис)

Суми – 2023

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет електроніки та інформаційних технологій
Кафедра прикладної математики та моделювання складних систем
Рівень вищої освіти перший (бакалаврський)
Галузь знань 11 Математика та статистика
Спеціальність 113 Прикладна математика
Освітня програма освітньо-професійна «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМтаМСС

Коплик І.В.

« __ » _____ 2023 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Яковенко Анастасія Юріївна

1. Тема роботи: «Моделювання методом фазового поля росту зерен: динаміка та статистичні властивості»

Керівник роботи: Дворниченко А.В., кандидат фіз.-мат. наук, доцент
затверджено наказом по факультету ЕлІТ від «01» червня 2023 р. №0605-VI

2. Термін подання роботи студентом: «12» червня 2023 р.

3. Вихідні данні до роботи: Побудувати математичну модель кінцевого різницевого алгоритму розв'язання багатокomпонентного рівняння Аллена-Кана з незбереженою динамікою для росту зерен з різними кристалографічними орієнтаціями методом фазового поля.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити): Огляд літератури. Математичні моделі та реалізація комп'ютерного експерименту. Аналіз отриманих результатів. Висновки.

5. Перелік графічного матеріалу: Ілюстрації результатів комп'ютерного експерименту.

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «10» березня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання роботи	Примітка
1	Огляд літератури	17.04–23.04.2023	
2	Математичні моделі та реалізація комп'ютерного експерименту	24.04–14.05.2023	
3	Аналіз отриманих результатів	15.05–28.05.2023	
4	Оформлення кваліфікаційної роботи	29.05–11.06.2023	

Здобувач вищої освіти

(підпис)

Яковенко А.Ю.

Керівник роботи

(підпис)

Дворниченко А.В.

РЕФЕРАТ

Кваліфікаційна робота: 53с., 7 рисунків, 13 джерел.

Мета роботи: побудувати математичну модель кінцевого різницевого алгоритму розв'язання багатокomпонентного рівняння Аллена-Кана з незберігаючою динамікою для росту зерен з різними кристалографічними орієнтаціями методом фазового поля.

Об'єкт дослідження: рівняння Аллена-Кана для росту зерен з різними кристалографічними орієнтаціями методом фазового поля.

Предмет дослідження: математична модель кінцевого різницевого алгоритму багатокomпонентного рівняння Аллена-Кана для росту зерен з різними кристалографічними орієнтаціями методом фазового поля.

Методи навчання: часткові диференціальні рівняння, моделювання методом фазового поля росту зерен в металургії.

Для побудови математичної моделі методом фазового поля росту зерен було розв'язано рівняння Аллена-Кана для росту зерен з різними кристалографічними орієнтаціями. За допомогою побудованих математичних моделей було побудовано комп'ютерну модель моделювання методом фазового поля росту зерен.

Ключові слова: МАТЕМАТИЧНА МОДЕЛЬ, МЕТОД ФАЗОВОГО ПОЛЯ, РІСТ ЗЕРЕН, ЗЕРНА В МЕТАЛАХ, РІВНЯННЯ АЛЕНА-КАНА, ГРАНИЧНІ УМОВИ.

Зміст

1. Вступ.....	6
2. Мета роботи.....	8
3. Теоретичні основи.....	9
3.1 Опис поняття зерна та його росту.....	9
3.2 Огляд існуючих методів моделювання росту зерен.....	11
4. Модель фазового поля росту зерен.....	14
4.1 Метод фазового поля.....	14
4.2 Варіативні формулювання.....	16
4.3 Модель фазового поля.....	17
5. Чисельна реалізація.....	18
6. Програмна реалізація.....	19
7. Результати.....	20
8. Динаміка та статистичні властивості.....	22
8.1 Еволюція росту зерен.....	24
9. Висновок.....	25
10.Список використаної літератури.....	26
11.Додаток А.....	28

Вступ

Характеристика мікроструктури матеріалів таких як залізо, або сталь є водночас центром уваги для науковців та труднощами в галузі матеріалознавства та інженерії. Мікроструктура матеріалу визначає його механічні властивості, тоді як параметри фазового перетворення впливають на властивості матеріалу. Формування більшості матеріалів із заліза та сталі включає численні фазові перетворення; однак під час їх формування необхідно контролювати багато параметрів, що супроводжується багатьма труднощами, викликаними їх непрозорістю, розміром і високими температурами.

Ріст зерен є фундаментальним процесом, що впливає на структуру та властивості матеріалів. Зерна є основними будівельними блоками металів, полімерів, кераміки та інших матеріалів, їхні розміри та організація визначають різні характеристики. Мікроструктура більшості прикладних матеріалів складається з багатої кількості зерен з різними кристалографічними орієнтаціями. Розуміння факторів, що впливають на еволюцію зернової структури має велике науково-технічне значення, оскільки фізичні властивості, такі як стійкість до корозії, тепло- та електропровідність; і механічні властивості, такі як міцність, пластичність і в'язкість, залежать від середнього розміру зерна та гранулометричного складу.

Дослідження росту зерен має велике значення для розуміння та покращення властивостей матеріалів. Розуміння динаміки та механізмів росту зерен дозволяє прогнозувати структурні зміни під час виготовлення, обробки та експлуатації матеріалів. Це може сприяти покращенню їхньої міцності, тривалості служби, стійкості до зносу та корозії.

Моделювання методом фазового поля росту зерен є потужним інструментом для вивчення динаміки та статистичних властивостей процесів росту зерен в металічних матеріалах. Цей метод базується на математичній моделі, яка використовується для вирішення взаємних проблем на межі розділу

фаз, зокрема, динаміки росту зерен. Шляхом моделювання формування та еволюції мікроструктури розміру зерен отримують кількісні співвідношення між параметрами та кінцевою мікроструктурою. Їх застосовують у реальному виробництві для оптимізації процесу, таким чином, відбувається покращення промислового виробництва. У матеріалознавстві та суміжних галузях метод фазового поля широко використовується як потужний обчислювальний метод для моделювання складних мікроструктур під час затвердіння та фазового перетворення металів і сплавів.

Суть моделі полягає у використанні часткових диференціальних рівнянь для визначення додаткового поля (фазового поля), яке замінює граничні умови на межі розділу фаз. Рівняння ґрунтуються на функції вільної енергії, яка залежить від параметра порядку та дифузійного поля. Формула для моделювання росту зерен залежить від конкретного застосування та може бути досить складною. Найчастіше вона включає в себе часткові диференціальні рівняння та функції вільної енергії, що відповідають властивостям матеріалу. Для точного моделювання необхідно враховувати багато факторів, таких як розмір зерен, температура, склад сплаву та інші.

Отже, тема моделювання методом фазового поля росту зерен є досить місткою та неймовірно цікавою для вивчення. Ріст зерен в матеріалах лежить в основі багатьох галузей, від його вивчення залежить велика кількість параметрів. Також, ця тема торкається нашого повсякденного життя, а саме матеріалів, які ми використовуємо. Виходячи з усього вище написаного, не складно зрозуміти чому була обрана тема «Моделювання методом фазового поля росту зерен: динаміка та статистичні властивості» в якості огляду для бакалаврської роботи.

Мета роботи

Мета даної роботи на тему «Моделювання методом фазового поля росту зерен: динаміка та статистичні властивості» полягає в дослідженні та розумінні процесу росту зерен у матеріалах за допомогою методу фазового поля. Основною метою роботи є проведення детального аналізу динаміки росту зерен, а також вивчення їх статистичних властивостей. Провести літературний огляд та аналіз попередніх досліджень на дану тему.

Побудувати математичну модель кінцевого різницевого алгоритму розв'язання багатокomпонентного рівняння Алена-Кана для росту зерен з різними кристалографічними орієнтаціями методом фазового поля, що описує множення зерен у матеріалах. На основі, побудованої математичної моделі, реалізувати алгоритми та комп'ютерну програму. Провести аналіз отриманих результатів, провести їх обробку та зробити висновки.

Загальна мета даної роботи – це отримання розуміння процесу росту зерен та певним чином внести свій внесок у розвиток матеріалознавства шляхом моделювання методом фазового поля, а також вивчення статистичних властивостей даного процесу. Таким чином, загальна мета передбачає розробку математичної моделі, числових методів та алгоритмів для моделювання росту зерен, проведення числових експериментів, аналіз отриманих результатів і висновків, що в подальшому сприятимуть розумінню процесу росту зерен у матеріалах та його впливу на їх властивості.

