

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-06

Дипломний проект

здобувача освіти денної форми навчання

КГ.06.17.000.ДП

***МАЙСТРЕНКА
ГЛЕБА РУСТАМОВИЧА***

**м. Одеса
2023 р.**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Моделювання перколяційних явищ програмними засобами

Проектний матеріал складається з пояснювальної записки на 72 сторінках та графічного (презентаційного) матеріалу на 17 аркушах (слайдах).

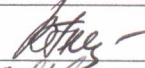
Дипломник  (Майстренко Г.Р.)

Керівник  (Кривченко Ю.В.)

Консультанти:

з економічної частини  (Копайгородська Т.Г.)

з охорони праці  (Чорновол Н.І.)

з дотримання вимог ЄСКД  (Петрашова В.І.)

старший консультант  (Кривченко А.А.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділення  (Скорнякова О.В.)

Захист «23» червня 2023 р.

Протокол ДКК № 5

Оцінка ДКК 5 (відмінно)

Секретар ДКК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та III
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка і Web-дизайн»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР 
Беркань І.В.

“ ” 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Майстренку Глебу Рустамовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Моделювання перколяційних явищ програмними засобами

затверджена наказом по коледжу від “ 17 ” ЖОВТНЯ 202 2 р. № 235-А2-ОД




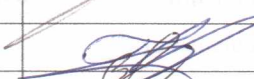
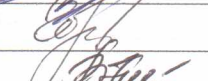

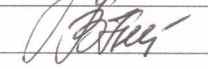

2. Термін здачі закінченого проекту (роботи) 12.06.2023

3. Вихідні данні до проекту (роботи) 1. Математична модель перколяції та кластероутворення, формули для підрахунку параметрів; 2. Елементи кластерної системи мають бути змодельовані у вигляді шарів діаметром 1-10 одиниць; 3. Реалізувати кластероутворення та появу перколяційного кластеру у плоскому та об'ємному середовищі; 4. Передбачити пошук перколяційного (нескінченного) кластеру та аналіз його фрактальної розмірності; 5. Використовувати програмний графічний інтерфейс OpenGL

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Аналітична частина;
Моделювання кластероутворення та розрахунок параметрів перколяційних кластерів;
Розробка алгоритмів імітаційної моделі;
Програмна реалізація комп'ютерної моделі перколяції та кластероутворення

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Предмет досліджень теорії перколяції; Задачі теорії перколяції; Приклади застосування теорії перколяції; Зріст кластерної системи і поява зв'язування; Визначення параметрів кластерів; Діаграма взаємодії функціональних компонент програми; Діаграма взаємодії класів у комп'ютерній моделі; БСА роботи моделі по тактам та коагуляції часток; БСА роботи моделі по тактам та коагуляції часток; Інтерфейс головного вікна на етапі завершення моделювання та модуль аналізу параметрів кластерів; Інтерфейс вікна статистики розподілення кластерів за розмірами та отримання графіків фрактальної розмірності кластерів

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1. Технологічний розділ	Кривченко Ю.В.	 01.05	 12.06
2. Екон. частина	Копайгородська Т.Г.		
3. Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		

7. Дата видачі завдання 01.05.2023

Керівник

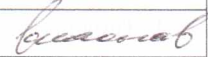
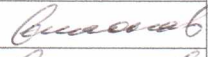
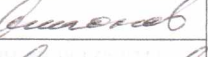
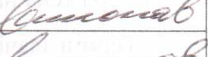
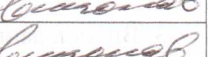
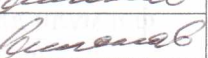


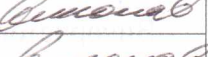
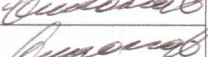
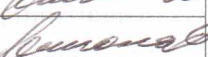
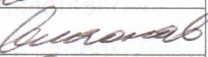
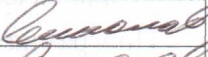


Кривченко Ю.В.

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ. Постановка задачі проектування	22.05.2023	
2.	Аналіз технічного завдання та пошук літератури	24.05.2023	
3.	Огляд моделей кластероутворення	25.05.2023	
4.	Вивчення методів визначення розмірності кластерів	26.05.2023	
5.	Обґрунтування створення моделі кластероутворення	29.05.2023	
6.	Моделювання перколяційних явищ	30.05.2023	
7.	Аналіз результатів роботи перколяційної моделі	31.05.2023	
8.	Визначення розмірностей перколяційних кластерів	1.06.2023	
9.	Реалізація алгоритму виділення та взаємодії кластерів	2.06.2023	
10.	Реалізація алгоритму коагуляції часток	5.06.2023	
11.	Програмна реалізація моделі кластероутворення	6.06.2023	
12.	Реалізація алгоритмів розрахунку параметрів КЧ	7.06.2023	
13.	Реалізація взаємодії функціональних компонент моделі	8.06.2023	
14.	Розробка інтерфейсу ПЗ, тестування моделі	9.06.2023	
15.	Економічні розрахунки і розробка питань охорони праці	10.06.2023	
16.	Виконання графічної частини проекту	11.06.2023	

Дипломник

(підпис)

Керівник

(підпис)

ЗМІСТ

Вступ.....	6
1 Технологічний розділ.....	7
1.1 Обґрунтування створення імітаційно-статистичної моделі.....	7
1.2 Постановка задачі.....	7
1.3 Модель кластероутворення у речовині.....	9
1.4 Опис моделі кластероутворення.....	9
1.5 Аналіз результатів роботи моделі.....	10
1.6 Визначення розмірностей кластерів.....	13
1.7 Програмний інструментарій розробки.....	19
1.8 Реалізація алгоритму виділення кластеру.....	19
1.9 Реалізація алгоритму пересування і взаємодії кластерів.....	21
1.10 Реалізація алгоритму коагуляції часток.....	24
1.11 Реалізація алгоритму роботи моделі у покроковому режимі.....	27
1.12 Програмна реалізація моделі взаємодії часток.....	29
1.13 Реалізація алгоритмів розрахунку розмірностей.....	33
1.14 Реалізація запису статистичних показників.....	38
1.15 Реалізація модуля для розрахунку ступеня впорядкованості зображень.....	39
1.16 Реалізація алгоритмів оцінки хаотизації часток.....	41
1.17 Взаємодія функціональних компонент реалізованої моделі.....	42
1.18 Реалізація алгоритмів обробки помилок.....	45
1.19 Автоматизація роботи у реалізованій моделі.....	46
2 Економічна частина.....	48
3 Охорона праці.....	53
Висновки.....	58
Перелік використаних джерел.....	59
Додаток А. Фрагмент вихідного коду головного модуля програми для моделювання перколяційних явищ.....	60
Додаток Б. Слайди мультимедійної презентації.....	65

ВСТУП

Перколяційні явища характеризуються «критичною точкою», в якій певні властивості системи різко змінюються. До критичних явищ відносяться також фазові переходи другого роду (наприклад, перехід металу з нормального стану в надпровідний при зниженні температури). Фізика всіх критичних явищ дуже своєрідна і має спільні риси, найважливіша з яких полягає в тому, що поблизу критичної точки система як би розпадається на блоки із відмінними властивостями, причому розмір окремих блоків (кластерів) необмежено зростає при наближенні до критичної точки. Обриси блоків при цьому випадкові.

Виявляється, що кластери з різними законами взаємодії при невеликому числі часток володіють багатьма загальними властивостями, зокрема, оболонковою структурою, яка конкурує з виникненням всередині кластера зародка зі структурою "об'ємної фази" (тобто трикутної решітки для двовимірних систем). Відомо, що кластери часток (КЧ) – це основні структуроутворюючі елементи багатьох макроскопічних утворень, виникаючих в різних фізико-хімічних процесах і явищах.

Експериментальне вивчення перколяційних явищ у речовині досить складне: дослідники стикаються з неможливістю отримання досить широкого діапазону структурних параметрів, а також з відсутністю надійної методики їх діагностики. Сучасні відомості про перколяційні явища у речовині багато в чому базуються на теоретичних моделях і не завжди узгоджуються з експериментом. Одна з причин такого становища в тому, що в аналітичних і чисельних дослідженнях заздалегідь фіксуються характер процесу агрегації, тип руху кластерів і часток, ймовірність коагуляції при взаємодії.

У зв'язку з необхідністю вивчення перколяційних явищ на самих ранніх стадіях зародження і зростання кластерів і труднощами, з якими стикаються дослідники в своїй роботі, великі надії покладаються на імітаційне комп'ютерне моделювання. У представлений дипломній роботі виконується спроба створення програмного застосунку саме для моделювання перколяційних явищ.

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

Перколяційні явища та процеси кластероутворення є предметом теоретичних і експериментальних досліджень в різних галузях науки і техніки. Відомо, що кластери – це основні структуроутворюючі елементи багатьох макроскопічних утворень, що виникають в різних фізико-хімічних процесах і явищах.

1.1 Обґрунтування створення імітаційно-статистичної моделі

Імітаційно-статистична модель створена для візуалізації та аналізу перколяційних явищ у речовині, що дозволить вивчати процеси кластероутворення часток в специфічних умовах. При побудові моделі в її основу закладені максимально загальні припущення, засновані на аналізі фізичної ситуації взаємодії часток і кластероутворення.

Мета створення моделі – детальний розгляд процесів кластероутворення в модельних експериментах, дослідження можливості керування ними, вивчення можливості інтенсифікації процесів, пов'язаних з кластероутворенням; визначення умов створення стійких кластерів з часток; визначення умов для створення кластерів зі зниженою величиною механічної напруженості; дослідження кореляції між характеристиками речовини і властивостями кластерів. Для цього в імітаційних експериментах вивчалися питання наявності у речовині кластерів різних типів, ступеня хаотичності та персистентності речовини, а також супутні питання.

1.2 Постановка задачі

Метою дипломної роботи є візуалізація та аналіз перколяційних явищ у речовині та вивчення закономірностей кластероутворення за допомогою комп'ютерних обчислювальних експериментів і розробка для цієї мети моделі і комплексу комп'ютерних програм.

Для цього в роботі поставлені і мають бути вирішені наступні задачі:

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

- розробка математичної моделі кластероутворення;
- розробка методів прогнозування структури і властивостей модельних кластерів;
- створення методики розрахунку полів кластерів;
- візуалізація процесів кластероутворення, перколяційних явищ в речовині.

При успішному вирішенні поставлених задач з'явиться можливість виконання аналізу і модифікації структури кластерів, можливості впливу на їх властивості, кореляції між характеристиками середовища і властивостями кластерів, дослідження їх стійкості до механічних впливів.

Для вирішення поставлених задач розроблено та реалізовано методи:

- побудови кластерів та генерації часток на полі;
- розрахунку взаємодії часток і кластерів;
- розрахунку параметрів для модельних кластерів.

Для створення моделі процесів кластероутворення необхідно розробити комп'ютерну методику імітації руху, взаємодії і коагуляції часток; створити методику визначення структури модельних кластерів, виходячи з гіпотези їх мультікластерної природи. Для цього необхідно створити метод визначення мезоскопічних асимптотик мультікластерів. Знаючи ці межі і скориставшись набором розмірностей Реньї [1], можна отримати спектр кількісних характеристик структури мультікластера.

У модельних експериментах необхідно набрати статистичні дані про структуру, властивості і габітус кластерів, визначити умови переважання одного з типів кластерів і розрахувати механічні напруженості в них з урахуванням того, що існує зацікавленість в інтенсифікації процесу коагуляції, а, отже, серед утворюваних кластерів повинні переважати стійкі до механічних впливів утворення.

Для визначення механічних напружень в кластерах необхідно створити методику розрахунку силових полів в них. З урахуванням складної структури мультікластерів, наявності в них порожнеч різних масштабів, необхідно

модифікувати запропонований Н.Є. Жуковським [2] спосіб розрахунку взаємодії геометрично правильних тіл і створити методику розрахунку для тіл довільної форми, що знаходяться на кінцевій відстані. Методика повинна базуватися на теоремі Н.Є. Жуковського [3] про тяжіння матеріальної точки площини матеріальною пластиною кінцевих розмірів.

1.3 Модель кластероутворення у речовині

Імітаційно-статистична модель процесів кластероутворення повинна мати можливість виявити закономірності еволюції кластерів, мати прогностичні властивості по відношенню до структури і властивостей модельних кластерів, а також можливість визначення ступеня стійкості агрегатів, що утворюються.

У моделі зроблені наступні основні припущення:

1. Частки рухаються у тривимірному зразку речовини;
2. Частки мають нормальний розподіл за розмірами;
3. Частки мають непостійну щільність;
4. Швидкості часток і утворюваних кластерів мають детерміновану і випадкову складову;
5. Взаємодія між тілами виникає на відстані, що залежить від їх маси;
6. Розпад кластерів визначається величиною механічних напруженостей, які розраховуються в моделі.

1.4 Опис моделі кластероутворення

В [6] зазначається, що «одна з найбільш цікавих властивостей аерозолів – їх безперервна і мимовільна коагуляція. Частки будь-якої речовини при зіткненні зливаються або злипаються, аерозоль стає все більш грубим ... ».

У реальній дисперсній системі частки твердої фази, природно, присутні як сили відштовхування, наприклад електростатичні, так і тяжіння. До останніх слід віднести Ван-дер-Вальсові сили, які, як відомо, складаються з трьох компонентів: сил В.Х. Кеезома [4] (діполь-діпольна взаємодія), сил П. Дебая [1,2] (індукційна взаємодія) і сил. Ф. Лондона (дисперсійна взаємодія) [5].

Існування перших двох типів взаємодії передбачає наявність, принаймні, наведеного дипольного моменту [4]. Між полярними молекулами діють тільки дисперсійні сили, які обумовлені флуктуаціями зарядів, що виникають внаслідок руху електронів.

Докладний огляд моделей процесів коагуляції описаний в Розділі 1. Надалі ми будемо користуватися тим, що на процес коагуляції впливає висока концентрація і розмір часток, можлива полідисперсність і гравітаційна коагуляція. Істотна і форма частки, а також інтенсивність перемішування часток в потоці.

У моделі передбачається, що частки об'єднуються в кластери, коли відстань між ними стає менше деякого значення. Величина цього параметра не фіксується в моделі, а визначається масами взаємодіючих часток і кластерів, для яких розглядається акт парної взаємодії. Це правило застосовується до взаємодії між будь-якими утвореннями в двофазному потоці: між частками, між кластерами і в парі частка-кластер. Це крок у напрямку створення універсальної моделі: як раніше згадувалося в розділі 1, відомі моделі обмежують тип структурної взаємодії.

1.5 Аналіз результатів роботи моделі

Аналіз результатів роботи імітаційно-статистичної моделі показав, що передбачена в моделі візуалізація дає можливість побачити відмінності в габітусі одержуваних модельних кластерів. Виявилось, що в системі утворюються два типи модельних кластерів – квазісиметричні компактні і витягнуті – асиметричні кластери зі зниженою щільністю (рис. 1.1 а, б). Для вивчення можливості зміни типу переважаючих кластерів в якості керуючого параметра Ξ обрано відношення середніх значень переносної V_{trans} і випадкової V_{chaot} складових швидкості $\Xi = V_{trans}/V_{chaot}$.

У модельних експериментах при однакових умовах (параметри зразка речовини і часток залишалися незмінними) задавалися різні значення переносної і випадкової складових. Зокрема, з'ясувалося, коли діапазон значень переносної швидкості знаходився в проміжку від 0 до 100 умовних лінійних одиниць (ум. лін.

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

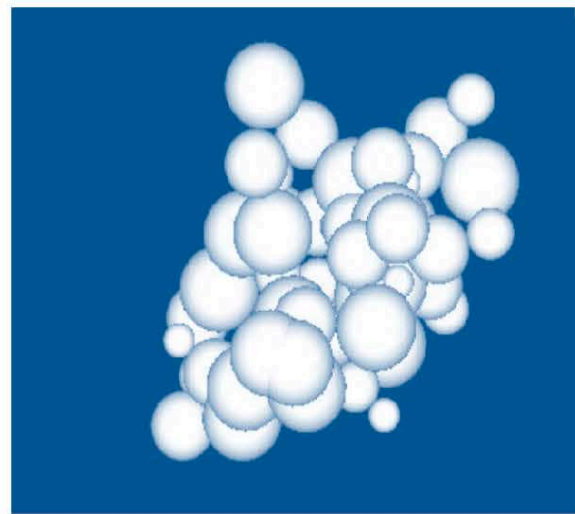
од.), а випадкова – 200 ум. лін. од., тоді тип кластера залишався незмінним – квазісиметричним.

В діапазоні значень переносної швидкості від 100 до 140, а випадкової – 200, тип кластерів перебуває в проміжній фазі, тобто візуально утворювані об'єкти нагадують сферу, але мають відростки, довжина яких сягає 2-4 часток. В діапазоні значень від 140 до 200 умовних лінійних одиниць експериментально встановлено, що тип переважаючих кластерів – асиметричний. Об'єкти набувають відростки, діапазон довжини яких становить не менше 7 часток. Форма і відростки асиметричних кластерів залежать від загального числа часток, які беруть участь в комп'ютерному експерименті.

У модельних експериментах встановлено, що при $\Xi < 0.425$, ймовірність появи асиметричних модельних кластерів, аналогічних представленому на рис. 1.1а, прямує до нуля; при $\Xi > 0.575$, практично усі утворювані в системі кластери – асиметричні.



а) асиметричний



б) компактний

Рисунок 1.1. Характерний вид кластерів, одержуваних при моделюванні

Результати модельних експериментів наведені в табл. 1.1. У другому рядку табл. 1.1 явно показано відношення між переносною і випадковою складовими швидкості. Такі результати можна пояснити тим, що відносно малі значення переносної швидкості і фіксоване значення випадкової швидкості дають можливість часткам «впасти» так, щоб максимально глибоко проникнути в

порожнини кластера. В інших випадках значення швидкостей настільки великі, що ймовірність максимального проникнення частки дуже мала. Слід зазначити, що інтерес може представляти лише кількісна складова результату: переважання одного з типів кластерів в крайніх значеннях відношення між складовими швидкості легко передбачувано.

Таблиця 1.1. Результати, що показують розходження у розмірі кластерів

$\Xi < 0.425$	$\Xi > 0.575$
$V_{trans} / V_{chaot} \approx 2/5$ і менше	$V_{trans} / V_{chaot} \approx 3/5$ і більше
Кластери компактні	Із зниженою щільністю
Квазісиметричні кластери	Асиметричні

Особливості структури кластерів визначалися за допомогою фрактальної, інформаційної та кількох кореляційних розмірностей.

У табл. 1.2 представлені результати статистичної обробки отриманих значень спектра розмірностей Реньї модельних кластерів, що складаються з не менше ніж 100 часток. Результати свідчать про їх мультикластерну природу. В цьому випадку істотно розрізняються розмірності Реньї, наведені в табл. 1.2 (при відносній погрішності рахунку 0,99%). Величину розмірностей кластерів можна регулювати за допомогою зміни виду розподілу за розмірами вхідного потоку часток. У моделі експериментально показано, що максимальна щільність кластерів виходить, якщо частки, що влітають, мають гаусовий розподіл за розмірами.

Таблиця 1.2. Спектр розмірностей модельних кластерів

Розмірність	Квазісиметричні кластери	Асиметричні кластери
d_{max}	$2,98 \pm 0,05, \varepsilon = 0,016$	$2,75 \pm 0,06, \varepsilon = 0,021$
d_{fract}	$2,90 \pm 0,03, \varepsilon = 0,01$	$2,67 \pm 0,08, \varepsilon = 0,03$
d_{inf}	$2,73 \pm 0,04, \varepsilon = 0,014$	$2,54 \pm 0,40, \varepsilon = 0,018$
d_{corr}	$2,67 \pm 0,02, \varepsilon = 0,007$	$2,47 \pm 0,09, \varepsilon = 0,15$
d_{third}	$2,59 \pm 0,03, \varepsilon = 0,001$	$2,34 \pm 0,02, \varepsilon = 0,008$
d_{fourth}	$2,50 \pm 0,05, \varepsilon = 0,02$	$2,29 \pm 0,04, \varepsilon = 0,017$
d_{min}	$2,31 \pm 0,02, \varepsilon = 0,008$	$2,11 \pm 0,05, \varepsilon = 0,023$

1.6 Визначення розмірностей кластерів

У процесі вивчення структури і властивостей кластерів, отриманих в модельних експериментах стало зрозуміло, що з великою ймовірністю кластери будуть володіти мультикластерною структурою.

Існує зручний математичний апарат для визначення властивостей об'єкта, який був запропонований угорським математиком Альфредом Реньї – узагальнений спектр розмірностей [4,5]. Відомо, що ці розмірності мають різні значення. Зокрема, відомо, що всі розмірності будуються в ряд в залежності від їх індексу. Вираз 1.3 являє собою загальну формулу для розрахунку спектра розмірностей Реньї [3].

$$D_q = \frac{\tau(q)}{q-1}, \quad (1.1)$$

$$\text{де } \tau(q) = \lim_{\varepsilon \rightarrow 0} \frac{\ln Z(q, \varepsilon)}{\ln \varepsilon} \quad (1.2)$$

$$Z(q, \varepsilon) = \sum_{i=1}^{N(\varepsilon)} p_i^q(\varepsilon) \quad (1.3)$$

Таким чином, формули для розрахунку розмірностей матимуть такий вигляд:

$$D_{\max} = \frac{\log p_{\min}}{\log \varepsilon} \quad \text{– максимальна розмірність} \quad (1.4)$$

$$D_0 = -\lim_{\varepsilon \rightarrow 0} \frac{\log N(\varepsilon)}{\log \varepsilon} \quad \text{– фрактальна розмірність} \quad (1.5)$$

$$D_1 = \lim_{\varepsilon \rightarrow 0} \frac{\sum_{i=1}^{N(\varepsilon)} p_i \log p_i}{\log \varepsilon} \quad \text{– інформаційна розмірність} \quad (1.6)$$

$$D_2 = \lim_{\varepsilon \rightarrow 0} \frac{\sum_{i=1}^{N(\varepsilon)} p_i^2}{\log \varepsilon} \quad \text{– кореляційна розмірність} \quad (1.7)$$

$$D_{\min} = \frac{\log p_{\max}}{\log \varepsilon} \quad \text{– мінімальна розмірність} \quad (1.8)$$

В результаті вивчення структури кластерів стало зрозуміло, що вони мають неоднорідну структуру. Показником є розбіжність значення фрактальної розмірності зі значеннями спектра розмірностей.

Як зазначалося, в роботі розраховані перші п'ять, а також мінімальна і максимальна розмірності Реньї. Для визначення розмірності використовувався об'ємний аналог палетки, а фрактальна розмірність вимірювалася двома методами – палетки і сфер [5].

У разі визначення фрактальної розмірності кластера в просторі, метод палетки реалізується програмно. За відповідним алгоритмом розраховується кількість зайнятих віртуальних кубічних комірок, що містять об'єкт. А в разі визначення спектра розмірностей Рені відбувається визначення ймовірності заповнення комірок.

Потім, в подвійному логарифмічному масштабі, визначається фрактальна розмірність шляхом апроксимації графіка для мезоскопічної області. Графіки залежності логарифма кількості відповідних комірок від логарифма ребра комірки палетки мають характерний вигляд для всіх розмірностей Реньї, що складаються з трьох ділянок: двох майже горизонтальних і розташованого під кутом до осей. Відомо, що саме він відповідає проміжній асимптотиці. За його кутовим коефіцієнтом визначаються значення розмірностей. Горизонтальні ділянки графіка описують масштаби, відповідні макро- і мікроопису.

Для розрахунку фрактальної розмірності методом сфер в просторі виділяється необхідний кластер і знаходиться його центр мас. Для підвищення точності результатів розрахунків моделі навколо кластера будується сфера, радіус якої вдвічі більше самого маленького його елемента. Кожна наступна ітерація будується зі збільшенням сфери на одиницю. Процес триває до тих пір, поки чергова сфера не покриє саму крайню частку кластера. На кожному кроці розраховується кількість об'єктів, що потрапили в сферу певного радіуса. Розрахунок частини обсягу тіла, яка знаходиться в даній комірці, проводився

методом Монте-Карло [1]. Для побудови графіка залежності визначення фрактальної розмірності, по осях ОХ та ОУ відкладаються значення логарифма кількості часток і логарифма радіусу сфери. Проводиться лінійна апроксимація за методом найменших квадратів.

Застосування методу сфер для розрахунку тривимірних кластерів в просторі, як показано в роботі, має обмеження в своєму використанні: виявилось, що дослідження двох типів кластерів, одержуваних в моделі, асиметричних і квазісиметричних дає результати, що різко відрізняються. Коректно можна проводити розрахунки тільки для квазісиметричних кластерів: отримані результати відрізняються від результатів отриманих методом палетки на 3-7%. Спектр отриманих значень квазісиметричних кластерів, природно, розташовується в діапазоні від 2 до 3, що відповідає, описаному в [2].

Застосування методу сфер для розрахунку фрактальної розмірності асиметричних кластерів дає інший результат. Отримані в модельних експериментах значення фрактальної розмірності методом сфер для асиметричних кластерів дуже великі, що не відповідає показаному в [3].

У модельних експериментах показано, що асиметричні кластери, довжина відростка яких перевищує значення 7 часток, втрачають самоподобу за рахунок величини відростків. Неможливість проводити розрахунки пов'язана з тим, що щільність кластера виходить дуже низька, а відростки довгі і різні. Таким чином, зі збільшенням радіуса кожної наступної сфери на деяке значення, різко змінюється кількість часток, що потрапляють в сферу. Отже, значення розмірності стає набагато більше, ніж раніше встановлено [4]. Некоректне визначення фрактальної розмірності свідчить і про те, що кластер перестає перебувати в мезоскопічній області.

Для усунення такого недоліку був зроблений наступний крок. Змінюючи кількість точок, що піддаються аналізу, можна відсікати відростки асиметричних кластерів і залишати їх в проміжній асимптотиці. Таким чином, можна регулювати властивості асиметричних кластерів. Результати розрахунку

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

фрактальних розмірностей модельних кластерів двома методами наведені в табл. 1.3 (при відносній погрішності рахунку 0,98).

Таблиця 1.3. Результати розрахунку фрактальної розмірності різними способами

Розмірність	Метод палетки		Метод сфер	
	Квазісиметричні кластери	Асиметричні кластери	Квазісиметричні кластери	Асиметричні кластери
d_{\max}	$2,98 \pm 0,03$ $\varepsilon = 0,010$	$2,97 \pm 0,03$ $\varepsilon = 0,010$	$2,90 \pm 0,04$ $\varepsilon = 0,013$	$2,90 \pm 0,04$ $\varepsilon = 0,013$
d_{fract}	$2,90 \pm 0,03$ $\varepsilon = 0,010$	$2,82 \pm 0,02$ $\varepsilon = 0,070$	$2,82 \pm 0,02$ $\varepsilon = 0,007$	$2,77 \pm 0,01$ $\varepsilon = 0,003$
d_{inf}	$2,73 \pm 0,04$ $\varepsilon = 0,014$	$2,70 \pm 0,05$ $\varepsilon = 0,018$	$2,70 \pm 0,03$ $\varepsilon = 0,011$	$2,63 \pm 0,04$ $\varepsilon = 0,015$
d_{corr}	$2,67 \pm 0,02$ $\varepsilon = 0,007$	$2,59 \pm 0,04$ $\varepsilon = 0,015$	$2,60 \pm 0,01$ $\varepsilon = 0,003$	$2,50 \pm 0,02$ $\varepsilon = 0,008$
d_{third}	$2,59 \pm 0,03$ $\varepsilon = 0,011$	$2,47 \pm 0,01$ $\varepsilon = 0,004$	$2,49 \pm 0,02$ $\varepsilon = 0,008$	$2,40 \pm 0,03$ $\varepsilon = 0,012$
d_{fourth}	$2,50 \pm 0,05$ $\varepsilon = 0,020$	$2,37 \pm 0,03$ $\varepsilon = 0,012$	$2,43 \pm 0,05$ $\varepsilon = 0,020$	$2,31 \pm 0,02$ $\varepsilon = 0,008$
d_{\min}	$2,31 \pm 0,02$ $\varepsilon = 0,008$	$2,24 \pm 0,03$ $\varepsilon = 0,013$	$2,31 \pm 0,02$ $\varepsilon = 0,008$	$2,16 \pm 0,01$ $\varepsilon = 0,004$

В процесі роботи з спектром розмірностей Реньї стало зрозуміло, що графіки для мезоскопічної області інформаційної та максимальної розмірності неможливо апроксимувати. Це пов'язано з тим, що крапки не вибудовувалися уздовж кривої апроксимації і представляли собою хмару, що представлена на рис. 1.2.

У процесі розрахунку розмірностей з'ясувалося, що межа мезоскопічної області для розмірностей з різними індексами не збігаються. В роботі спеціально досліджувалося питання визначення меж проміжної асимптотики, всередині яких всі розмірності кластера будуть коректно визначені. За допомогою комп'ютерного експерименту була отримана залежність розмірності кластера від дисперсії часток.

Такий розгляд дозволив зробити крок до вирішення питання, які стоять перед дослідниками в подібних задачах, – які межі проміжної асимптотики в даному випадку.

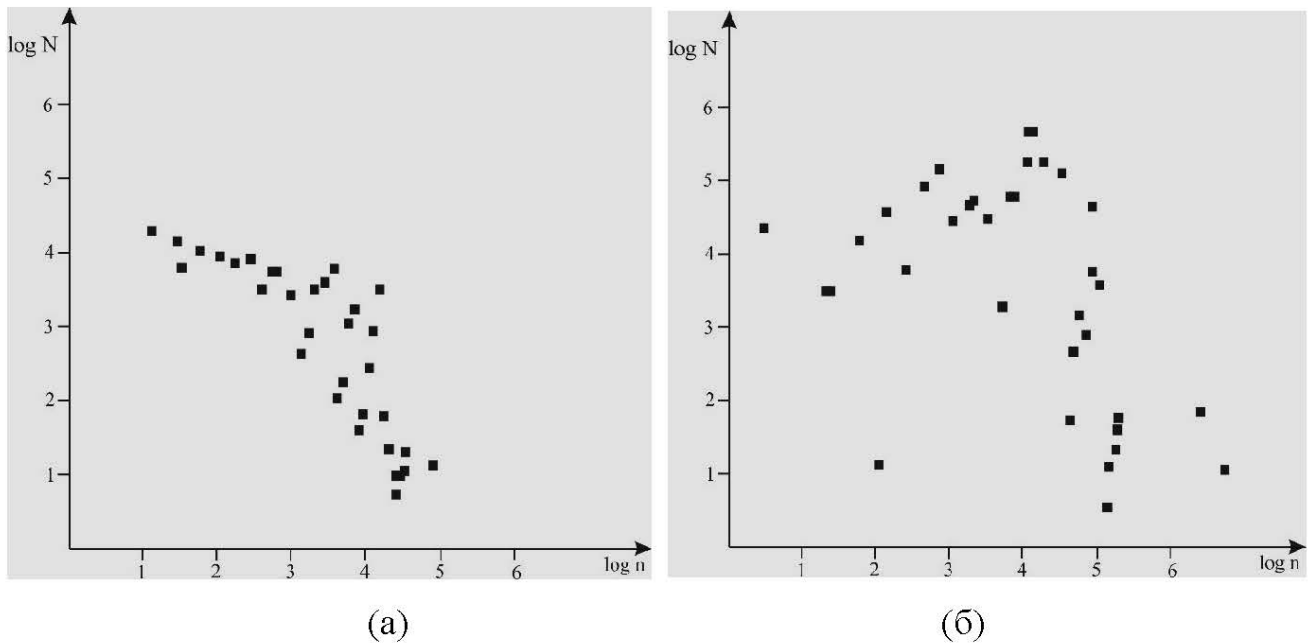


Рисунок 1.2. «Хмари» значень інформаційної (а) і максимальної (б) розмірностей для асиметричних кластерів

Для того щоб весь спектр розмірностей будувався, необхідно, щоб виконувалося співвідношення (2.9) між розмірностями:

$$d_{max} > d_0 > d_1 > d_2 > d_3 > \dots > d_{min} \quad (1.9)$$

Це співвідношення описане в [4] і для цього були проведені додаткові експерименти. В ході аналізу утворених кластерів було встановлено, що найбільш чутливими є інформаційна та максимальна розмірності.