Теоретичні основи

Опис поняття зерна та його росту

Зерно в матеріалах є мікроскопічними кристалічними структурами, які мають власну орієнтацію атомів. Зерна формуються під час процесу кристалізації або рекристалізації матеріалу, і вони мають розміри, що варіюються від нанометрів до мікрометрів. Сукупність зерен утворює структуру матеріалу, відому як зернограниці, які визначаються орієнтацією зерен відносно один одного. У матеріалознавстві поняття "зерно" відноситься до окремих кристалічних областей або частинок, які утворюють твердий матеріал. Кожне зерно має свою власну кристалічну структуру, характеризувану певним розташуванням атомів або молекул.

Ріст зерен - це процес збільшення розмірів та перебудови зерен у матеріалі з плином часу. Він може відбуватися під впливом температури, механічного напруження, хімічних реакцій та інших факторів. Ріст зерен включає переміщення зернограниць, а також перебудову кристалічної структури в межах окремих зерен.

Зростання зерен, з свого боку, означає зміну розміру, форми та орієнтації цих окремих зерен протягом часу. Це фундаментальний процес, який відбувається під час затвердіння, відпускання або рекристалізації матеріалів. Зазвичай зростання зерен відбувається через дифузію атомів через межі зерен. При підвищених температурах атоми мають достатню енергію для міграції і переупорядкування у матеріалі. Під час даного процесу атоми з менших зерен можуть перейти до більших зерен, що призводить до збільшення розміру більших зерен та зменшення кількості менших зерен.

Рушійною силою зростання зерен є мінімізація загальної енергії матеріалу. Більші зерна мають меншу енергію міжфазової поверхні на одиницю об'єму порівняно з меншими зернами, тому матеріал має тенденцію розвиватися у стан

з меншою кількістю, але більшими зернами. Цей процес відомий як процес Оствальда.

Кінетика зростання зерен залежить від різних факторів, таких як температура, час та наявність домішок. Вищі температури та довші часи відпускання сприяють швидшому зростанню зерен. Домішки або частинки другої фази можуть гальмувати або підсилювати зростання зерен, в залежності від їх взаємодії з межами зерен. Зростання зерен може мати значний вплив на властивості та характеристики матеріалів. Більші зерна можуть призводити до поліпшення механічних властивостей, таких як підвищена міцність і зменшена схильність до деформації. Однак надмірне зростання зерен також може призводити до ущільнення мікроструктури та потенційного зниження властивостей матеріалу.

Контроль зростання зерен має велике значення у процесах обробки та проектуванні матеріалів. Застосовуються такі методи, як термічна обробка, легування та інженерія меж зерен, щоб впливати на розмір та розподіл зерен з метою досягнення необхідних властивостей матеріалу.

Загалом, розуміння зростання зерен є важливим для формування мікроструктури та властивостей матеріалів, що дозволяє розробляти передові матеріали з певними характеристиками для різних застосувань.

Огляд існуючих методів моделювання росту зерен

В мезомасштабі застосовувались різноманітні обчислювальні методи: Модель Монте-Карло Поттса, метод фронтального відстеження, метод клітинного автомату. У симуляціях беруть участь тисячі зерен. Відстеження окремих меж зерен, а також застосування конкретних конститутивних відносин до їх руху, обчислювально досить вимогливі. Отже, фазово-польове моделювання широко застосовуються для моделювання росту зерна.

Методи клітинного автомату: Моделюють ріст зерен на основі клітинної решітки, де кожна клітина відповідає окремому зерну. Зміни орієнтації та рух зернограниць задаються за допомогою правил, які враховують енергетичні та кінетичні фактори.

Метод мікроскопічних рівнянь: Використовується система диференціальних рівнянь для опису росту зерен на основі законів дифузії та кристалічного росту. Цей підхід враховує мікроскопічні процеси, такі як дифузія атомів та взаємодія зернограниць.

Метод вузлових точок: Застосовується графова структура, де вузлові точки представляють зерна, а ребра - зернограниці. Цей підхід моделює ріст зерен шляхом зміни топології графу, де ребра можуть розриватися, з'єднуватися або перебудовуватися.

Методи міжфазних полів: Використовуються для моделювання росту зерен як еволюцію міжфазової границі між зернами. Вони враховують енергетичні та кінетичні взаємодії між зернами та зернограницями.

Методи випадкових матриць: Базуються на використанні статистичних методів та випадкових матриць для генерації мікроструктури матеріалу та її росту. Цей підхід враховує стохастичний характер росту зерен та його вплив на статистичні властивості матеріалу.

Проте всі ці обчислювальні дослідження, незважаючи на відмінності в підходах, підтвердили параболічність росту зерна на початку 1950-х років та

були представлені Берком і Турнбуллом. Було запропоновано кілька моделей фазового поля для моделювання кінетики росту зерен полікристалічних матеріалів. Одною із найперших і найбільш широко застосованою моделлю, в яких зерна різної кристалографічної орієнтації є представленим набором незбереженого порядку поля параметрів – це модель Чен та Ян. Ця модель використовувалася в 2D і 3D моделюванні фазового поля зростання зерна.

Інша модель багатофазного поля була запропонована І. Стейнбахом, з обмеженням на параметри. Наприклад, сума всіх параметрів порядку при заданій точці поступається одиниці. Фізична інтерпретація цього обмеження полягає в тому, що параметри порядку являють собою об'ємну частку зерен різної орієнтації. Уоррен і колеги запропонували модель параметрів двох порядків, в якій представлений кристалічний порядок і переважна локальна орієнтація кристала.

Ці методи моделювання росту зерен мають свої переваги та обмеження, і вибір конкретного методу залежить від особливостей досліджуваної проблеми та доступних ресурсів.

Використання цих методів дозволяє нам глибше розуміти процес росту зерен, прогнозувати його динаміку та статистичні властивості, а також розробляти стратегії для керування ростом зерен з метою поліпшення властивостей матеріалів. Це має велике значення для різних галузей, включаючи авіацію, автомобілебудування, електроніку, медицину та інші, де потрібні матеріали з покращеними характеристиками.

Таким чином, вивчення росту зерен є важливим завданням в області прикладної математики і матеріалознавства. Ріст зерен визначає структуру та властивості матеріалів і має велике значення для розробки нових матеріалів з покращеними характеристиками. Моделювання росту зерен методом фазового поля дозволяє вивчити динаміку та статистичні властивості росту зерен, а також розробити стратегії для керування цим процесом. Розуміння та використання цих

методів допомагає вдосконалювати матеріали та вирішувати реальні проблеми в різних галузях промисловості.

Модель фазового поля росту зерен

Метод фазового поля

Метод фазового поля є числовим методом, який використовується для моделювання росту зерен у матеріалах. В основі цього методу лежить подання мікроструктури матеріалу у вигляді фазового поля, де кожна точка простору відповідає конкретному зерну. Фазове поле визначається функцією, яка приймає значення відповідної фази (зерна) в кожній точці простору.

Основна ідея методу полягає в моделюванні еволюції фазового поля в часі, враховуючи динаміку та взаємодію зерен. Зерна ростуть, розділяються або об'єднуються в залежності від локальних умов, таких як енергетика зернограниць, кінетика росту та транспорт маси. В результаті моделювання отримуємо інформацію про розподіл та розміри зерен у матеріалі після певного часу.

У методі фазового поля використовуються математичні рівняння для опису росту зерен та їхньої взаємодії. Основними рівняннями, що використовуються, є:

1. Рівняння росту зерен: Воно описує зміну розмірів зерен з часом і залежить від кінетики росту та масового переносу. Це рівняння зазвичай базується на законах дифузії та кристалічного росту.
2. Рівняння динаміки фазового поля: Це рівняння описує зміну фазового поля з часом. Воно враховує ріст зерен, поділ зерен та з'єднання зернограниць. Рівняння включає кінетичні члени, які моделюють розподіл та перерозподіл маси між зернами та зернограницями.
3. Рівняння енергетики зернограниць: Це рівняння описує енергетичні взаємодії між зернами та зернограницями. Воно враховує енергетичні переваги чи недоліки різних зернограниць, які впливають на ріст та перебудову зерен.