В роботі аналізувалися кластери, що містять не менше 40 часток. Побічним результатом вивчення виявилось, що для кластерів від 40 до 100 часток всі розмірності будуються відповідно до описаних, хоча максимальна кількість часток, яка може входити в кластер, для кожного типу різна. Для них графіки інформаційної та максимальної розмірності апроксимуються коректно, що показано на рис. 1.3. Також для проведення коректної апроксимації графіків важлива і форма кластера.

Розрахунок інформаційної та максимальної розмірності асиметричних кластерів залежить від обсягу (кількості часток) і довжини відростків, тобто від максимальної відстані від центру мас кластера до його найдальшої точки.

Експериментально встановлено, що якщо кількість часток, що знаходяться у відростку кластера перевищує 7, то графіки цих розмірностей стають схожими на хмару, що показано на рис. 1.2 а) і 1.2 б).

На рис. 1.3 представлені графіки інформаційної (а) і максимальної (б) розмірностей, що приймають коректні значення. У межі довжини відростка до 7 часток ці дві розмірності будуються відповідно до зазначених вище співвідношень. Проміжки впливу кількості часток, що входять в кластер, варіюються в проміжку від 40 часток до 150. При більшому обсязі асиметричного кластера відростки настільки великі, що самоподоба і структурність кластерів зникають. Квазісиметричні кластери також мають проміжки, де спектр розмірностей перестає задовольняти зазначеним співвідношенням. Аналіз квазісиметричних кластерів, що містять менше 40 часток, є недоцільним, тому що в них не спостерігається структурної і статистичної самоподоби.

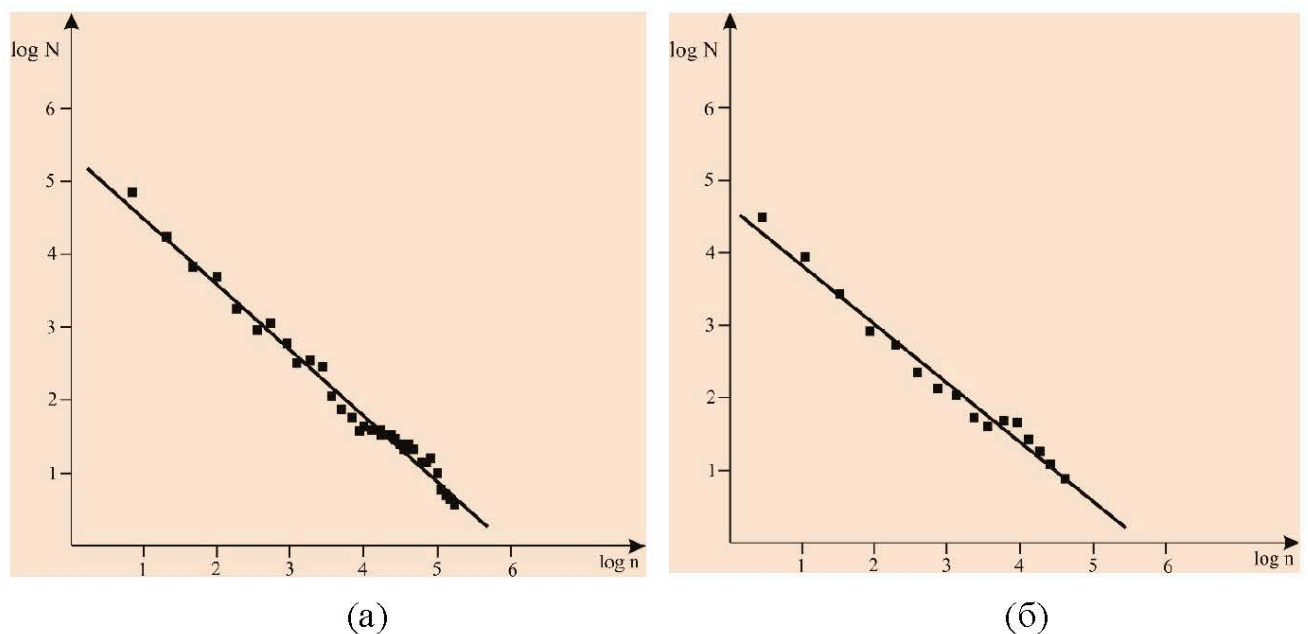


Рисунок 1.3. Коректна апроксимація інформаційної (а) і максимальної (б) розмірностей

Кластери, у складі яких більше 40 часток, починають проявляти статистичну самоподобу і можуть бути проаналізовані за допомогою всього спектру розмірностей. У проміжку від 40 і до 150 часток, що входять до складу квазісиметричного кластера, весь спектр розмірностей задовольняє зазначеним співвідношенням.

1.7 Програмний інструментарій розробки

Комп'ютерна імітаційно-статистична модель створена з використанням об'єктно-орієнтованих (ОО) принципів. Для розробки програмного продукту було використано інтегроване середовище розробки Embarcadero RAD Studio 10.3 та ОО-мову програмування Delphi. Такий підхід дозволив описати всі структури, реалізовані в моделі, за допомогою класів, процедур і функцій. Використовуючи конструктори і деструктори, вдалося оптимізувати деякі алгоритми, зокрема, рух одиночних часток і кластерів, розрахунок значень спектра розмірностей, візуалізацію кластероутворення і, відповідно, часток у речовині. При розробці програмного коду широко застосовувалися алгоритми рекурсивного аналізу даних, а також релевантного пошуку. Для зберігання тимчасових структур використовувалися двовимірні динамічні масиви, що дозволили забезпечити необхідну швидкість моделі, яка досягнуто при автоматизації таких процесів як розрахунок механічних напруженостей, що виникають в кластерах і візуалізації їх подальшого розпаду. Для відображення проміжної інформації, такої як дисперсія часток, параметри речовини, зовнішні умови системи і т.д., широко застосовувалися списки і поля введення/виведення. Модель складається з наступних базових класів: речовина, частка, кластери, які в свою чергу мають ряд дочірніх класів, що відповідають за певні обчислення або дії.

1.8 Реалізація алгоритму виділення кластеру

Програмна реалізація моделі містить у собі можливість зміни освітлення сцени, повороту зображення по всіх осях, наближення і віддалення об'єкту, а також можливість вписати кластер в тривимірну сферу. На базі відповідних об'єктів, методів і класів, був досягнутий ефект панорамної розгортки зображення. Це дає можливість динамічно перемальовувати об'єкт в момент функціонування комп'ютерної моделі. За допомогою розробленого алгоритму зменшення затримки для виведення зображення вирішена проблема виведення на екран кластерів різних типів, що складаються з більш ніж 100 часток.

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

У моделі розроблено алгоритм відображення всіх існуючих кластерів із зазначенням кількості часток, що входять до їх складу. Для цього використовувалися списки (TListBox) зі стандартного інструментарію Embarcadero RAD Studio. Для програмної реалізації кластерів як структур використані динамічні масиви даних (ClusterParams: array of TClusterParams). Структура даних частки (TClusterParams) містить наступні значення: r – радіус частки, dx , dy , dz – зміщення частки відносно центру мас кластера по осях Ox , Oy , Oz відповідно. Масив представлений у вигляді матриці розміру n , n де n – це кількість часток. У свою чергу, матриця є так званої таблицею зв'язків, яка зберігає інформацію про сформовані кластери і показує, які частки з якими пов'язані. Її ініціалізація відбувається при старті системи, а головна діагональ заповнюється прапорами зі значенням 2, що говорить про наявність зв'язку між частками. Існують наступні прапори, які в певному випадку встановлюються в необхідну позицію: 0 – зв'язок не встановлений, 1 – зв'язок необхідно перевірити, 2 – зв'язок сформований. Потім створюється ще один динамічний масив, розмірність якого m , де m – це об'єкт класу частки. У ньому зберігається інформація про координати центру частки, величина та напрямки векторів її швидкості, а також методи і властивості для оперування наведеними параметрами. У процесі руху частки і кластери твердої фази взаємодіють, отже, для розрахунку статистичних показників необхідно проводити виділення кластерів по таблиці зв'язків. Алгоритм виділення наступний:

1. Обирається одна з часток, параметри якої на даному етапі моделювання ще не були змінені, і маркується як модифікована;
2. Номер обраної частки поміщається в створений динамічний масив, що визначає розглянутий кластер;
3. Розглядаються всі частки по таблиці зв'язків, з якими дана має зв'язок, і кожна нова частка заноситься в масив кластера. Таким чином, розглядається рядок з номером обраної частки і в кінець масиву кластера записуються номери стовпців, в яких є нульове значення, але які ще не записані в поточний масив;

4. Обирається наступна частка з масиву, що описує розглянутий кластер і повторюється пункт 3.

Пункти 3 і 4 обираються до тих пір, поки не будуть виділені всі частки, записані в масив кластера. Більш докладно алгоритм виділення кластера представлений блок-схемою на рис. 1.4.

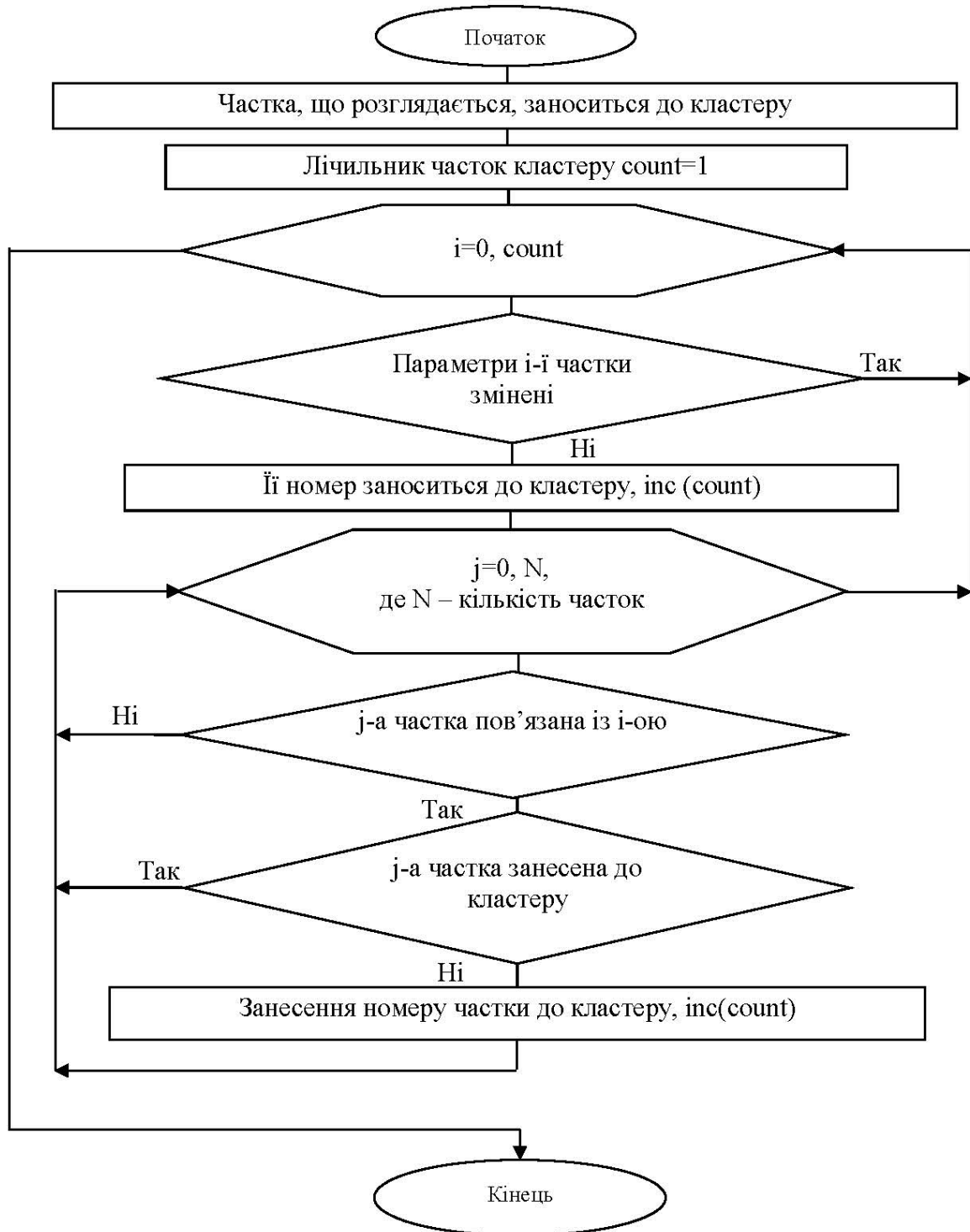


Рисунок 1.4. Блок-схема алгоритму виділення кластера

Зм.	Арк.	№ докум.	Підпис	Дата

КГ 06. 17 000. 00 ДП ПЗ

Арк.

21

1.9 Реалізація алгоритму пересування і взаємодії кластерів

Алгоритм переміщення кластера можна умовно розділити на 6 етапів:

1. Пошук частки у кластері з максимальною відстанню від осі зразку речовини, з огляду на траєкторію руху;
2. Розрахунок наступного положення частки;
3. Перевірка: якщо нове положення знаходиться за межами конструкції (зразку речовини), то необхідно перейти до п. 4, інакше – до п. 5;
4. Фіксація координати удару частки об стінку конструкції (межу розподілу зразку речовини). Якщо удар був у області закругленої частини, то залишений відбиток (координати і радіус), заносяться в масив;
5. Визначається зміщення поточної частки щодо попереднього положення і зміщення інших часток кластера. Параметри часток, що перемістилися, змінюються, а точка перетину приймається за початкові координати. Відбувається циклічний перехід до пункту 1.

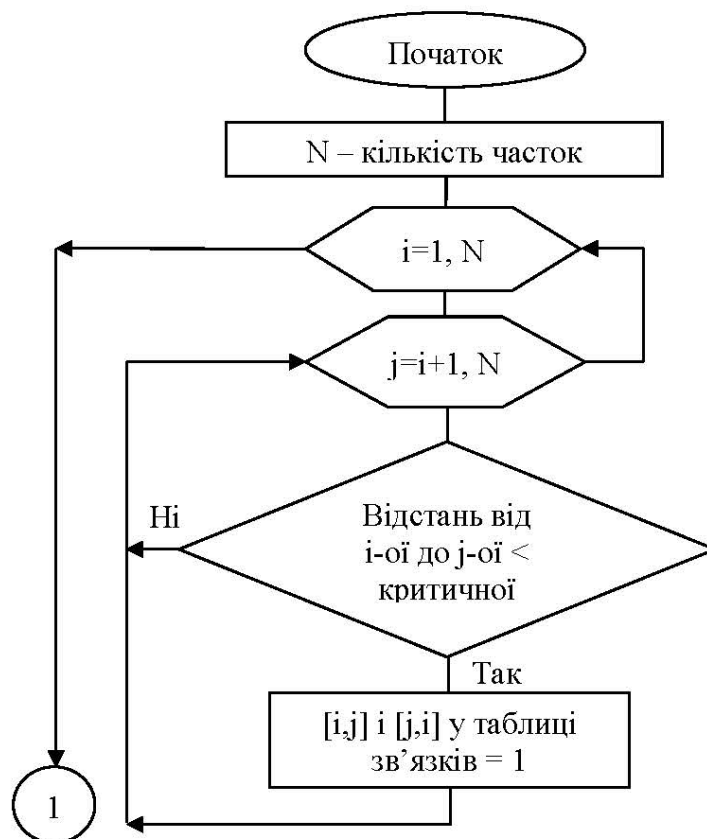


Рисунок 1.5а. Блок-схема алгоритму пересування і взаємодії кластерів

На рис. 1.5 представлена деталізована блок-схема алгоритму пересування і взаємодії кластерів.



Рисунок 1.5б. Блок-схема алгоритму пересування і взаємодії кластерів

1.10 Реалізація алгоритму коагуляції часток

Для відображення заплямованості закругленої частини конструкції (межі розподілу зразку речовини) використані списки типу ListBox (клас TListBox), які відображали ступінь її заповнення в процентному співвідношенні і містили значення відносин площ відбитків кластерів до загальної площі стінки для кожного проходу частками всієї конструкції, що знаходяться у ній. Розрахунок відбувався в момент проходження останньої часткою повного циклу, тобто всієї конструкції. Для того, щоб визначити момент проходження часткою повного циклу, необхідно уточнити наступний факт: проходженням часткою всієї конструкції n -й раз вважається шлях, пройдений нею не менше, ніж третина шляху всіх часток k раз, де k може приймати значення $k \geq n$ для кожної частки окремо.

Для відображення і розрахунку загальної площі «сліду» стінки в моделі зроблено припущення, що рахуються тільки ті відбитки, що не покриваються повністю будь-якими іншими «слідами».

Процес коагуляції одиночних часток в імітаційній моделі відбувається за наступним алгоритмом:

1. Розглядається відстань між одиночними частками, тим самим, реалізуючи перевірку відстані між кожною часткою. Кожна i -а частка перевіряється на можливість коагуляції тільки з j -ми частками, де $j > i$. Якщо відстань більше заданої, то розглядаються наступні пари часток, інакше в комірці $[i, j]$ і $[j, i]$ заноситься «1»;
2. Виділення кластера, утвореного з колишніх і тільки що взаємодіючих часток, тобто умовою додавання нового елемента в кластер є наявність зв'язку між частками з вагою зв'язку більше 0;
3. Визначаються частки, які мають тільки що утворені зв'язки;
4. Визначається найменша відстань між частками, що вже «склеїлися» і новими;

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

5. Змінюється розташування нового кластера щодо старого по прямій, що з'єднує частки, які знаходяться на найкоротшій відстані одна від одної;

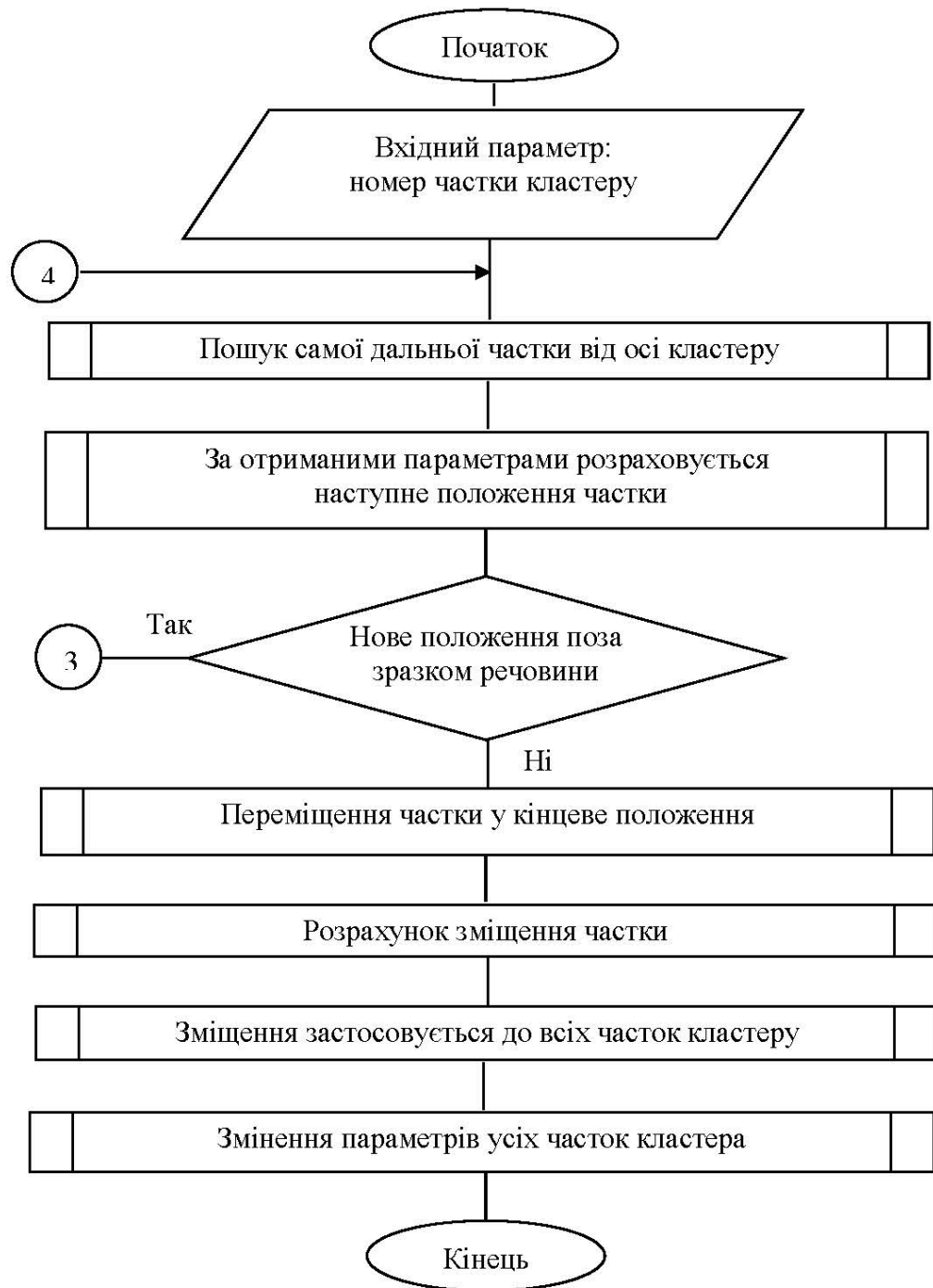


Рисунок 1.6а. Блок-схема роботи алгоритму коагуляції часток

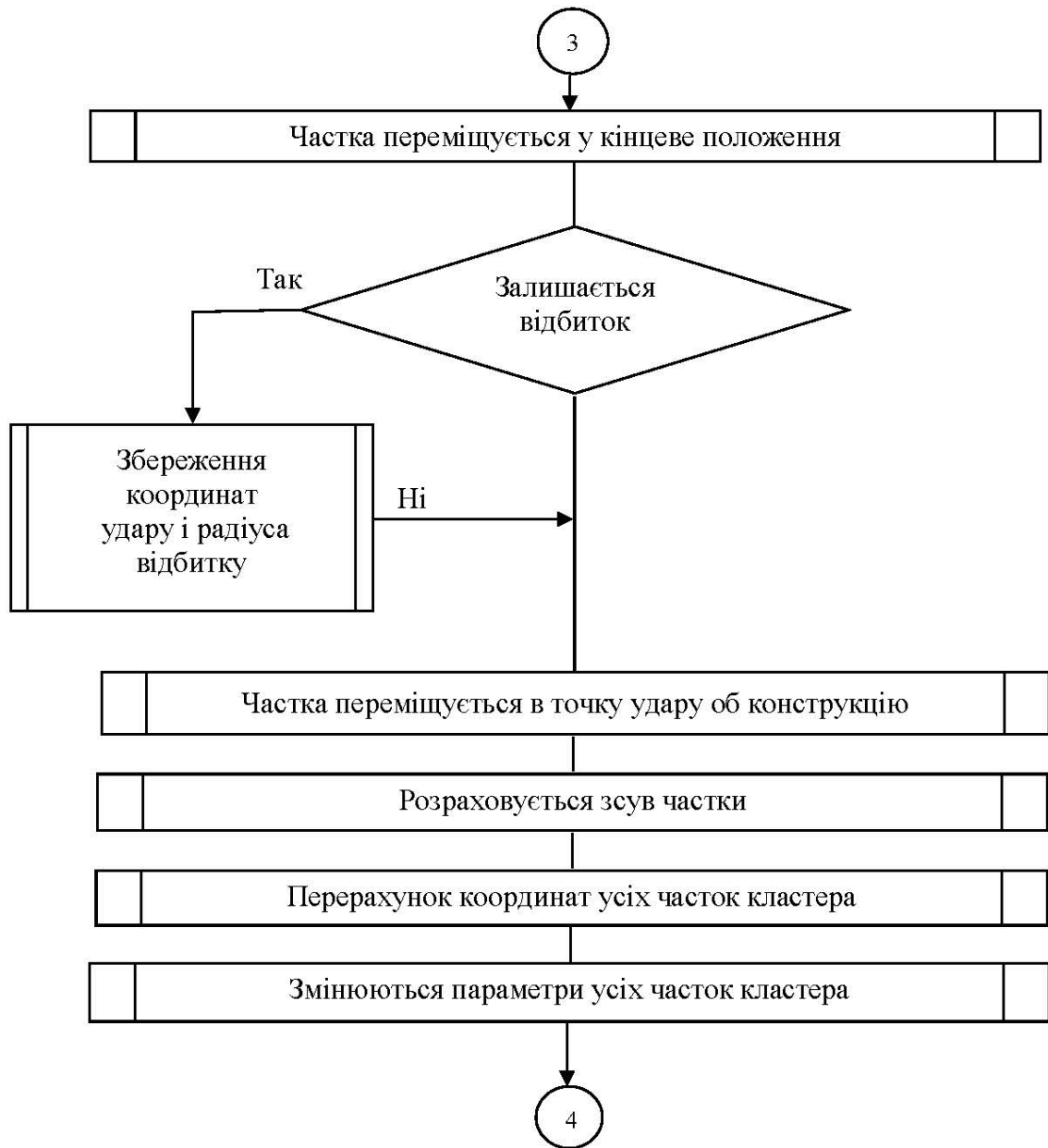


Рисунок 1.6б. Блок-схема роботи алгоритму коагуляції часток

6. Кожен кластер розглядається як частка зі своєю швидкістю, напрямком руху і своєю масою, яка дорівнює сумі мас часток, що знаходяться в ньому. Також необхідно відзначити, що кластери можуть складатися і з однієї частки. Такий облік часток необхідний для передачі параметрів кластерів у функцію розрахунку нових значень швидкостей;
7. Далі знаходяться нові значення параметрів кластерів, що визначають швидкість пересування;
8. В таблиці зв'язків змінюються всі одиничні зв'язки нового кластера зі старим на зв'язку з вагою «2».

Пункти 5-8 працюють циклічно, поки не зникнуть всі одиничні зв'язки в таблиці. Для імітації «склеювання» часток розроблено алгоритм, який присвоює однакові параметри руху взаємодіючим часткам.

Подальший рух утвореного кластера, тобто сукупності коагульованих часток, відбувається з однаковою швидкістю і напрямком.

Алгоритм роботи реалізованої функції коагулювання представлений на рис. 1.6.

1.11 Реалізація алгоритму роботи моделі у покроковому режимі

Моделювання може бути проведено з використанням покрокового і автоматичного режимів. Нижче наведено алгоритм роботи моделі в покроковому режимі:

1. Таймер збільшується на 1;
2. Відбувається ініціалізація і візуалізація зразку речовини;
3. Виділяються кластери по таблиці зв'язків часток;
4. Відбувається пересування кластера процедурою (MoveCoagulant);
5. Здійснюється перевірка, чи «склеїлись» які-небудь частки/кластери і відбувається перевірка відстані між ними;
6. Виділяється кластер, що утворився;
7. Розташування часток в кластері змінюється;
8. Вносяться зміни в параметри склеєних часток і кластерів.

В автоматичному режимі моделювання експеримент триває до певного моменту, який визначається користувачем. Рисунок 3.4 показує блок-схему роботи моделі в покроковому режимі. Варто відзначити, що комп'ютерна реалізація моделі оперує тільки кластерами, в тому числі і частками, які розглядаються як кластери, що складаються з одного елемента, і є тимчасовою структурою, що містить в собі інформацію, яка використовується тільки на поточній ітерації процесу моделювання. Постійною ж структурою, що зберігає

інформацію про стан системи, є масив часток і таблиця зв'язків, з використанням яких і формуються тимчасові структури даних кластерів.

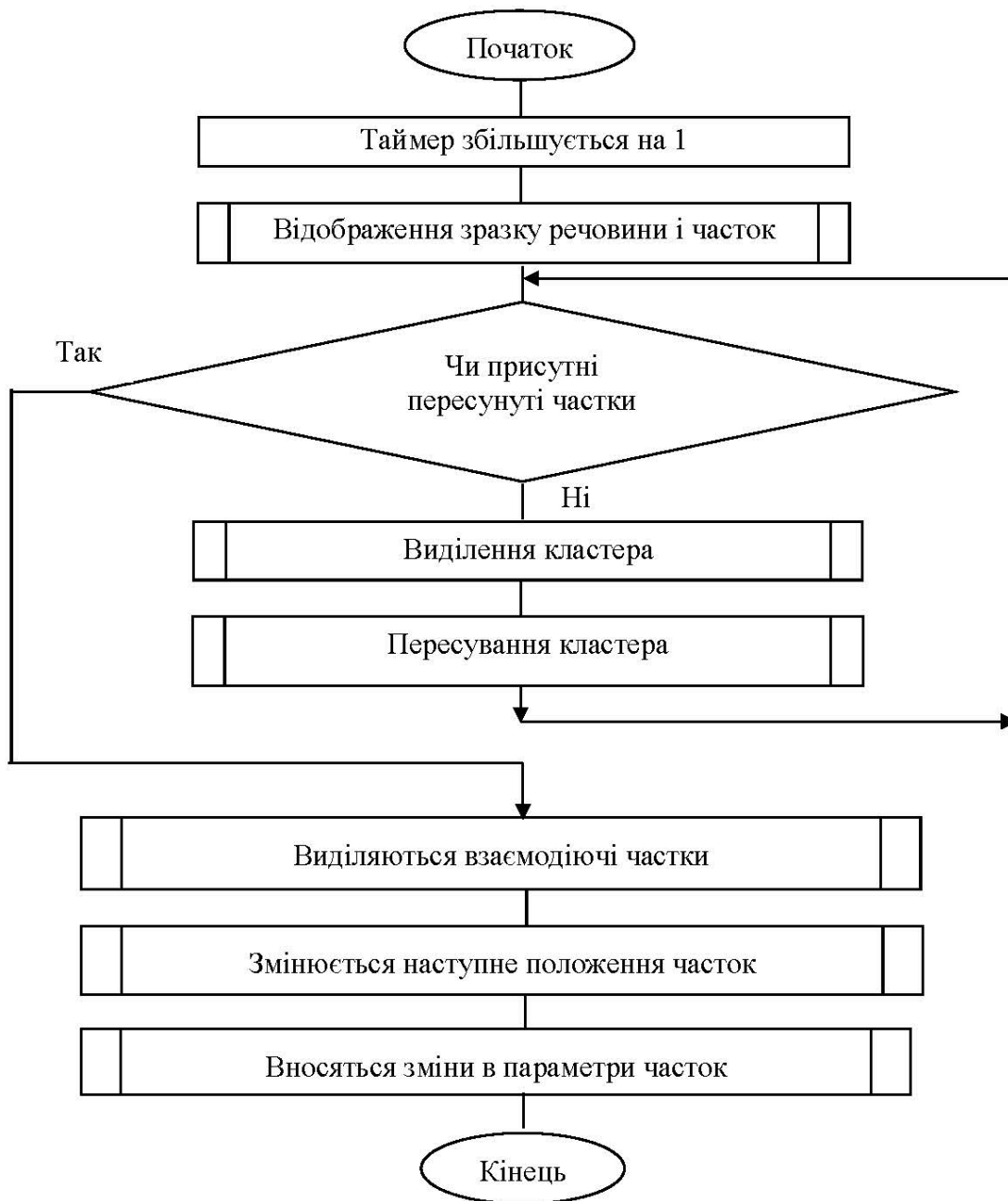


Рисунок 1.7. Блок-схема алгоритму роботи моделі по тактам

Необхідно зауважити, що модель дозволяє задавати часовий інтервал для автоматичного розрахунку розмірностей, а також динамічно змінювати цей параметр під час проведення експерименту.

1.12 Програмна реалізація моделі взаємодії часток

Базовий клас (*TParticle*) має наступні поля: *X_position*, *Y_position*, *Z_position* – координати центру частки у зразку речовини; *Radius* – радіус частки; *Alpha_move* – кут між проекцією вектора швидкості на площину *XOY* і позитивним напрямком осі *X*; *Zeta_move* – кут між вектором швидкості частки і віссю *Z*; *v_random* – величина випадкової складової швидкості частки; *Centrifugal* – величина відцентрової складової швидкості частки, початкове значення якої дорівнює нулю. Далі її значення визначається, виходячи з маси кластера і його попередніх параметрів. Вектор відцентрової складової завжди спрямований від центру циліндра перпендикулярно задній стінці; *p* – щільність частки, визначається як випадкова величина з діапазону, значення якого вибираються в проміжку $[0.1; 1]$ з кроком 0.1. Головний інтерфейс комп'ютерної моделі представлений на рис. 1.8.