Метод фазового поля – це спосіб математичного моделювання для вирішення міжфазних задач. В основному він застосовується до динаміки затвердіння, але також цей метод застосовується в інших ситуаціях, таких як в'язке переміщення(в'язкі пальці), механіка руйнування, воднева крихкість, динаміка пухирців. Метод замінює граничні умови на межі розділу на рівняння в частинних похідних для еволюції допоміжного поля (фазового поля), яке виконує роль параметра порядку. Це фазове поле приймає два різних значення (наприклад, $+1$ і -1) у кожній із фаз, з плавною зміною між обома значеннями в зоні навколо межі розділу, яка потім дифузійна з кінцевою шириною. Дискретне розташування інтерфейсу можна визначити як сукупність усіх точок, де фазове поле приймає певне значення.

Моделі фазового поля зазвичай будуються для того, щоб відтворити задану міжфазну динаміку. Наприклад, у задачах затвердіння динаміка фронту визначається рівнянням дифузії для концентрації або температури в об'ємі та деякими граничними умовами на межі розділу (умова локальної рівноваги та закон збереження), що становить чітку модель розділу. Рівняння фазового поля в принципі відтворюють міжфазну динаміку, коли ширина межі розділу мала порівняно з найменшим масштабом довжини в задачі. З обчислювальної точки зору інтеграція диференціальних рівнянь із частинними похідними, що розв'язують такий малий масштаб, є непомірною. Із зростанням потужності комп'ютерів і теоретичним прогресом у моделюванні фазового поля моделі, фазові поля стали корисним інструментом для чисельного моделювання проблем на їх межі.

Варіативні формулювання

Модель для фазового поля може бути побудована за допомогою фізичних аргументів, якщо є явний вираз для вільної енергії системи. Простий приклад проблем затвердіння наступний:

$$F[e, \varphi] = \int dr [K|\nabla\varphi|^2 + h_0 f(\varphi) + e_0 u(\varphi)^2], \quad (4.2.1)$$

де φ - фазове поле, $u(\varphi) = e / e_0 + h(\varphi) / 2$, e – місцева ентальпія на одиницю об'єму, h — певна поліноміальна функція від φ , та $e_0 = L^2 / T_m c_p$ (де L - прихована теплота, T_m – температура плавлення, c_p – питома теплоємність). $\nabla\varphi$ відповідає міжфазній енергії. Функція $f(\varphi)$ зазвичай береться як подвійний потенціал, що описує щільність вільної енергії об'єму кожної фази, яка сама по собі відповідає двом мінімумам функції $f(\varphi)$. Константи K та h_0 мають розміри енергії на одиницю довжини та енергії на одиницю об'єму відповідно. Тоді ширина інтерфейсу визначається як $W = \sqrt{K / h_0}$. Тоді модель фазового поля можна отримати з таких варіаційних співвідношень:

$$\partial_t \varphi = -\frac{1}{\tau} \left(\frac{\partial F}{\partial \varphi} \right) + \eta(r, t) \quad (4.2.2)$$

$$\partial_t e = D e_0 \nabla^2 \left(\frac{\partial F}{\partial e} \right) - \nabla q_e(r, t) \quad (4.2.3)$$

де D – коефіцієнт дифузії для змінної e , η , q_e це стохастичні терміни, що враховують термічні флуктуації (їх статистичні властивості можна отримати з теореми про дисипацію флуктуацій). Перше рівняння дає рівняння для еволюції фазового поля, тоді як друге є рівнянням дифузії, яке зазвичай переписують для температури або для концентрації (у випадку сплаву).

Модель фазового поля

Для цього прикладу моделлю фазового поля є адаптація моделі росту зерна, наведеної Фан і Чень. У моделі кожна зернина описується одним параметром порядку η_i , який приймає значення одного для визначеного зерна та значення нуль в інших зернах. Еволюція параметрів порядку, що описуються незбереженими Рівняння Алена–Кана у вигляді

$$\frac{\partial \eta_i}{\partial t} = -L_i \frac{\delta F}{\delta \eta_i}, i = 1, 2, \dots, N \quad (4.3.1)$$

де L_i — коефіцієнт рухливості, а F — функціонал вільної енергії, який приймається як

$$F = \int_V \left[f(\eta_1, \eta_2 \dots \eta_N) + \sum_i^N \frac{k_i}{2} |\nabla \eta_i|^2 \right] dv \quad (4.3.2)$$

де k_i — енергетичні коефіцієнти градієнта f — локальна густина вільної енергії.

Специфічна форма локальної вільної енергії, яка є незалежно від орієнтації надається як,

$$f(\eta_1, \eta_2 \dots \eta_N) = \sum_i^N \left(-\frac{A}{2} \eta_i^2 + \frac{B}{4} \eta_i^4 \right) + \sum_i^N \sum_{i \neq j}^N \eta_i^2 \eta_j^2 \quad (4.3.3)$$

де A і B — додатні константи

для $A = B = 1$, для двох параметрів порядку η_1 і η_2 форма цього локального вільного функціоналу, як видно, проходить мінімум на однаковій глибині для η значень нуль і один.

Чисельна реалізація

Підставляючи рівняння. 4.3.2 і 4.3.3 у рівняння 4.3.1, керівним рівнянням для еволюції стає

$$\frac{\partial \eta_i}{\partial t} = -L_i \left(-A\eta_i + B\eta_i^3 + 2\eta_i \sum_{i \neq j}^N \eta_j^2 - k_i \nabla^2 \eta_i \right), i = 1, 2, \dots, N \quad (5.1)$$

Моделювання проводилося для квадрата моделювання, що має $N_x = 64$ і $N_y = 64$ і $dx = dy = 0,5$. Всі параметри використовувалися в симуляції безрозмірної форми.

Перший набір моделювання було проведено для ідеального росту зерна, що є окремим випадком. Нормальний ріст зерна на межі зерна рух керується лише місцевою кривизною межа зерна. Для цього конкретного випадку це так показано, що швидкість росту пропорційна рушійна сила, а рушійна сила пропорційна загальній кількості межі зерна енергії. Показано, що для сферичного зерна, що зменшується, вбудованого у велике друге зерно зміна радіуса зерна, що зменшується, може бути наближено як:

$$D^2 - D_0^2 = kt \quad (5.2)$$

де D і D_0 – поточне та початкове зерно радіус сферичного зерна, відповідно, t – час, а k — константа, що залежить від температури за допомогою рівняння Арреніуса:

$$k = k_0 \exp\left(\frac{-Q}{RT}\right) \quad (5.3)$$

де k_0 — константа, T — абсолютна температура, R — газова стала, Q — енергія активації рухливості меж зерен.

Програмна реалізація

Для моделювання росту зерен методом фазового поля необхідно розробити програмне забезпечення, яке дозволяє виконувати чисельні розрахунки та аналізувати результати. Основні функції програмного забезпечення можуть включати:

1. Створення фазового поля: Програма має забезпечувати можливість генерації початкового фазового поля або імпортування реальних мікроструктурних зображень для подальшого моделювання.
2. Моделювання росту зерен: Програма повинна мати функціональність для обчислення динаміки росту зерен на основі математичних рівнянь, які були описані раніше. Це включає розрахунок зміни розмірів зерен, поділу та з'єднання зернограниць.
3. Аналіз результатів: Програмне забезпечення повинно дозволяти аналізувати отримані результати моделювання, такі як статистичні характеристики розмірів та орієнтації зерен, розподіли розмірів, енергетичні властивості зернограниць тощо.
4. Візуалізація результатів: Для зручного відображення та аналізу результатів моделювання програма повинна мати можливість візуалізувати фазове поле, зерна, зернограниці та інші структурні елементи.

Програма Matlab готує вхідний файл для цих полікристалів моделювання на основі тесселяції Вороного. Для реалізації було використано оптимізовану версію `fd_ca_v2.m`, функції `init_grain_micro.m`, `free_energ_ca_v2.m`, `write_vtk_grid_values.m`, `laplacian.m`. Всі вони представлені у додатку А.

Результати

Перший набір моделювання, як вже було зазначено, було проведено для ідеального росту зерна. В моделюванні використовувалося одне сферичне зерно. В результаті було отримано демонстрацію з якої чітко видно часову еволюцію зерна, що зазначена на малюнку.