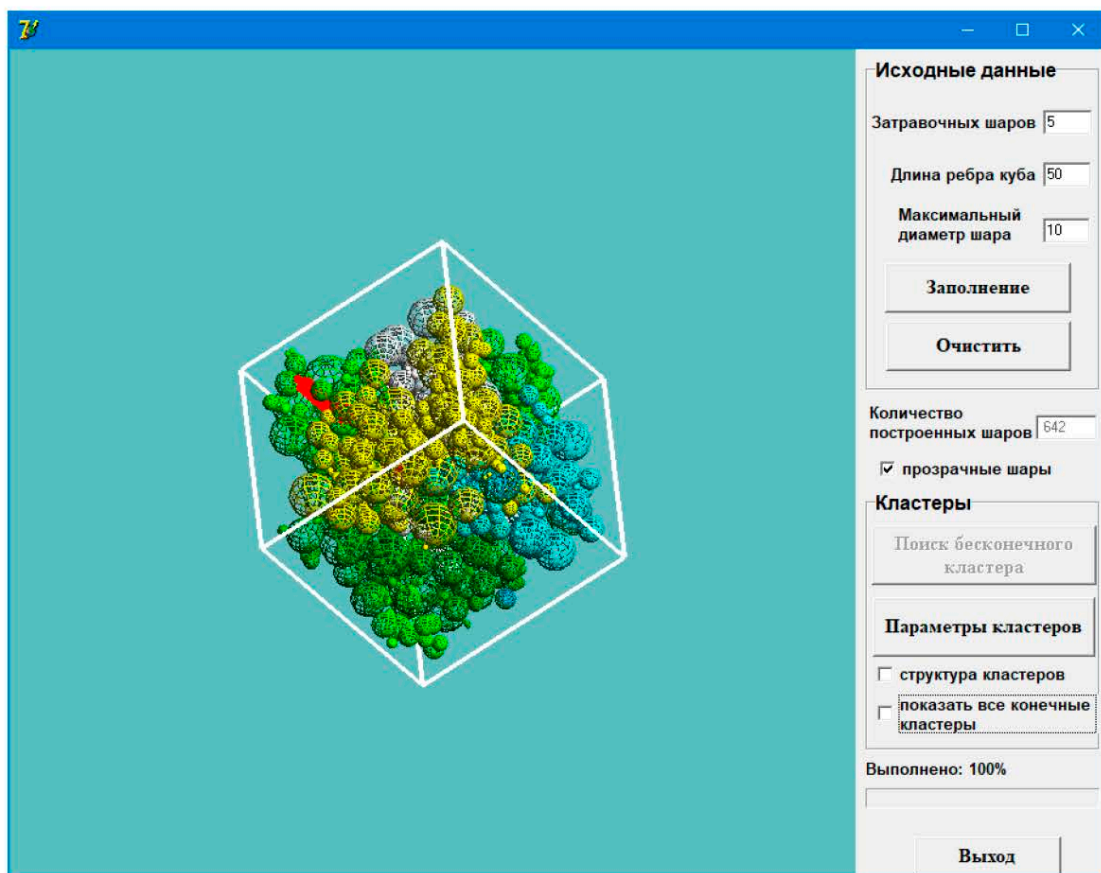


Рисунок 1.8. Головний інтерфейс на етапі завершення моделювання

Зм.	Арк.	№ докум.	Підпис	Дата

КГ 06. 17 000. 00 ДП ПЗ

Арк.

29

На рис. 1.9 показано інтерфейс дочірнього вікна, яке призначене для виведення параметрів КЧ та кластероутворення, розрахованих за методикою, наведеною вище. На рис. 1.10 наведено графік фрактальної розмірності, який виводиться у додатковому вікні.

Параметры кластеров

Параметры бесконечного кластера

Масса кластера 245
 Мощность кластера 0,0020
 Радиус-вектор центра масс кластера 26,33
 Квадрат радиуса гирации для изотропных кластеров 176,42
 Степень анизотропии по осям ху $293,4 : 20,7 = 14,10$
 Степень анизотропии по осям хz $292,5 : 15,7 = 18,60$
 Степень анизотропии по осям уз $31,8 : 17,0 = 1,87$
 Фрактальная размерность 0,94
 Индекс, показывающий изменение доли ячеек, принадлежащих кластеру из z ячеек при изменении z 4,21
 Коэффициент пропорциональности, связывающий число ячеек в кластере и радиус гирации 21,78

Параметры конечных кластеров

Масса кластера Количество
 Мощность кластера 0,0036
 Доля ячеек кластеров, принадлежащих кластерам данной массы 0,0053
 Радиус-вектор центра масс кластера 19,60
 Квадрат радиуса гирации для изотропных кластеров 69,68
 Степень анизотропии по осям ху $15,4 : 8,9 = 1,73$
 Степень анизотропии по осям хz $15,5 : 4,4 = 3,50$
 Степень анизотропии по осям уз $8,9 : 4,6 = 1,92$
 Фрактальная размерность 1,34
 Индекс, показывающий изменение доли ячеек, принадлежащих кластеру из z ячеек при изменении z 3,23
 Коэффициент пропорциональности, связывающий число ячеек в кластере и радиус гирации 26,26

Общие параметры

Вероятность, что выбранный узел принадлежит какому-либо кластеру 0,6880
 Доля ячеек, занятых всеми конечными кластерами 0,6870

Средняя масса конечного кластера 114,00
 Среднее значение радиуса гирации 7,38
 Корреляционная длина (длина связности) 11,49

График фрактальной размерности

Гистограмма распределения конечных кластеров

Закреть

Рисунок 1.9. Інтерфейс модулю аналізу параметрів КЧ у комп'ютерній моделі

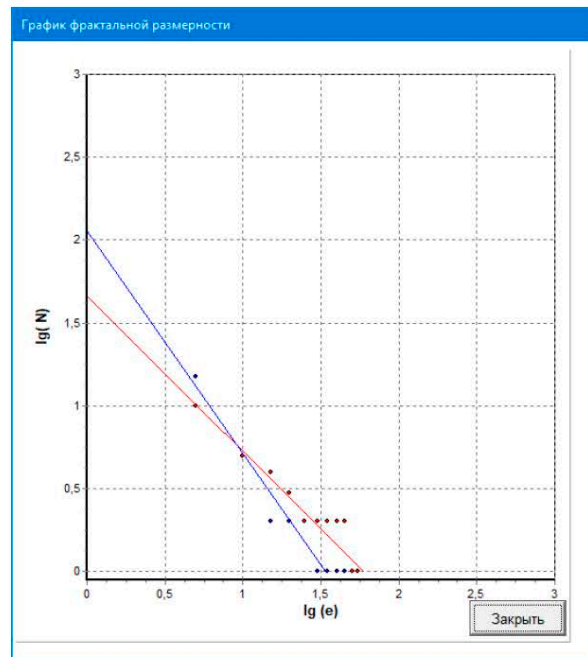


Рисунок 1.10. Додаткове вікно відображення графіку ФР у комп'ютерній моделі

На рис. 1.11 наведено приклад розподілення кінцевих кластерів за розмірами, що виводиться у спеціальному дочірньому вікні.

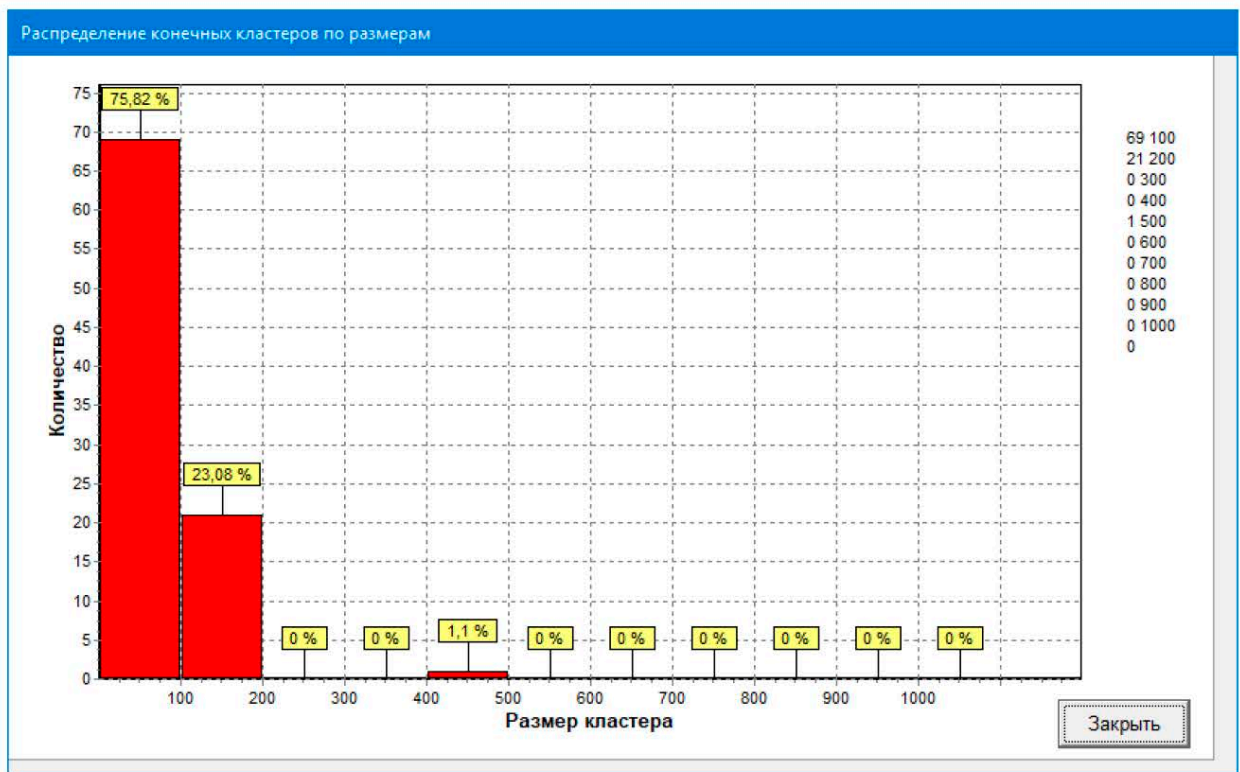


Рисунок 1.11. Додаткове вікно відображення розподілу кінцевих кластерів за розмірами у комп'ютерній моделі

Розподіл часток за розмірами представлено цілочисельними змінними і зберігається в полях даних. Зовнішні фактори, що впливають на систему, (ExternalFactors: TExtFactors) містять такі поля даних: optimalDistanse – найбільша відстань, при якій виникає взаємодія між частками; maxWind – максимально можливе значення переносної швидкості; maxRandom – максимально можливе значення випадкової складової швидкості частки.

Базовий клас TCanal. Модель функціонує у тривимірному просторі. При цьому використовується система координат, яка розташована так: початок осей знаходиться на осі циліндра, вісь OX направлена по осі циліндра вправо, вісь OY – вниз і вісь OZ, відповідно, в площину зображеної проєкції.

Тривимірний простір відповідає зразку речовини і являє собою порожнину, вісь якої в плані – півколо. Отже, необхідно реалізувати алгоритм перевірки на стикування закруглених частин. У разі невиконання цієї умови, можуть виникати

систематичні помилки, які приведуть до вибування часток з експерименту, і, отже, до великих погрешностей проведених розрахунків. Рішення проблеми досягнуто шляхом розрахунку самого радіусу і радіуса кривизни зразку, і їх стикування з іншими ділянками. Радіус кривизни зразку обмежений граничним значенням, яке знаходиться виходячи з радіуса зразку.

На рис. 1.12 представлена коротка діаграма взаємодії базових класів в розробленій комп'ютерній моделі.

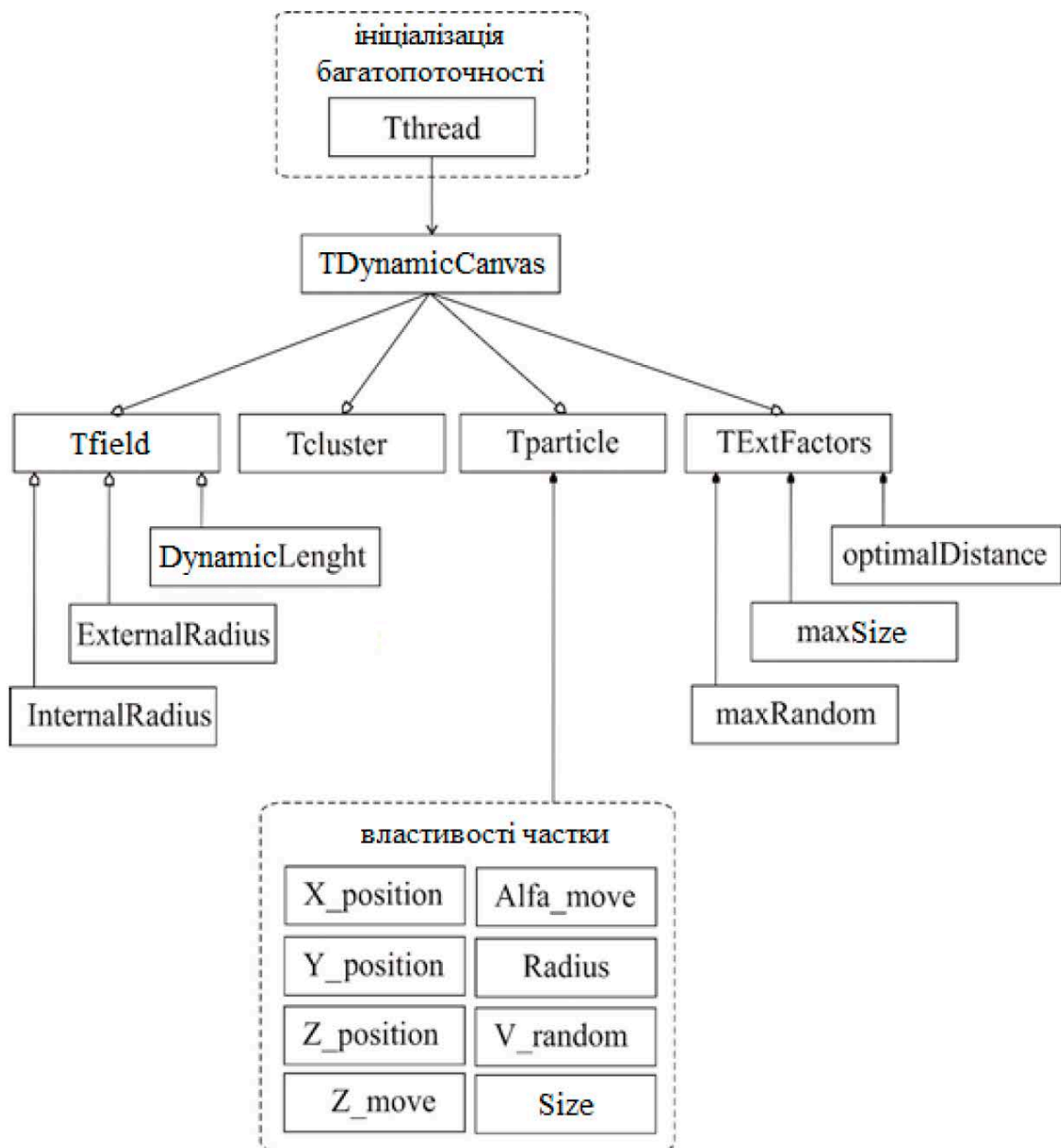


Рисунок 1.12. Коротка діаграма взаємодії базових класів у комп'ютерній моделі

Базовий клас TCluster. Для опису кластерів, як об'єктів, розроблені алгоритми їх візуалізації. Зокрема, використовуючи бібліотеки OpenGL (Open Graphic Language), були реалізовані форми, що відображають склад кластера.

Вибір OpenGL обґрунтований тим, що це найбільш поширений стандарт, який розроблявся і позиціонувався для відображення тривимірної графіки. Він підтримується всіма системами і апаратними засобами.

У додатку наведено вихідний код головного модуля ПЗ.

1.13 Реалізація алгоритмів розрахунку розмірностей

Алгоритм розрахунку розмірностей виділений в окремий програмний модуль, який є глобальним, тобто укладений в розділ глобальних змінних головного модуля програми і використовується іншими модулями моделі. Модуль з розрахунку спектра розмірностей для тривимірних об'єктів – кластерів, містить методи для розрахунку результатів обчислень.

Вхідним параметром модуля розрахунку розмірностей є вказівник на масив об'єктів, що описують частки, з яких складається кластер.

Вихідними даними модуля є побудова точок і подальша апроксимація графіків залежностей. Крім того, на виході формуються файли двох видів, що містять необхідні результати розрахунку статистичних показників на основі проведених модельних експериментів.

Розрахунок спектра розмірностей можна умовно розділити на два етапи: визначення фрактальної розмірності (розмірність Реньї нульового ступеня) і визначення розмірностей Реньї ненульового ступеня (максимальна, інформаційна, ряду кореляційних і мінімальна).

Розрахунок фрактальної розмірності проводиться методами сфер і палетки, реалізація яких винесена в дві окремі функції. Алгоритм розрахунку методом палетки наступний:

1. Тривимірний об'єкт, що розглядається, вписується в паралелепіпед, параметри якого розраховуються на основі вхідних даних;

2. Отриманий паралелепіпед розбивається на кубічні комірки з деяким заданим значенням довжини його ребра;
3. Проводиться підрахунок комірок, які містять в собі будь-яку частину об'єкта. Для скорочення перебору при визначенні наявності об'єкта в даній комірці, був розроблений алгоритм, завдяки якому розглядаються тільки об'єкти, що знаходяться в деякій невеликій околиці заданої комірки, а не всі об'єкти кластера;
4. Отримане число комірок, що містять об'єкт і довжина ребра, зберігаються в динамічно створюваному масиві структур спеціального формату для подальшої обробки;
5. Довжина ребра комірки палетки збільшується на деяке задане значення;
6. Пункти 2-5 повторюються, поки довжина ребра не досягне верхнього граничного значення, обчислення якого проводиться автоматично. Значення залежить від отриманих параметрів паралелепіпеда, що охоплює об'єкт;
7. Значення розмірів комірок і відповідної їм кількості заповнених комірок можна представити у вигляді пар координат на площині, внаслідок чого виходить певний набір точок для двовимірної декартової системи координат. Апроксимація відбувається в подвійному логарифмічному масштабі. Значення кутового коефіцієнта прямої, що проходить через ці точки, є значенням фрактальної розмірності.

Роботу функції розрахунку фрактальної розмірності на основі методу сфер можна описати таким чином:

1. Визначається початкове значення радіуса сфери, центр якої розташовується в центрі мас кластера;
2. Будується сфера з заданим радіусом;
3. Визначається частка обсягу всього об'єкта, що міститься в побудованій сфері;
4. Отримане значення частки обсягу і радіусу зберігаються в динамічному масиві структур спеціального формату для подальшої обробки;

5. Значення радіусу збільшується на заздалегідь задане значення;
6. Пункти 2-5 повторюються до тих пір, поки величина радіуса не досягне величини радіуса сфери, яка описує об'єкт;
7. Отримані значення величин радіусів і відповідних їм часткам охопленого обсягу досліджуваного об'єкта можна представити у вигляді пар координат на площині, внаслідок чого буде отримано деякий набір точок для двовимірної декартової системи координат. Значення кутового коефіцієнта прямої, проведеної через ці точки, методом найменших квадратів дорівнює фрактальній розмірності.

Розрахунок всіх інших розмірностей відбувається за таким самим алгоритмом. Відмінності полягають тільки в значеннях деяких констант, що визначаються показником ступеня. Процедуру розрахунку розмірностей можна описати таким чином:

1. Задається початкове значення ребра палетки;
2. Будується палетка з заданим значенням ребра комірки;
3. Для кожної комірки обчислюється ймовірність наявності досліджуваного об'єкта в ній;
4. Далі обчислюється значення суми, отримані значення і значення ребра палетки зберігаються в динамічному масиві структур спеціального формату для подальшої обробки;
5. Далі довжина ребра комірки палетки збільшується на деяке задане значення.

Пункти 6 і 7 ідентичні пунктам для розрахунку фрактальної розмірності методом палетки.

Слід зазначити, що для розрахунку ймовірностей необхідно було вирішити задачу знаходження обсягу частин сфер, отриманих внаслідок їх відсікання площинами куба від сфер, що стоять поруч. Для вирішення задачі, що виникла, була реалізована функція знаходження ймовірності, що використовує метод Монте-Карло [6] для підрахунку шуканого обсягу. Виходячи з того, що побудова палетки, в разі розрахунку будь-яких розмірностей однакова, то щоб уникнути

повторних циклів і громіздких алгоритмів, розрахунки для всього спектру розмірностей проводилися паралельно на кожній ітерації. Побудова графіків залежностей здійснювалася за допомогою стандартного компонента (TChart). Результати розрахунків представлені у вигляді двомірної декартової системи координат. Для кожної з розмірностей будуються експериментальні точки і пряма апроксимації. Для розрахунку прямої апроксимації використовується процедура, розрахунки якої засновані на методі найменших квадратів.

Форма подання графічних результатів також надає можливість ручного вибору точок, на підставі яких буде проводитися апроксимація, а, отже, і розрахунок відповідної розмірності.

Отримані значення розмірностей, а також вихідні дані моделювання (розподіл часток за розмірами, параметри зразку речовини і зовнішніх чинників) вивантажуються в файл ініціалізації для забезпечення зручного перегляду незалежно від програми. Крім того, всі кінцеві результати і необхідні проміжні дані зберігаються у файлі спеціального формату (*.dat), який використовується для накопичення статистики моделювання з метою подальшого їх використання в модулі розрахунку статистичних параметрів моделі.

Для обчислення узагальненого спектру розмірностей Реньї великих кластерів, до складу яких входять 40 і більше часток, застосовуються алгоритми багатопоточності. Наприклад, використовуючи базовий клас (TThread), розрахунок розмірностей Реньї проводиться в 7 потоків, що дозволило прискорити процес розрахунку розмірностей. Варто відзначити, що клас (TThread) дещо розширено додатковими функціями, які не входять до нього спочатку.

У якості доповнень можна вважати автоматичне визначення необхідної кількості ресурсів і потоків для проведення деяких розрахунків. Використовуючи розроблений в програмі компонент (TOptimum), що виділяється під деяку дію, потік самостійно визначає потрібну кількість оперативної пам'яті. Використовуючи розширені засоби розробки WinApi 32, так звані пакети SDK (Service Development Kit), подається запит в систему. При наявності вільних

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

ресурсів система виділяє запитувану кількість пам'яті і створює єдиний батьківський потік.

Далі відбувається породження дочірніх потоків. При такій взаємодії моделі і ОС робота відбувається безпосередньо з оперативною пам'яттю (ОП), що надає швидкодію моделі. Виділення моделі окремого і незалежного процесу схоже на конвеєр. Слід зазначити, що при запуску виконуваного файлу програмного продукту, йому відразу присвоюється найвищий пріоритет. А зважаючи на те, що обробка відбувається конвеєрним способом, відповідно, запити, які подаються моделлю в систему, виконуються першими. Як зазначалося вище, такий спосіб більш ефективний, але менш надійний. Його ефективність полягає в тому, що швидкодія обмежена лише завантаженістю ОС і кількістю оперативної пам'яті. А ненадійність полягає в тому, що в разі будь-якого збою в оперативній пам'яті автоматично відбувається і збій програмного продукту.

У разі якщо система дає відмову на виділення ресурсів, відбувається звернення до файлу підкачки, який автоматично створюється при завантаженні системи. Такий метод запитів менш ефективний, хоча більш надійний, тому що всі критичні параметри моделі завантажуються у файл підкачки, потім переміщуються у відповідну частину реєстру, після чого встановлюються незалежним процесом. При такій взаємодії програмний комплекс використовує ресурси і швидкодію жорсткого диска (або SSD-накопичувача).

Обробка малих кластерів відбувається на апаратному рівні і, відповідно, не вимагає багато ресурсів системи. Алгоритм їх обробки проводиться безпосереднім зверненням до ОС і, вибравши оптимальну кількість ресурсів, ОС виконує обробку.

Не менш важливою проблемою, яка потребує вирішення при проектуванні багаторівневих систем, є автоматизація. Це пов'язано з тим, що представлений для користувача інтерфейс імітаційної моделі вимагає великої кількості часу для того, щоб зробити всі розрахунки. З метою полегшення цього процесу проведено автоматизацію ряду процесів. Зокрема, були автоматизовані процеси розрахунку спектра розмірностей, як для стінки зразку речовини, так і для кластерів. Ця

процедура також торкнулася і розрахунку напруженостей кластера і його подальшого розпаду.

Такого ефекту вдалося домогтися шляхом розробки процедур, що відстежують дії користувача. Зокрема, розрахунок спектра розмірностей не вимагає багаторазового натискання відповідної кнопки. Потрібно лише раз натиснути функціональну клавішу і розрахунок розмірностей відбувається автоматично.

1.14 Реалізація запису статистичних показників

На етапі проектування системи необхідно було вирішити проблему представлення та збереження статистики. Виходячи із зручності сприйняття, була реалізована система збереження кінцевих і проміжних даних в текстових файлах, а вхідні параметри в файлах ініціалізації. Такий підхід дозволяє скоротити час на збір і обробку статистики [7].

У текстовий файл заносяться дані про дисперсію часток і кластерів, отримані розмірності, а також деяка інформація про можливий розпад кластера. Розроблені алгоритми дозволяють згодом адаптувати виведення статистики будь-якого характеру.

Файл ініціалізації створюється при першому запуску системи і зберігає в собі інформацію про умови, з якими система почала функціонування. Зручність використання файлу ініціалізації полягає ще й в тому, що початкова і проміжна інформація не заносяться в зовнішні джерела, а взаємодіють всередині програми, що дозволяє зберегти необхідну швидкодію.

Файл ініціалізації і текстовий файл статистики є динамічними змінними. Їх створення відбувається в момент запуску моделі, а зберігання здійснюється в поточній директорії програмного продукту. Ці структури позначаються прапорами «тільки для читання» і надалі не можуть бути змінені без стороннього втручання. На відміну від файлу статистики, файл ініціалізації є тимчасовою змінною. Тому після завершення роботи моделі він автоматично видаляється з поточного каталогу.

Всі виконувані процедури програмно вбудовані і запускаються, використовуючи метод, прив'язаний до події OnExecute об'єкта RASave, класу TAction. Активізація вищевказаного набору процедур, що включають в себе процедуру виклику стандартного діалогу збереження файлу (що викликає процедуру) і процедуру безпосереднього збереження файлу статистики (виконуючу процедуру) відбувається при натисканні відповідної кнопки.

Викликаюча процедура відповідає за активізацію діалогу, створення файлу, виклик виконуючої процедури і закриття файлу. Оскільки діалог збереження файлу є елементом ActiveX, контроль над коректністю імені файлу і шляху, а також всі складнощі, які можуть виникнути з правами доступу в файловій системі NTFS, покладаються на саму операційну систему. Також в процедурі реалізовані виведення спеціального заголовка в файл і виведення спеціальних завершальних символів для забезпечення сумісності з іншими редакторами системного реєстру.

Виконуюча процедура використовує методику рекурсії для перебору імен та даних всіх підрозділів і значень. Дані записуються в файл двобайтовими символами в форматі Unicode.

1.15 Реалізація модуля для розрахунку ступеня впорядкованості зображень

Аналіз ступеня впорядкованості зображення слідів на стінці алгоритмічно реалізований шляхом подання відбитків стінки не тільки як чорно-білий об'єкт, але і в градаціях сірого кольору. Фіксація відбитка реалізована, використовуючи стандартну функцію OpenGL – ellipse. Отримання об'єкта певного кольору використовує розроблену функцію *DrawByPixels* ($x_c, y_c, r: double; mode: char$), де x_c, y_c – координати центру залишеної плями, r – радіус плями, а змінна *mode* є параметром, що визначає режим кольору об'єкту – Grayscale або Black and white. Цей параметр може набувати таких значень прапорів: 1, 2, 3, які відповідають певному радіусу сліду.

При ударі кластерів об стінки, функція *DrawByPixels* додає кожному пікселю відповідного відбитка чверть значення білого кольору, в результаті чого

зображення стінки може складатися з пікселів п'яти кольорів: чорний (значення 0), сірий (в діапазоні значень 0.25, 0.5, 0.75) і білий (значення 1). Такий спосіб відображення дає можливість побачити накладення декількох відбитків один на одного і, згодом, аналізувати кількість взаємодій зі стінками за кількістю білого кольору.

Зважаючи на те, що функція *DrawByPixels* розраховує характеристики тільки для однієї плями, був реалізований алгоритм рекурсивного розрахунку всіх слідів, що залишаються твердими частками і кластерами.

При виведенні кольорових зображень на екран проводиться автоматичне збереження аналізованих об'єктів. Зображення стінки з відбитками зберігається в форматі *.bmp в директорію BWFiles, створювану автоматично в поточному каталозі основного програмного продукту. Для збереження зображення необхідно викликати команду збереження, шляхом натискання на відповідну кнопку «Зберегти зображення». Обробка цієї події включає в себе наступні дії:

1. Розраховується розподіл білого кольору;
2. Відбувається перерахунок розподілу різниці кольорів сусідніх пікселів;
3. Відбувається нормування результатів, отриманих в пункті 2;
4. Дані з пунктів 1-3 зберігаються в масиві структур даних розподілу градацій сірого;
5. Тимчасові файли зображень маркуються як постійні і зберігаються в тій же директорії.

Порівняння зображень одне з одним можливо тільки для випадку, коли зображення відбитків на стінці, яке порівнюється, збережене раніше і має статус постійного файлу. Аналіз і порівняння зображень можна провести шляхом вибору зі списку вже збережених зображень і вибором відповідного пункту меню. Алгоритм роботи цієї функції виглядає наступним чином:

1. Після запуску форми, що відображає порівнювані зображення, відбувається ініціалізація всіх критичних характеристик;
2. Розраховується відносна різниця в кількості білого кольору між двома зображеннями;

3. Зображення з меншою кількістю білого кольору множитья на коефіцієнт, що розраховується програмно для зрівнювання з другим зображенням;
4. Відбувається виведення порівнюваних зображень в градаціях сірого кольору, а також зображення, отриманого після зрівнювання кількості білого кольору.

1.16 Реалізація алгоритмів оцінки хаотизації часток

Ступінь хаотизації часток вимірювалася трьома способами. У програмному продукті присутній алгоритм розрахунку ступеня стохастичності потоку часток. Використовуючи компонент Checkbox, вдалося вирішити проблему використання великої кількості додаткових форм. При кожній зміні цього параметра динамічно змінюються графіки, відображаючи відповідну інформацію. Для активації/деактивації необхідного графіка використовувалися прапори і маркери.

Розширивши програмним шляхом можливості компонента TChart зі стандартного інструментарію, вдалося домогтися максимальної швидкодії компонента. Для цього був реалізований ряд алгоритмів, що динамічно відстежують і розставляють відповідні точки на графіку. Уповільнення роботи компонента обумовлено тим, що при заповненні області координатних осей, кількість присутніх точок досягало близько 1,5 млн. Інтерфейс цієї форми складається з 4 графіків, три з яких показують зсув часток щодо осей OX, OY і OZ відповідно, а четвертий графік відображає зміщення другий частки відносно першої. По осях OX відкладено значення кількості тактів, а по осях OY – максимальна величина зсуву частки в певний такт часу. Для визначення розходження/збіжності потоку часток реалізована система обмеження, яка показує область, виходячи за яку, частки потоку розходяться. Статистичний збір інформації для подальшої обробки здійснюється шляхом рекурсивної перевірки заданих полів. Дані зберігаються в двійковому коді. Це реалізовано для того, щоб модель могла зіставляти і знаходити необхідну інформацію самостійно, використовуючи закладені алгоритми. Статистичні показники стінки і кластера

заносяться в різні файли BackWallData і ClusterData, відповідно. Створення файлів відбувається динамічно з першим тактом. Варто зазначити, що створені файли не будуть видалені в разі, якщо користувач занесе в них якусь інформацію. Інакше, після завершення роботи програми файли будуть автоматично видалені з поточної директорії. Це зроблено для того, щоб вирішити проблему накопичення великої кількості помилкових статистичних фалів. У разі, коли програмний комплекс «виявить» при запису, що такі файли вже існують, буде виведено відповідне вікно з повідомленням про наявність таких файлів в папці з повною інформацією про них. Також буде запропонований один із таких способів: змінити ім'я поточних файлів, перезаписати попередні файли або додати нову інформацію в існуючі файли. Такий підхід допомагає вирішити проблему помилкового видалення необхідних файлів статистики і, в той же час, накопичувати статистику для подальшої обробки. Реалізований метод накопичення статистики дає можливість користувачеві отримувати інформацію по мірі необхідності, а не відразу проводити громіздкі обчислення за один сеанс.