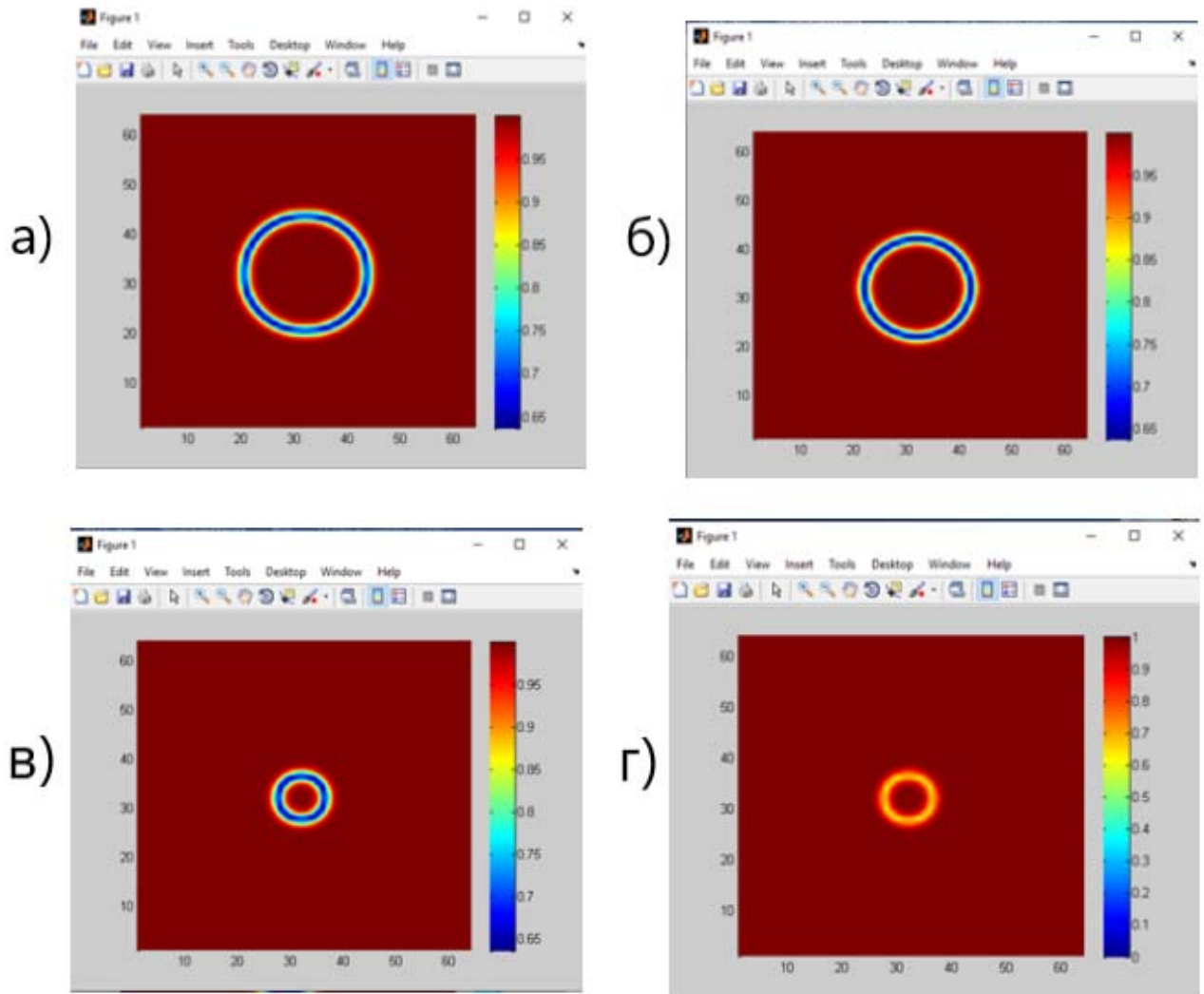


Рис.7.1 – часова еволюція сферичного зерна

а – час = 1000; б – час = 4000; в – час = 7000; г - час = 10000

Другий набір моделювання було проведено для довільно розподіленої мікроструктури з 20 зерен. Як видно, еволюція відбувається разом із зростанням більших зерен і зникнення менших. З часом маленькі зерна мають тенденцію зникати. Найстійкішими виявляються зерна, котрі мають шість граней. Це можна побачити на малюнку 7.2 часової еволюції полікристалічної мікроструктури відповідно до зернового росту.

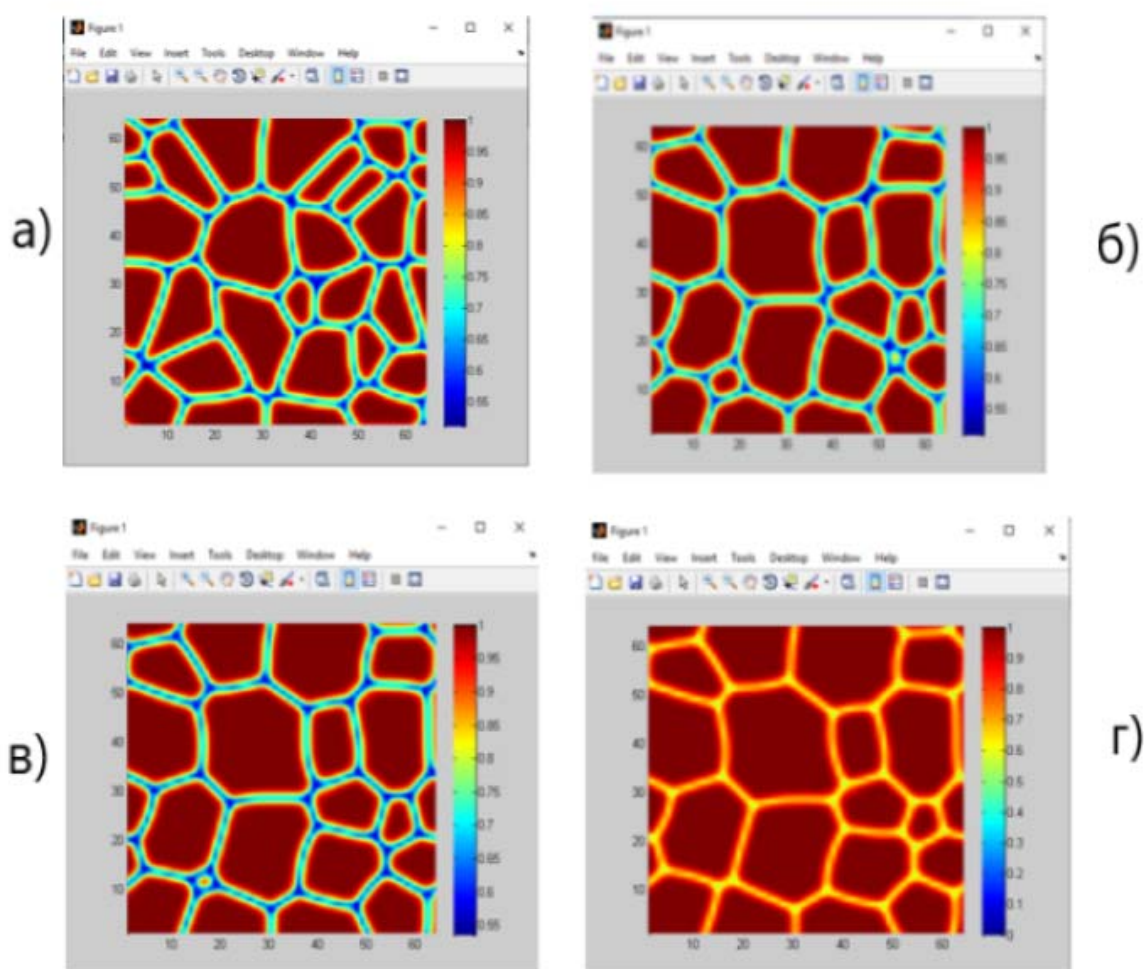


Рис.7.2 - часова еволюція полікристалічної мікроструктури зернового росту.

а – час = 1000; б – час = 4000; в – час = 7000; г - час = 10000

Динаміка та статистичні властивості

З отриманих результатів видно, що добре виражені зерна спостерігаються після певного інкубаційного часу, який складає приблизно 600 одиниць розрахункового часу. При цьому збільшення мобільності прискорює процеси еволюції зерен, середній розмір яких починає стрімко збільшуватися після 6000 розрахункового часу (синя крива). При цьому кількість зерен після цього часу стрімко зменшується, і на великих часових інтервалах досягає майже стаціонарного (постійного) значення, коли середній розмір зерен також перестає змінюватись.

У випадку малого значення мобільності, сформовані після інкубаційного періоду зерна впродовж всього досліджуваного часу майже не змінюються у розмірі і їх кількість лишається постійною (зелена крива).

Порівнюючи отримані результати для фіксованого значення мобільності та різних каппа, отримуємо, що у випадку великих каппа після подолання інкубаційного періоду кількість сформованих зерен є суттєво меншою, порівнюючи з випадком малого каппа. (червона і синя криві). При цьому середній розмір сформованих зерен є також меншим за розмір зерен при малих каппа. Слід відзначити, що збільшення каппа суттєво прискорює процеси еволюції зерен. При збільшенні часу моделювання кількість зерен стрімко зменшується, при цьому швидкість цього зменшення є більшою ніж її значення для випадку малих каппа. Разом з цим зерна зростають з часом і досягають стаціонарного значення більшого, ніж у випадку малих каппа.

Таким чином, динамікою росту зерен, а також їх статистичними властивостями, зокрема середнім розміром зерен та їх кількістю можна керувати як за рахунок зміни мобільності (вибір конкретного матеріалу) так і за рахунок градієнтного коефіцієнту каппа (це може бути зміна температури, або параметри камери, де вирощуються зерна)

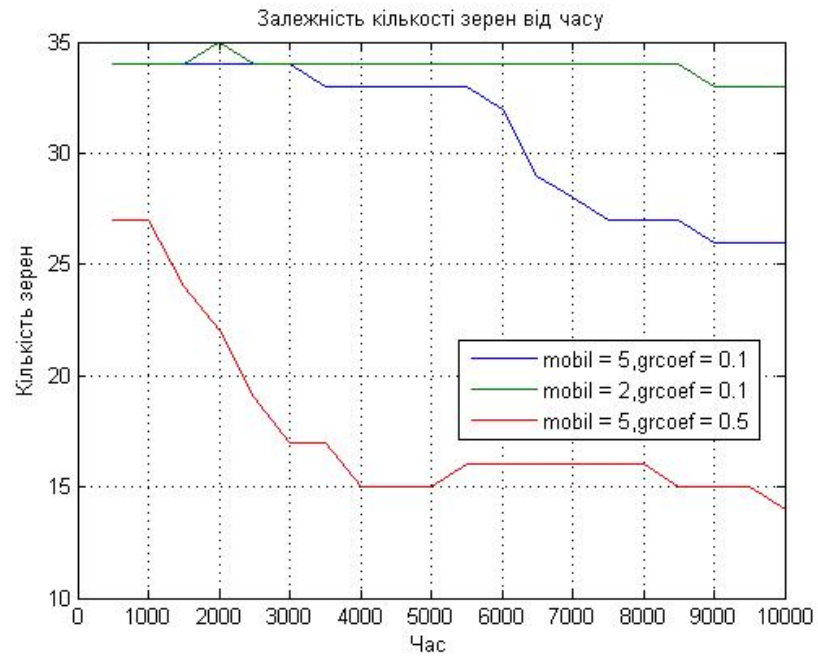


Рис.8.1 – Залежність кількості зерен від часу

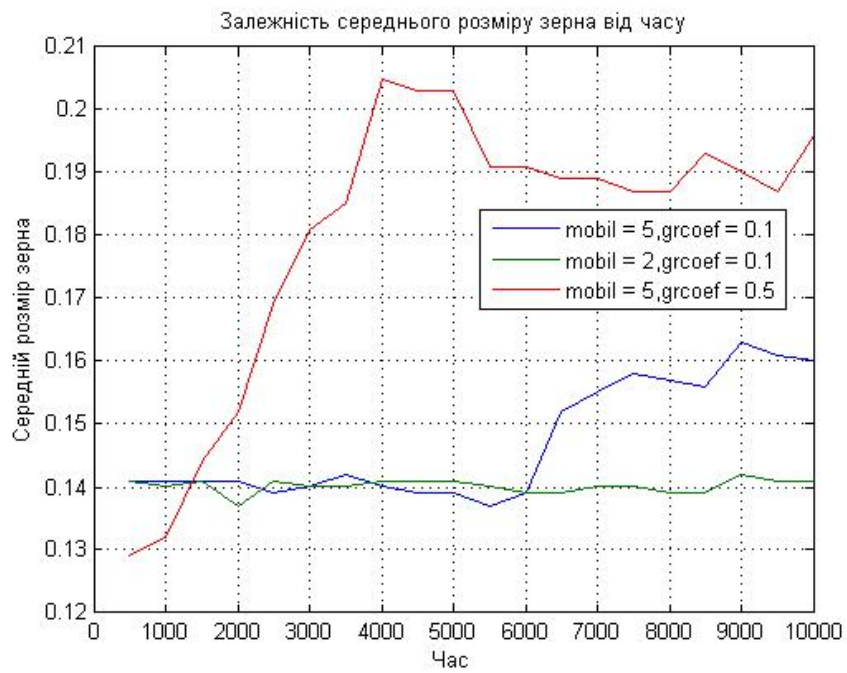


Рис. 8.2 – Залежність середнього розміру зерна від часу

Еволюція зерен має наступний вигляд:

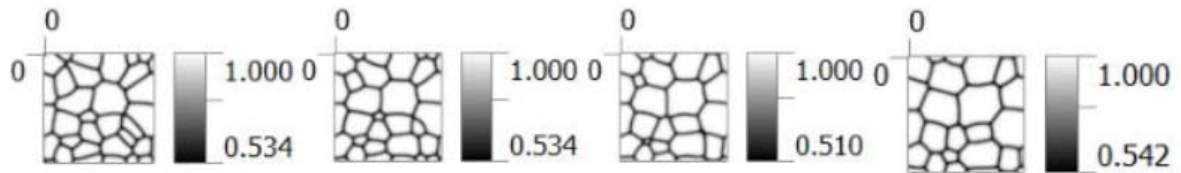


Рис.8.1.1 – Еволюція при параметрах $mobil = 5$, $grcoef = 0.1$.
Моменти часу – 1000, 4000,7000,10000.

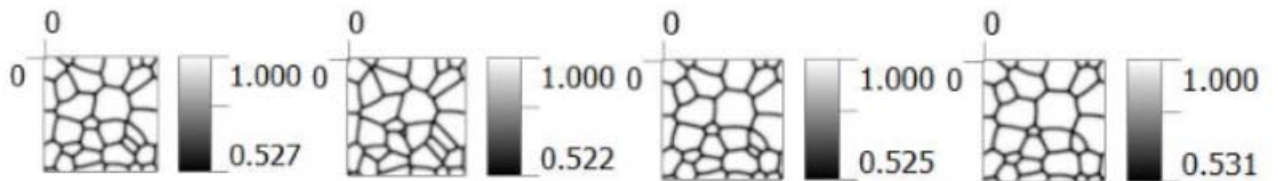


Рис. 8.1.2– Еволюція при параметрах $mobil = 2$, $grcoef = 0.1$.
Моменти часу – 1000, 4000,7000,10000.

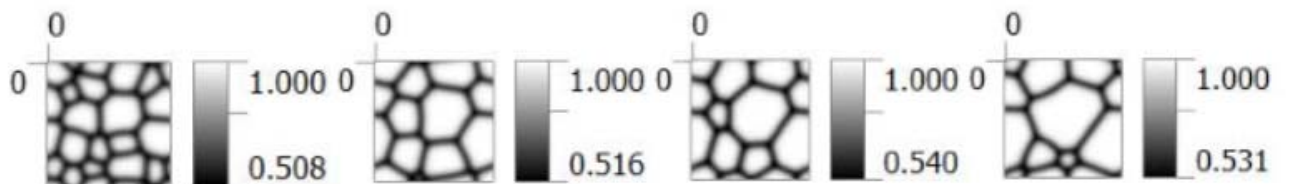


Рис. 8.1.3– Еволюція при параметрах $mobil = 5$, $grcoef = 0.5$.
Моменти часу – 1000, 4000,7000,10000.