1.17 Взаємодія функціональних компонент реалізованої моделі

Як зазначалося вище, головним вікном програми є об'єкт класу TFormMain. У методах цього класу містяться процедури створення всіх дочірніх вікон програм крім вікон розрахунку всіляких параметрів. На рис 4.6 представлена коротка діаграма взаємодії функціональних компонент імітаційної моделі.

Також членами цього класу є всі об'єкти класів TAction, TImageList, TMainMenu, TToolBar і всі системні діалоги. Спосіб завершення роботи програми реалізований з використанням механізму системних повідомлень Windows (Windows Messaging). У класі головного вікна програми TFormMain за допомогою макросу BEGIN_MESSAGE_MAP описана подія отримання вікном стандартного повідомлення ОС WM_CLOSE, а пункт меню «Вихід» просто посилає дане повідомлення головного вікна.

					КГ 06. 17 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

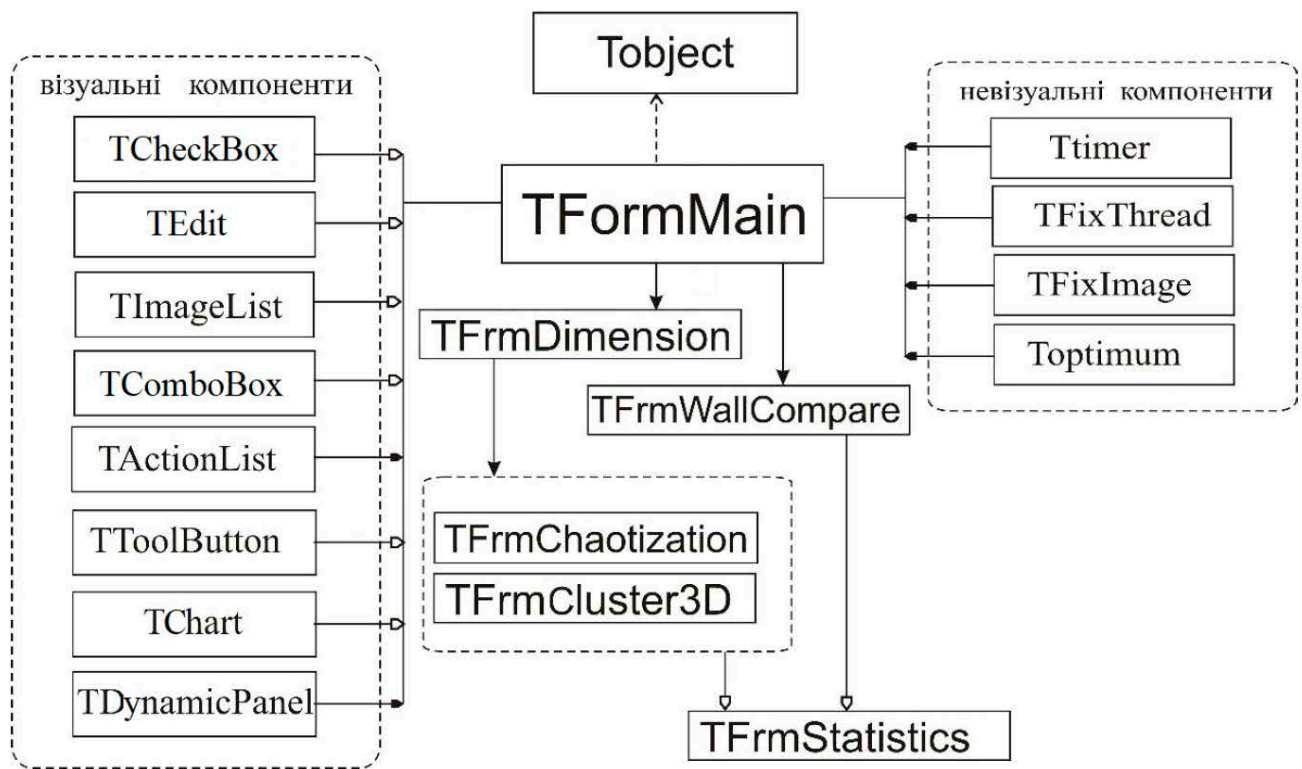


Рисунок 1.13. Коротка діаграма взаємодії функціональних компонент

Таким чином забезпечується простий, ефективний і надійний метод, що дозволяє контролювати завершення роботи, виключаючи випадкове завершення роботи моделі.

Зважаючи на те, що процес моделювання відбувається з використанням багатовіконного режиму, інтерфейс програмного продукту побудований на базі технології MDI (Multi Document Interface).

Для виконання всіляких маніпуляцій використовується клас TfrmSecondary, який містить в собі практично всі методи курування і всі методи відображення даних. Так була досягнута простота керування і доступний інтерфейс програми.

Коли модель знаходиться в режимі очікування, ОС надсилає системне повідомлення WM_APPIDLE. Всі об'єкти класу TAction обробляють це повідомлення і при отриманні викликають подію OnUpdate (TAction.OnUpdate), що дозволяє в цей момент проводити дії, спрямовані на зміну елементів керування в залежності від поточного стану даних. Модель використовує цю подію для того, щоб зробити доступними або недоступними деякі елементи керування.

Зважаючи на те, що модель передбачає розпад кластера на кілька частин, необхідно було реалізувати досить громіздкий алгоритм розрахунку напруженостей. При цьому була поставлена задача зберігати швидкодію імітаційної моделі.

Для реалізації візуальної частини був обраний OpenGL. Апаратна частина реалізована з використанням класів та динамічних структур, що дозволило добитися максимальної швидкодії системи і при цьому не втратити її функціональність. При цьому необхідно зазначити, що візуалізація процесу розпаду, як окремого класу не залежить від розпаду безпосередньо в моделі.

У моделі розпад відбувається наступним чином. Після старту системи, кластера розподіляються за розмірами до і після розпаду. У разі виявлення кластера, що включає не менше 30 часток, відбувається розрахунок його напруженостей. Розрахунок цієї характеристики здійснюється всередині кластера. І, якщо значення дорівнює заданому, відбувається розпад кластера на кілька частин. Перерахунок цього параметра відбувається на кожному наступному такті.

Для визначення того, як і на які частини розпадається кластера, був застосований наступний алгоритм.

Координати отриманого кластера перевизначаються для моделі розрахунку напруженостей. Далі визначається куб, який описує кластер. Потім в кожній точці отриманого куба з заданим кроком розраховуються і будуються вектора напруженості. Серед отриманих значень знаходяться максимальне і мінімальне значення. Після цього будується площина перпендикулярно відрізка, що з'єднує точки максимальних і мінімальних напруженостей, ділить відрізок на дві рівні частини.

Отримана площина і є роздільником, відносно якого відбувається розпад кластера. Частки, що знаходяться по різні сторони цієї площини, відриваються одна від одної і рухаються як окремі кластери.

Слід зазначити, що комп'ютерна модель дозволяє варіювати інтервал розрахунку напруженостей і критичний розмір кластера, при якому до нього застосовується алгоритм розпаду.

1.18 Реалізація алгоритмів обробки помилок

Програмний комплекс складається з безлічі модулів, форм, графіків, діаграм і гістограм. Кожна подія обробляється на програмному рівні. У разі помилкового введення параметрів, обробник помилок перехоплює дію і виводить відповідне повідомлення.

Складність реалізації обробника помилок полягає в тому, що не завжди можна передбачити, де програма може некоректно розрахувати або прийняти дані. Для усунення цього недоліку був проведений ряд експериментальних досліджень, який показав, що, використовуючи структуру *try* і *except* можна обійти «вузькі місця» в системі. Таким чином, імітаційна модель контролює введення вхідних параметрів.

Для того, щоб відобразити динаміку руху часток і кластерів на формі, було розглянуто ряд компонент як сторонніх розробників, так і стандартних компонент з палітри Borland Delphi 7. Встановлено, що для відображення динамічних об'єктів, що рухаються в тривимірному просторі, немає достатньої кількості характеристик, щоб описати цю дію. В рамках моделі, було прийнято рішення розробити свій компонент і впровадити його в модель.

Розроблений компонент і відповідний йому клас (TDynamicCanvas) дозволяє описати всі визначені для імітаційної моделі властивості. Зокрема, рух по трьох осях координат, фіксування ударів часток потоку об межі конструкції (меж розподілу зразку речовини), розрахування кількісних характеристик одержуваних об'єктів системи. Існує можливість додавання додаткових параметрів, що істотно підвищує універсальність моделі в цілому.

Компонент розроблявся таким чином, щоб вирішити проблему мерехтіння і відтворення зображення на екрані. Розроблені функції, процедури і методи дозволяють відображати те, що відбувається в каналі без мерехтіння і «втрати» зображення. Проведення оптимізації показало, що для досягнення максимальної швидкодії і коректної роботи необхідне використання конструкторів («constructor») і деструкторів («destructor»).

Для виведення попереджуючих повідомлень використовуються процедури, які звертаються до функцій WinAPI32. Практична реалізація цієї структури наведена у фрагменті наступного коду:

```
...
try
pointsFilled: = 0; // кількість початкових часток;
for i: = 0 to length (sizes) -1 do // цикл запуску заданої кількості часток
except
pointsFilled: = pointsFilled + sqr (sizes [i] .radius) * pi * sizes [i] .number;
MessageDlg ( 'Для стабільної роботи програми необхідно задати менше
число часток', mtWarning, [mbOk], 0);
if not CheckFilling then MessageDlg ( 'Не всі критичні параметри
задані', mtWarning, [mbOk], 0)
...
```

Такий підхід допомагає вирішити проблему, по-перше, сумісності програмного продукту з іншими версіями ОС і компіляторів, по-друге, вирішує проблему переносимості моделі на інші платформи і, по-третє, виключає можливість отримання помилкових значень уникаючи тупикових ситуацій. Слід зазначити, що дана структура є стандартом для всіх компіляторів.

Зважаючи на те, що моделювання стохастичної системи вимагає чималих обчислювальних потужностей і машинного часу, була проведена оптимізація не тільки процесів візуалізації, але і розрахункової частини програми. Шляхом заміни вкладених циклів на їх аналоги, що виконуються рекурсивно, був скорочений час виконуваних розрахунків майже втричі.

1.19 Автоматизація роботи у реалізованій моделі

Для того, щоб автоматизувати і прискорити роботу користувача з імітаційною моделлю, реалізовано систему «гарячих» клавіш. Це дає можливість скоротити час пошуку і натискання функціональних клавіш моделі. Для програмної реалізації цих функцій використовувалися засоби перехоплення

					КГ 06. 17 000. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

переривань від клавіатури. Необхідною умовою реалізації цієї функції було наступне. Необхідно було створити перелік таких «гарячих» клавіш, які б не використовувала сама ОС. Інакше, трапилася б тупикова ситуація, яка привела б до колізії ОС і моделі. З цією метою реалізовано низку перевірок і, зокрема, розроблено окремий модуль, який дозволяє користувачеві самому налаштувати і коригувати комбінації «гарячих клавіш».

За замовчуванням прийняті такі позначення «гарячих клавіш»:

Alt + P – введення вхідних параметрів для класу TParticle;

Ctrl + K – введення початкових параметрів для класу TChannel;

Ctrl + C / P – копіювання і вставка параметрів, що дублюються;

Ctrl + Break – переведення ПП у режим очікування (пауза);

Ctrl + S – повна зупинка роботи програми;

Ctrl + E – активізація форми розрахунків розмірностей кластера і стінки;

Ctrl + W – розрахунок розмірностей стінки з відбитками;

Ctrl + Shift + K – розрахунок розмірностей кластера;

Ctrl + G – розрахунок напруженостей кластерів.

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

2 ЕКОНОМІЧНА ЧАСТИНА

2.1 Резюме

В даному дипломному проекті реалізоване моделювання перколяційних явищ програмними засобами.

У зв'язку з необхідністю вивчення перколяційних явищ на самих ранніх стадіях зародження і зростання кластерів і труднощами, з якими стикаються дослідники в своїй роботі, великі надії покладаються на імітаційне комп'ютерне моделювання.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , ум. машинних команд
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП імітаційного моделювання	7800 – 8800

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл.2.2.

Таблиця 2.2. Норми часу

Обсяг ПП, тис.умов.машин. команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357
8.00	385
9.00	414
10.00	445

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7\div 0,8$): $T_{ар} = 306 \times 0,8 = 244,8$ (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a \times L_1 \times K_H \quad (2.1)$$

$$T_{ПП} = T^a \times L_2 \times K_H \quad (2.2)$$

$$T_{РП} = T^a \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.3.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.)

Таблиця 2.3. Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПЗ

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L ₁)	0,15	0,12	0,12
ТП (L ₂)	0,16	0,15	0,11
РП (L ₃)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K _н
А	Принципово нові ПЗ	1,75 – 1,2
Б	ПЗ – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПЗ маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПЗ типовими програмами, %	Значення K _т
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T_a * L_1 * K_n = 244,8 * 0,12 * 0,8 = 23,50 \text{ (люд/годин)} \quad (2.4)$$

Трудомісткість розробки технічного проекту

$$T_{ТП} = T_a * L_2 * K_n = 244,8 * 0,15 * 0,8 = 29,37 \text{ (люд/годин)} \quad (2.5)$$

Трудомісткість розробки робочого проекту

$$T_{РП} = T_a * L_3 * K_n * K_t = 244,8 * 0,58 * 0,8 * 0,7 = 79,51 \text{ (люд/годин)} \quad (2.6)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{ТЗ}=3$ (стр), розробка ТП $N_{ТП}=19$ (стр), розробка робочого проекту $N_{РП}=25$ (стр), пояснювальна записка відповідно $N_{ПЗ}=36$ (стр)

Розрахунок зведений у таблицю 2.6.

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
	1.ТЗ	$T_{РТЗ}=23,50$	$T_{КК}=0,7*N_{ТЗ}= 0,7*3=2,1$
2.Розробка ТП	$T_{РТП}=29,37$	$T_{КК}=0,7*N_{ТП}=0,7*19=13,3$	$T_{НК}=0,15*N_{ТП}=0,15*19=2,85$
3.Розробка РП	$T_{РРП}= 79,51$	$T_{КК}=0,7*N_{РП}=0,7*25=17,5$	$T_{НК}=0,15*N_{РП}=0,15*25=3,75$
4.Розробка ПЗ	$T_{РПЗ}=1,5*N_{ПЗ}= 1,5*36 =54$	$T_{КК}=0,7*N_{ТЗ}=0,7*36=25,2$	$T_{НК}=0,15*N_{ПЗ}=0,15*36 =5,4$
Усього, в т.ч.:	256,93		
- на розробку	$\Sigma T_p=186,38$		
- контроль керівника		$\Sigma T_{КК}=58,1$	
- нормоконтроль			$\Sigma T_{НК}=12,45$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, загальні витрати на розробку ПП. Розрахунок основної заробітної плати виконавців приведений у таблиці 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2023» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року – 6700 гривень; мінімальну погодинну тарифну ставку – 40,46 грн.

Таблиця 2.7 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	186,38	45,00	8387,10
2.Контроль керівника	58,1	100,00	5810,00
3.Нормоконтроль	12,45	100,00	1245,00
Усього	-	-	$\Sigma_{\text{зо}} = 15442,10$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку ПЗ

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	85	3.0	255,0
Разом	-	-	-	$V_{Mi}=255,0$
Транспортно – заготівельні Витрати (10%)				$V_{тр\ з} = 0,1 \times V_{M1} = 0,1 * 255 = 25,50$
Усього				$V_M = V_{Mi} + V_{тр\ з} = 280,50$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9.