Висновки

У даній роботі було проведено дослідження процесу формування та росту зерен за допомогою методу фазового поля. Моделювання здійснювалося з використанням комп'ютерних симуляцій, що дозволило зрозуміти основні закономірності процесу та дослідити статистичні властивості зростаючих зерен. Досліджено вплив коефіцієнта мобільності та градієнту енергії на динаміку росту зерен, їх кількість та середній розмір.

При виконанні роботи було побудовано математичну модель, що базується на рівняннях Алена-Кана для багатокомпонентної системи багатьох зерен з різними кристалографічними орієнтаціями. Для розв'язання просторово-часових динамічних рівнянь в часткових похідних було використано метод кінцевих різниць. На основі, побудованої математичної моделі, було реалізовано алгоритми числового розв'язку та комп'ютерну програму, що дозволяє наглядно побачити часову еволюцію зерен при різних контролюючих параметрах системи. При проведенні моделювання початкова конфігурація матеріалу була обрана у якості твердого розчину (однорідний розподіл зародків зерен у всьому об'ємі); граничні умови були періодичні.

Було встановлено, що починаючи з майже однорідної конфігурації (твердого розчину) добре виражені зерна реалізуються після певного інкубаційного часу. Зі збільшенням часу моделювання кількість зерен з часом зменшується, тоді як середній розмір зростає. Було виявлено, що збільшення як мобільності (кінетичного коефіцієнту) так і енергетичного градієнту приводить до прискорення динаміки росту середнього розміру зерен. На великих часових інтервалах як кількість зерен так і їх середній розмір набувають майже стаціонарного значення.

Отримані результати дозволяють краще зрозуміти процес росту зерен в матеріалах і мають практичне значення для розробки нових матеріалів з бажаними механічними та фізичними властивостями. Розроблені алгоритми

дозволяють прогнозувати мікроструктуру матеріалу із наперед заданими властивостями.

У подальших дослідженнях можна розглядати розширення моделей, враховувати більш складні фізичні процеси та використовувати експериментальні дані для підтвердження та уточнення результатів. Також варто звернути увагу на можливості оптимізації параметрів процесу росту зерен з метою поліпшення властивостей матеріалів.

Список використаної літератури

1. Aflyatunova, D. (2018). Novel Cellular-Automata (CA) based modelling for dynamic recrystallization and grain growth in materials - theory and applications (Докторська дисертація). The University of Sheffield.
2. Biner, S. Bulent. "Programming Phase-Field Modeling." (сторінки 35-50).
3. Cole, D.G., Feltham, P., Gillam, E. "On the Mechanism of Grain Growth in Metals, with Special Reference to Steel." Отримано з:
<https://www.annualreviews.org/doi/pdf/10.1146/annurev.matsci.32.112001.132041>
4. Kundin, J., Farhandi, H., Ganesan, K.P., Almeida, R.S.M., Tushtev, K., Rezwan, K. "Phase-field modeling of grain growth in presence of grain boundary diffusion and segregation in ceramic matrix mini-composites."
5. Ma, N. "Theory and Modeling of Microstructural Evolution in Polycrystalline Materials: Solute Segregation, Grain Growth, and Phase Transformations."
Дисертація для отримання наукового ступеня доктора філософії, Ohio State University.
6. ResearchGate. "Validation of a novel higher-order multi-phase-field model for grain growth simulations using anisotropic grain boundary properties." Отримано з:
https://www.researchgate.net/publication/284252016_Validation_of_a_novel_higher-order_multiphasefield_model_for_graingrowth_simulations_using_anisotropic_grain-boundary_properties
7. Wikipedia. "Phase-field model." Отримано з: https://en.wikipedia.org/wiki/Phase-field_model.
8. Wikipedia. "Grain growth ." Отримано з: https://en.wikipedia.org/wiki/Grain_growth
9. Yang, M., Wang, L., Yan, W. "Phase-field modeling of grain evolutions in additive manufacturing from nucleation, growth, to coarsening."
10. Krill III, C.E. "Phase-Field Modeling of Grain Growth."
11. Suwa, Y. "Phase-field Simulation of Grain Growth."
12. Дворниченко, В., Харченко, Д.О., Харченко, В.О. "Моделювання динаміки формування та росту нанорозмірних поверхневих структур в системах «плазма–конденсат»."

13.Исупова, И.Л., Трусов, П.В. "ОБЗОР МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ДЛЯ ОПИСАНИЯ ФАЗОВЫХ ПРЕВРАЩЕНИЙ В СТАЛЯХ."

Додаток А

В додатку А представлені використані функції для реалізації програми

write_vtk_grid_A1.m

Ця функція записує значення точок сітки до файлу vtk для створення контурних графіків, які можна переглядати за допомогою Paraview. Ця функція та її незначно змінені версії використовуються протягом всієї книги для візуалізації та анімації результатів. Незважаючи на те, що у Matlab і Octave є вражаючі графічні можливості, Paraview має більш потужні функції для візуалізації та просторового та часового аналізу даних.

```
function [ ] = write_vtk_grid_A1(nx,ny,dx,dy,istep,data1)
format long
fname = sprintf('grains_time_%d.vtk',istep);
out =fopen(fname,'w');
nz=1;
npoin =nx*ny*nz;
% start writing ASCII VTK file:
% header of VTK file
fprintf(out,'# vtk DataFile Version 2.0\n');
fprintf(out,'time_10.vtk\n');
fprintf(out,'ASCII\n');
fprintf(out,'DATASET STRUCTURED_GRID\n');
%--- coords of grid points:
fprintf(out,'DIMENSIONS %5d %5d %5d\n',nx,ny,nz);
fprintf(out,'POINTS %7d float\n',npoin);
for i = 1:nx
    for j = 1:ny
        x =(i-1)*dx;
        y =(j-1)*dy;
```

```

    z = 0.0;
    fprintf(out, '%14.6e %14.6e%14.6e\n',x,y,z);
end
end
%--- write grid point values:
fprintf(out,'POINT_DATA %5d\n',npoint);
fprintf(out,'SCALARS CON float 1\n');
fprintf(out,'LOOKUP_TABLE default\n');
for i = 1:nx
    for j = 1:ny
        ii=(i-1)*nx+j;
        fprintf(out,'%14.6e\n',data1(i,j));
    end
end
fclose(out);
end

```

voronoi_1.m

Ця програма генерує випадкову полікристалічну мікроструктуру за допомогою тесселяції Вороного для моделей поля фаз.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generation of polycrystal %
% grain microstructure %
% with Voronoi tessellation %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
out=fopen('Voronoi_vertices.out','w');
out1=fopen('plot_1.out','w');
out2=fopen('final_plot.p','w');
out3=fopen('original_points.out','w');
out4=fopen('cell_1.out','w');
out5=fopen('grain_25.inp','w');
%----
%%% number of initial grains
ngrain = 100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
npoin=25;
xmax=32.0;
ymax=32.0;
x0 =0.0;
y0 =0.0;
extra = 2.0;
%-----
% generate random points
% for voronoi tessellation
%-----
rand('state',7)
```

```

x =xmax*rand(npoin,1);
y =ymax*rand(npoin,1);
%-----
% Duplicate the points for symmetry:
%-----
for ipoin=1:npoin
    jpoин=npoin+ipoin;
    x(jpoин)=x(ipoin);
    y(jpoин)=y(ipoin)-ymax;
end
%---
for ipoin=1:npoin
    jpoин=npoin*2+ipoin;
    x(jpoин)=x(ipoin)+xmax;
    y(jpoин)=y(ipoin)-ymax;
end
%---
for ipoin=1:npoin
    jpoин=npoin*3+ipoin;
    x(jpoин)=x(ipoin)+xmax;
    y(jpoин)=y(ipoin);
end
%---
for ipoin=1:npoin
    jpoин=npoin*4+ipoin;
    x(jpoин)=x(ipoin)+xmax;
    y(jpoин)=y(ipoin)+ymax;
end
%---
for ipoin=1:npoin

```



```

    jpoин=npoin*5+ipoin;
    x(jpoин)=x(ipoin);
    y(jpoин)=y(ipoin)+ymax;
end
%---
for ipoin=1:npoin
    jpoин=npoin*6+ipoin;
    x(jpoин)=x(ipoin)-xmax;
    y(jpoин)=y(ipoin)+ymax;
end
%---
for ipoin=1:npoin
    jpoин=npoin*7+ipoin;
    x(jpoин)=x(ipoin)-xmax;
    y(jpoин)=y(ipoin);
end
%---
for ipoin=1:npoin
    jpoин=npoin*8+ipoin;
    x(jpoин)=x(ipoin)-xmax;
    y(jpoин)=y(ipoin)-ymax;
end
%%--- Print-out original random points: %%

for i=1:npoin
    fprintf(out3,'%4.6e %14.6e\n',x(i),y(i));
end
%%-- Print the simulation cellcoordinates: %%
fprintf(out4,'%14.6e %14.6e\n',x0,y0);
fprintf(out4,'%14.6e %14.6e\n',xmax,y0);

```

```

fprintf(out4,'%14.6e %14.6e\n',xmax,ymax);
fprintf(out4,'%14.6e %14.6e\n',x0,ymax);
fprintf(out4,'%14.6e %14.6e\n',x0,y0);
%=====
%--- generate voronoi diagram
%=====
[c,f] = voronoin([x,y]);
%-----
%- rearrange voronoin output
%- for their connectivity
%-----
nvelem =size(f);
ncount=0;
for i=1:nvelem
    flag=1;
    vnodes =f{i,:};
    nnode=size(vnodes,2);
    for j=1:nnode
        if(vnodes(j) == 1)
            flag=0;
        end
    end
end
if(flag == 1)
    ncount = ncount+1;
    for j=1:nnode
        lnods(ncount,j)=vnodes(j);
    end
end
end
end

```

```

%-----
% print voronoi results to file
% to be viewed later in Voroni_vertices.out
%-----

for i=1:ncount
    fprintf(out,'# i %d\n',i);
    nnode=size(lnods,2);
    for j=1:nnode
        kk = lnods(i,j);
        if(kk ~= 0)
            fprintf(out,'%14.6e %14.6e\n',c(kk, 1),c(kk,2));
        end
    end
    kk = lnods(i,1);
    fprintf(out,'%14.6e %14.6e\n',c(kk,1),c(kk,2));
    fprintf(out,'\n');
end

%=====
% Clip far outside voronoi elements
% from the simulation cell
%=====

nelem=0;
for i=1:ncount
    flag=0;
    for j=1:nnode
        kk = lnods(i,j);
        if(kk ~= 0)
            if(c(kk,1) >= -extra && c(kk,1)<= xmax+ extra)
                if(c(kk,2) >= -extra && c(kk,2)<= ymax+ extra)

```

```

        flag=1;
    end
end
end
end %j
if(flag == 1)
    nelem=nelem+1;
    jnode=0;
    for j=1:nnode
        kk=lnods(i,j);
        if(kk ~=0 )
            jnode=jnode+1;
            lnods2(nelem,jnode) =lnods(i,j);
        end
    end %j
    nnode2(nelem)=jnode;
end % iflag
end %icount
%=====
% Assign grain numbers to
% voronoi elements
%=====
twopi =8.0*atan(1.0);
epsilon=1.0e-4;
for isector=1:9
    for ipoin=1:npoin
        jpoin=(isector-1)*npoin+ipoin;
        for ielem=1:nelem
            theta=0.0;
            nnode=nnode2(ielem);

```

```

for inode=1:nnode
    kk =lnods2(ielem,inode);
    xv1=c(kk,1);
    yv1=c(kk,2);
    jnode=inode+1;
    if( inode == nnode)
        jnode=1;
    end
    jj=lnods2(ielem,jnode);
    xv2=c(jj,1);
    yv2=c(jj,2);
    p2x =(xv1-x(jpoin));
    p2y =(yv1-y(jpoin));
    p1x =(xv2-x(jpoin));
    p1y =(yv2-y(jpoin));
    x1=sqrt(p1x*p1x+p1y*p1y);
    x2=sqrt(p2x*p2x+p2y*p2y);
    if(x1*x2 <= epsilon)
        theta=twopi;
    else
        tx1=((p1x*p2x+p1y*p2y)/(x1*x2));
        if(abs(tx1) >=1.0)
            tx1=0.9999999999;
        end
        theta=theta+acos(tx1);
    end
end %inode
if(abs(theta-twopi) <= epsilon)
    igrain(ielem)=ipoin;
end

```

```

        end % ielem
    end % ipoin
end
%%-----
%% print out as input file
%%-----
[nn1,nn2]=size(c);
nnode=size(lnods2,2);

fprintf(out5,'%5d %5d %5d %5d\n',(nn1-1),nnode,nelem, ngrain);
for i=2:nn1
    fprintf(out5,' %5d %14.6e %14.6e\n', i,c(i,1),c(i,2));
end
for i=1:nelem
    fprintf(out5,'%5d',i);
    for j=1:nnode
        fprintf(out5,'%5d',lnods2(i,j));
    end
    fprintf(out5,'%5d',igrain(i));
    fprintf(out5,'\n');
end
%=====
% graphic output
%=====
for i=1:nelem
    fprintf(out1,'# i %d %d\n',i,nnode2( i));
    nnode=size(lnods2,2);
    ncount=0;
    xcod =0.0;
    ycod =0.0;

```

```

%nnode = nnode2(ielem);
for j=1:nnode
    kk = lnods2(i,j);
    fprintf(out1,'# %5d %5d\n',j,kk);
    if(kk ~= 0)
        fprintf(out1,'%14.6e %14.6e\n',c(kk,1),c(kk,2));
        ncount=ncount+1;
        xcod =xcod +c(kk,1);
        ycod =ycod +c(kk,2);
    end
end
kk = lnods2(i,1);
fprintf(out1,'%14.6e %14.6e\n',c(kk,1),c(kk,2));
fprintf(out1,'\n');
xcod =xcod/ncount;
ycod =ycod/ncount;
fprintf(out2,'set label');
fprintf(out2,'');
fprintf(out2,'%d',i);
fprintf(out2,' at');
fprintf(out2,'%14.6e , %14.6e\n',xcod,ycod);
fprintf(out2,'\n');
fprintf(out2,'set label');
fprintf(out2,'');
fprintf(out2,'%d',igrain(i));
fprintf(out2,' at');
fprintf(out2,'%14.6e , %14.6e\n',xcod+1.5,ycod+1.5);
fprintf(out2,'\n');
end
fprintf(out2,'plot "plot_1.out" w l, "cell_1.out" w l,"original_points.out"\n');

```

%=====

laplacian.m

Ця функція генерує оператор Лапласа за допомогою п'ятиточкового шаблону з періодичними границями.

```
function [grad] =laplacian(nx,ny,dx,dy)
format long;
nxny=nx*ny;
r=zeros(1,nx);
r(1:2)=[2,-1];
T=toeplitz(r);
E=speye(nx);
grad=-(kron(T,E)+kron(E,T));
%-- for periodic boundaries
for i=1:nx
    ii=(i-1)*nx+1;
    jj=ii+nx-1;
    grad(ii,jj)=1.0;
    grad(jj,ii)=1.0;
    kk=nxny-nx+i;
    grad(i,kk)=1.0;
    grad(kk,i)=1.0;
end
grad = grad /(dx*dy);
end %endfunction
```

fd_ca_v2.m

Ця програма розв'язує багатокомпонентне рівняння Аллен-Кана з незберігаючою динамікою за допомогою алгоритму скінченних різниць з п'ятиточковим шаблоном. Часова інтеграція здійснюється за допомогою явної схеми Єйлера. Програма оптимізована для Matlab/Octave.

Програма викликає наступні функції:

init_grain_micro.m

laplacian.m

free_energ_ca_v2.m

write_vtk_grid_values.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% FINITE DIFFERENCE PHASE-FIELD
% CODE FOR SOLVING %
% ALLEN-CAHN EQUATION %
% (OPTIMIZED FOR MATLAB/OCTAVE) %
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%==get initial wall time
time0=clock();
%format long;
%out2 =fopen('area_frac.out','w');
%-- Simulation cell parameters:
Nx =64;
```

```

Ny =64;
NxNy=Nx*Ny;
dx =0.5;
dy =0.5;
%--- Time integration parameters:
nstep =10000;
nprint=50;
dtime =0.005;
ttime =0.0;
%--- Material Parameters
mobil =5.0;
grcoef =0.1;
%
%--- Generate initial grain_structure
%
iflag =2;
isolve =2;
[etas,ngrain,glist] =init_grain_micro(Nx,Ny,dx,dy,iflag,isolve);
%
%--- Get Laplacian templet
%
[grad] =laplacian(Nx,Ny,dx,dy);
%
% Evolve:
%
eta =zeros(Nx*Ny,1);
for istep =1:nstep
    ttime =ttime+dtime;
    for igrain=1:ngrain
        if(glist(igrain) ==1)

```

```

eta =etas( :, igrain);
%%--Derivative of free energy%
dfdeta =free_energ_fd_ca_v2(Nx,Ny,ngrain,etas,eta,igrain);
%%--- Time integration:%
eta =eta - dtime*mobil*(dfdeta -grcoef*grad*eta);
%-- for small deviations:
inrange =(eta >=0.9999);
eta(inrange) =0.9999;
inrange =(eta < 0.00001);
eta(inrange) =0.00001;
%--
etas( :,igrain) =eta;
%-- Check volume fraction of currentgrain:
grain_sum =0.0;
grain_sum =sum(eta)/NxNy;
if(grain_sum <=0.001)
    glist(igrain) =0;
    fprintf('grain: %5d is eliminated\n',igrain);
end%----
end %if
end %igrain
if(mod(istep,nprint) ==0)
    fprintf('done step: %5d\n',istep);
    %fname1 =sprintf('time_%d.out',istep);
    %out1 =fopen(fname1,'w');
    %for i=1:Nx
    %for j=1:Ny
    %ii =(i-1)*Nx+j
    % gg=0.0;
    %for igrain =1:ngrain

```

```

%gg =gg +etas(ii,igrain)^2;
%end
%fprintf(out1,'%5d %5d %14.6e\n',i,j,gg);
%end
%end
%fclose(out1);
%--- write vtk file & calculate area fraction of grains:
eta2=zeros(Nx,Ny);
fprintf(out2,'%14.6e',ttime);
for igrain=1:ngrain
    ncount=0;
    for i=1:Nx
        for j=1:Ny
            ii =(i-1)*Nx+j;
            eta2(i,j) =eta2(i,j)+etas(ii,igrain)^2;
            if(etas(ii,igrain) >=0.5)
                ncount=ncount+1;
            end
        end
    end
    ncount=ncount/(NxNy);
    fprintf(out2,'%14.6e',ncount);
end
fprintf(out2,'\n');
%--
%%% output on the screen
hold off
[X0,Y0] = meshgrid(1:Nx,1:Ny);
[X1,Y1] = meshgrid(1:0.1:Nx);
Z1 = interp2(X0,Y0,eta2,X1,Y1,'cubic');

```

```

SF=surf(X1,Y1,Z1);
colormap jet;%hsv;
shading interp;
colorbar
axis([1 Nx 1 Ny])
view(0,90)
drawnow();
caxis([0 1]);
%%% output to the file
write_vtk_grid(Nx,Ny,dx,dy,istep,eta2);
end %end if
end %istep
%--- calculate compute time:
compute_time =etime(clock(),time0);
fprintf('Compute Time: %10d\n',compute_time);

```

init_grain_micro.m

Ця функція генерує параметри порядку для кругового бікристалу або полікристалічної мікроструктури, яка складається з 25 зерен, що генерується за допомогою вихідного коду.

```
function [etas,ngrain,glist] =init_grain_micro(Nx,Ny,dx,dy,iflag,isolve)
format long;
if(iflag == 2);
    in =fopen('grain_25.inp','r');
end
%-----
% generate two grains
%-----
if(iflag == 1)
    ngrain =2;
    %-etas(, 1) first grain
    %-etas(, 2) second grain
    x0 = Nx/2;
    y0 = Ny/2;
    radius = 14.0; % radius of second grain
    for i=1:Nx
        for j=1:Ny
            ii=(i-1)*Nx+j;
            if(isolve == 2)
                etas(ii,1)=1.0;
                etas(ii,2)=0.0;
            else
                etas(i,j,1) =1.0;
                etas(i,j,2) =0.0;
            end
            xlength =sqrt((i-x0)^2+(j-y0)^2);
```

```

    if(xlength <= radius);
        if(isolve == 2)
            etas(ii,1)=0.0;
            etas(ii,2)=1.0;
        else
            etas(i,j,1)=0.0;
            etas(i,j,2)=1.0;
        end
    end %if
end %j
end %i
end %iflag
%-----
% generate polycrystalmicrostructure %
%-----
if(iflag == 2)
    %-----
    % read the data generated from voroni_1.m
    %-----
    twopi=8.0*atan(1.0);
    epsilon=1.0e-4;
    ndime=2;
    nvpoin = fscanff(in,'%d',1);
    nvnode = fscanff(in,'%d',1);
    nvelem = fscanff(in,'%d',1);
    ngrain = fscanff(in,'%d',1);
    for ipoin=1:nvpoin
        jpoin =fscanff(in,'%d',1);
        dummy =fscanff(in,'%lf %lf',[2,1]);
        for idime=1:ndime

```



```

        vcord(jpoin,idime)=dummy(idime);
    end;

end

for ielem=1:nvelem
    jelem=fscanf(in,'%d',1);
    dummy=fscanf(in,'%d',[nvnode+1,1]);
    for inode=1:nvnode+1
        vlnods(ielem,inode)=dummy(inode);
    end
end

end

%-----
for ielem=1:nvelem
    jnode=0;
    for inode=1:nvnode
        knode=vlnods(ielem,inode);
        if(knode ~= 0)
            jnode = jnode + 1;
        end
    end
    nnode2(ielem)=jnode;
end

%-----
% form the grid%-----
for i=1:Nx
    gx(i) =i*dx;
end

for j=1:Ny
    gy(j)=j*dy;

```

```

end %------
% initilize order parameters
%------
for i=1:Nx
    for j=1:Ny
        for igrain=1:ngrain
            if(isolve == 2)
                ii=(i-1)*Nx+j;
                etas(ii,igrain)=0.0;
            else
                etas(i,j,igrain) =0.0;
            end
        end %igrain
    end %j
end %i%--%--
for i=1:Nx
    for j=1:Ny
        ii=(i-1)*Nx+j;
        for ielem=1:nvelem
            igrain=vlnodes(ielem,nvnode+1);
            theta =0.0;
            mnode = nnode2(ielem);
            for inode=1:mnode
                knode=vlnodes(ielem,inode);
                xv1 =vcord(knode,1);
                yv1 =vcord(knode,2);
                jnode=vlnodes(ielem,inode+1);
                if(inode == mnode)
                    jnode =vlnodes(ielem,1);
                end
            end
        end
    end
end

```

```

xv2=vcord(jnode,1);
yv2=vcord(jnode,2);
p1x =xv1-gx(i);
p1y =yv1-gy(j);
p2x =xv2-gx(i);
p2y =yv2-gy(j);
x1=sqrt(p1x*p1x+p1y*p1y);
x2=sqrt(p2x*p2x+p2y*p2y);
if(x1*x2 <= epsilon)
    theta=twopi;
else
    tx1=((p1x*p2x+p1y*p2y)/(x1*x2));
end
if(abs(tx1) >=1.0)
    tx1=0.9999999999;
end
theta=theta+acos(tx1);
end %inode
if(abs(theta-twopi) <= epsilon)
    if(isolve == 2)
        etas(ii,igrain) =1.0;
    else
        etas(i,j,igrain)=1.0;
    end
end
end % ielem
end % i
end %j%-----
end %iflag
%--- initialize glist:

```

```
for igrain=1:ngrain
    glist(igrain) =1.0;
end
end %endfunction
```

free_energ_fd_ca_v2.m

Ця функція обчислює похідну вільної енергії для розглянутого зерна одночасно в усіх точках сітки.

```
function [dfdeta] = free_energ_fd_ca_v2(Nx,Ny,ngrain,etas,eta,igrain)
format long;
A=1.0;
B=1.0;
NxNy = Nx*Ny;
sum=zeros(NxNy,1);
for jgrain=1:ngrain;
    if(jgrain ~= igrain)
        sum = sum +etas(:,jgrain).^2;
    end
end
dfdeta = A*(2.0*B* eta .*sum+eta.^3 - eta);
end %endfunction
```