Таблиця 2.9 Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	280,50	V_M (див. табл. 2.8)
2. Основна заробітна плата	15442,10	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	1544,21	$Z_d = 0,1 \times Z_o = 10723,21 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	3736,99	$V_{е.с.в.} = 0,22 \times (Z_o + Z_d) = 0,22 * (15442,10 + 1544,21)$
5. Накладні витрати	6176,84	$V_{нак.} = 0,4 \times Z_o = 0,4 * 15442,10$
6. Повна собівартість	27180,64	$C_{пов} = V_M + Z_o + Z_d + V_{е.с.в.} + V_{нак.} = 280,50 + 15442,10 + 1544,21 + 3736,99 + 6176,84$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$\Pi = (C_{п} * P) / 100 = (27180,64 * 10) / 100 = 2718,06 \text{ грн} \quad (2.7)$$

де p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$C_o = C_{п} + \Pi = 27180,64 + 2718,06 = 29898,70 \text{ грн} \quad (2.8)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$C_p = C_o + ПДВ = 29898,70 + 29898,70 * 0.2 = 35878,45 \text{ грн} \quad (2.9)$$

3 ОХОРОНА ПРАЦІ

Вирішення завдань охорони праці базується на досягненнях ергономіки, наукової організації праці, технічної естетики, гігієни та фізіології праці, психофізіології. Крім того, успіх охорони праці визначається темпами впровадження передової техніки, підвищення рівня механізації і автоматизації виробничих процесів, удосконаленням технології та організації виробництва.

Безпека праці на підприємстві може бути на належному рівні тільки тоді, коли всебічно відповідає вимогам трудового законодавства, державним стандартам України, норм і правил, розроблених для збереження здоров'я працюючих. Важливе місце при цьому належить виконанню організаційних вимог з охорони праці, а також трудовій та виробничій дисципліні працюючих.

Дипломним проектом розглядається питання моделювання перколяційних явищ програмними способами, що передбачає працю з використанням персонального комп'ютера. Тому до розгляду в даному розділі беремо працю програміста та питання забезпечення його здорових умов праці.

3.1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці

На робочому місці розробника програмного забезпечення: підвищений рівень отримуваного електромагнітного випромінювання, статична електрика, високий рівень шуму, несприятливі умови мікроклімату, підвищена напруга на зір та мозок тощо.

Робота з ПК відноситься до розумового виду праці з переважно нерухомим видом виконання. Це при постійному впливі призводить до професійних захворювання зору, рухової системи, нервової системи, перевтомлення.

3.2 Гігієнічні вимоги до виробничого середовища

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою. В процесі роботи з комп'ютером необхідно дотримувати правильний режим.

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

3.2.1 Вимоги до приміщення

Розміщення робочих місць з ВДТ ЕОМ і ПЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено. Площа на одне робоче місце становить не менше 6,0 м², а об'єм – не менше ніж 20,0 м³. У приміщеннях слід щоденно робити вологе прибирання. Вони повинні бути оснащені аптечками першої медичної допомоги. При приміщеннях мають бути обладнані побутові приміщення для відпочинку. Об'ємно-планувальні рішення будівель та приміщень для роботи з ВДТ мають відповідати вимогам ДСанПІН 3.3.2.007-98.

3.2.2 Освітлення

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення, відповідно до ДБН В.2.5-28-2018. У приміщеннях доцільно, щоб вікна були орієнтовані на північ або північний захід. На вікнах повинні бути штора або жалюзі, що регулюють рівень освітленості і захищають від прямого влучення сонячних променів на робоче місце. При кольоровому оформленні виробничих і допоміжних приміщень необхідно враховувати орієнтацію їхніх вікон стосовно частин світу і використовувати гармонійне сполучення кольорів. Для стін і робочих поверхонь використовують мало насичені (основні) кольори, для невеликих помешкань або ділянок, що рідко потрапляють у поле зору працюючих, а також для створення контрастності – кольори середньої насиченості (допоміжні), для маленьких по площі поверхонь – насичені (акценти) – як функціональне фарбування. Стелі у всіх приміщеннях повинні бути білими. Поверхні устаткування в приміщеннях повинні бути матовими або напівматовими, для виключення випадку відблисків світла в очі працюючого, а стіни бути пофарбованими фарбами пастельних тонів.

3.2.3 Шум

Оптимальні показники рівня шумів у робочих приміщеннях конструкторських бюро, кабінетах розрахувачів, програмістів визначаються за ГОСТ 12.1.003-83 Припустимий рівень шуму при розумовій праці, що вимагає зосередженості для програміста, - 50 дБ. Для зменшення шуму й вібрації в приміщенні устаткування, апарати й прилади встановлюються на спеціальні

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

фундаменти й прокладки, що амортизують. Якщо стіни й стелі приміщення є джерелами шумоутворення, вони повинні бути облицьовані звуковбирним матеріалом. У виробничих приміщеннях на робочих місцях мають забезпечуватись оптимальні значення повинні бути облицьовані звуковбирним матеріалом.

3.2.4 Мікроклімат

У виробничих приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря – ДСН 3.3.6.042 – 99 Санітарні норми мікроклімату виробничих приміщень.

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пилу використовують вентиляцію. Механічна вентиляція (кондиціонери, вентилятори і т.д.) залежно від напрямку руху повітряних потоків, може бути витяжною, припливною і припливно-витяжною. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється з нього внаслідок різниці температур і тиску.. Механічна вентиляція забезпечується вентиляторами, що забирають повітря зовні і направляє його до будь-якого робочого місця. або устаткування, а також видаляють забруднене повітря

3.2.5 Вимоги до організації робочого місця працівника

Велике значення має раціональна конструкція та розташування елементів робочого місця, що важливо для підтримки оптимального робочого стану для працівника.

Робочі місця повинні бути розташовані так, щоб у поле зору працюючого не попадали поверхні, що мають властивість віддзеркалювання, вікна освітлювальні прилади. Відеотермінали повинні встановлюватися під кутом 90-100 градусів від вікон, так, щоб світло падало з боку. Робочі місця з ВДТ доцільно розміщати в глибині приміщення. Розташування відео терміналу, при якому працюючий

звернений обличчям або спиною до вікон, неприпустимо при будь-якому способі реалізації загального висвітлення, як прямим, так і відбитим світлом.

Робочий стіл повинен регулюватися по висоті в границях 680-800 мм, а ширина – забезпечувати можливість виконання операцій в зоні досяжності моторного поля. Рекомендовані розміри столу: висота 725 мм, ширина 600-1400 мм, глибина 800-1000 мм. Робочий стілець повинен бути оснащений підйомно-поворотним пристроєм для регулювання висоти сидіння і спинки, а також кута її нахилу. Регулювання кожного параметра повинне вироблятися легко, бути незалежним і надійно фіксуватися.

Розташування екрана ВДТ має забезпечувати зручність зорового спостереження у вертикальній площині під кутом $+30^{\circ}$ до нормальної лінії погляду працюючого.

Клавіатуру слід розташовувати на поверхні столу на відстані 100...300 мм від краю, звернутого до працюючого.

3.3 Пожежна безпека

Пожежна безпека пристрою, що проектується у даному дипломному проекті, має забезпечуватися відповідно до ГОСТ 12.1.004-91 «Пожарная безопасность. Общие требования», а вибухова безпека – у відповідності до ГОСТ 12.1.010-76 «Взрывобезопасность. Общие требования».

Пожежна безпека – це такий стан об'єкта, при якому з регламентованою ймовірністю виключається можливість виникнення й розвитку пожежі та впливу на людей небезпечних факторів пожежі, а також забезпечується захист матеріальних цінностей.

Системи запобігання пожеж, а також протипожежного захисту у сукупності повинні виключати вплив на людей небезпечних факторів пожежі. Пожежна безпека об'єкта забезпечується:

- Системою запобігання пожежі;
- Системою протипожежного захисту;

- Організаційно-технічними заходами.

Всі приміщення повинні бути забезпечені первинними засобами пожежогасіння: пожежним водопостачанням (пожежні крани ПК), пожежні щити з набором пожежного інструменту, вуглекислотними або порошковими вогнегасниками.



Рисунок 3.1. Вуглекислотний вогнегасник

У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефону 101 пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі.

ВИСНОВКИ

Проведене розробленими програмними засобами моделювання перколяційних явищ дозволило отримати певні результати, які можуть бути наведені, проаналізовані та опубліковані на наступних рівнях навчання. Однак виконані протягом дипломного проектування етапи розробки програмної моделі дозволили зрозуміти принципи кластероутворення у перколяційних системах, їх властивості та особливості моделювання засобами об'єктно-орієнтованого програмування.

Розглянуті класичні методи утворення перколяційних кластерів та вивчені модифікації і інтерпретації методів для аналізу кластерів можуть стати основою для подальших досліджень і моделювання кластерних систем.

Створена комп'ютерна імітаційно-статистична модель перколяційних явищ, зокрема – кластероутворення, дозволяє візуалізувати кластерну структуру, аналізувати її функціонування, оцінювати параметри кластероутворення і коригувати вхідні характеристики моделі.

Розглянуті закономірності процесів коагуляції модельних кластерів, дозволяють цілеспрямовано впливати на їх параметри. Отримані кількісні методи прогнозування структури і властивостей модельних кластерів є достатньо універсальними для багатьох перколяційних явищ.

Створене програмне забезпечення для моделювання перколяційних явищ дозволяє виконувати достатньо точну оцінку параметрів перколяційних кластерів. Проведення комп'ютерних експериментів з розробленим програмним застосунком дозволить узагальнити отримані результати. З практичної точки зору такі результати дозволять використовувати подібну комп'ютерну модель у різних галузях (металургія, наноелектроніка, медицина, матеріалознавство), доповнюючи створене середовище іншими налаштовуваними бібліотеками. Такий підхід значно скоротить час підготовки технічних систем, які вимагають великої кількості початкових умов і параметрів.

					<i>КГ 06. 17 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тарасевич Ю. Ю. Перколяция: теория, приложения, алгоритмы: Учебное пособие – М.: Едиториал УРСС, 2002. – 112 с.
2. Ландау Л.Д., Лифшиц Е.М. Статистическая физика. – М.: Физматлит, 2001 г. – 616 с.
3. Тарасевич Ю.Ю. Математическое и компьютерное моделирование. М: Едиториал УРСС, 2003. – 144с.
4. Москалев П.В., Шимтов В.В. Математическое моделирование пористых структур. М: Физматлит, 2007. – 120с.
5. Додонов А.Г., Ландэ Д.В. Живучесть информационных систем. Киев: Наукова думка, 2011. – 256с.
6. Асланов А. М., Ботнарь К.В., Герега А. Н. О корреляции свойств потока и кластеров в модели агрегации частиц // Сборник трудов «Информационные системы и технологии». – Одесса, 2006 – С.105 – 110.
7. Кривченко Ю. В. Комп'ютерне моделювання самоорганізації кластерних систем: залежність структури від генезису та керуючих параметрів / Ю. В. Кривченко // Науковий журнал ЧНТУ «Технічні науки та технології». – 2018. – Вип. 4 (14). – С. 166-174. // ISSN 2411-5363 / ISSN 2519-4569
8. Архангельский, А.Я. Программирование в Delphi: Учебник по классическим версиям Delphi / А.Я. Архангельский. – М.: Бинوم, 2013. – 816 с.
9. Белов, В.В. Программирование в Delphi: процедурное, объектно-ориентированное, визуальное: Учебное пособие для вузов / В.В. Белов, В.И. Чистякова. – М.: РиС, 2014. - 240 с.
10. Кузан, Д.Я. Программирование Win32 API в Delphi / Д.Я. Кузан. – СПб.: ВHV, 2013. – 368 с.
11. Осипов, Д. Delphi. Профессиональное программирование / Д. Осипов. - СПб.: Символ-плюс, 2015. – 1056 с.
12. Санников, Е. Курс практического программирования в Delphi. Объектно-ориентированное программирование – М.: Солон-пресс, 2013. – 188 с.

Фрагмент вихідного коду головного модуля програми для моделювання перколяційних явищ

```

//-----
procedure tfrmmain.findsquare;
var
maxp,i,j:integer;
stotal,spoints,s:double;
x,y,z,r:double;
coverd:boolean;
ri,rj:double;
begin
for i:=0 to length(crosspoints)-1 do
begin
maxp:=i;
x:=crosspoints[i,0];
y:=crosspoints[i,1];
z:=crosspoints[i,2];
r:=crosspoints[i,3];
for j:=i to length(crosspoints)-1 do
if crosspoints[j,3]>r then
begin
x:=crosspoints[j,0];
y:=crosspoints[j,1];
z:=crosspoints[j,2];
r:=crosspoints[j,3];
maxp:=j;
end;
for j:=maxp-1 downto i do
crosspoints[j+1]:=crosspoints[j];
crosspoints[i,0]:=x;
crosspoints[i,1]:=y;
crosspoints[i,2]:=z;
crosspoints[i,3]:=r;
end;//for i
stotal:=pi*udradius.position*udxcasis.position;
spoints:=0;
if length(crosspoints)>0 then
begin
r:=crosspoints[0,3]*(2/3)*(sqrt(5)/3);
spoints:=pi*sqr(r);
for i:=1 to length(crosspoints)-1 do
begin
coverd:=false;
ri:=crosspoints[i,3]*(2/3)*(sqrt(5)/3);
for j:=0 to i-1 do
begin
rj:=crosspoints[j,3]*(2/3)*(sqrt(5)/3);
if (ri<=rj+sqrt( sqrt(crosspoints[i,0]-crosspoints[j,0]) + sqrt(crosspoints[i,1]-crosspoints[j,1]) +
sqr(crosspoints[i,2]-crosspoints[j,2]) ) ) and
(rj>sqrt( sqrt(crosspoints[i,0]-crosspoints[j,0]) + sqrt(crosspoints[i,1]-crosspoints[j,1]) +
sqr(crosspoints[i,2]-crosspoints[j,2]) ) ) then
begin
coverd:=true;
break
end;//if ...
end;//for j ...
if not coverd then
spoints:=spoints+pi*sqr(ri);
end;//for i
end;//if length...

```

```

s:=spoints/stotal;
lbsquare.items.add(formatfloat('#,#####0.00000',s));
spoints:=0;
if length(crosspoints)>0 then
  begin
    r:=crosspoints[0,3]*(sqrt(7)/4);
    spoints:=pi*sqr(r);
    for i:=1 to length(crosspoints)-1 do
      begin
        covered:=false;
        ri:=crosspoints[i,3]*(sqrt(7)/4);
        for j:=0 to i-1 do
          begin
            rj:=crosspoints[j,3]*(sqrt(7)/4);
            if (ri<=rj+sqrt( sqrt(crosspoints[i,0]-crosspoints[j,0]) + sqrt(crosspoints[i,1]-crosspoints[j,1]) +
sqr(crosspoints[i,2]-crosspoints[j,2]) ) ) and
              (rj>sqrt( sqrt(crosspoints[i,0]-crosspoints[j,0]) + sqrt(crosspoints[i,1]-crosspoints[j,1]) +
sqr(crosspoints[i,2]-crosspoints[j,2]) ) ) then
              begin
                covered:=true;
                break
              end;//if ...
            end;//for j ...
          if not covered then
            spoints:=spoints+pi*sqr(ri);
          end;//for i
        end;//if length...
      s:=spoints/stotal;
      lbsquare_g.items.add(formatfloat('#,#####0.00000',s));
      spoints:=0;
      if length(crosspoints)>0 then
        begin
          r:=3;
          spoints:=pi*sqr(r);
          for i:=1 to length(crosspoints)-1 do
            begin
              covered:=false;
              ri:=3;
              for j:=0 to i-1 do
                begin
                  rj:=3;
                  if (ri<=rj+sqrt( sqrt(crosspoints[i,0]-crosspoints[j,0]) + sqrt(crosspoints[i,1]-crosspoints[j,1]) +
sqr(crosspoints[i,2]-crosspoints[j,2]) ) ) and
                    (rj>sqrt( sqrt(crosspoints[i,0]-crosspoints[j,0]) + sqrt(crosspoints[i,1]-crosspoints[j,1]) +
sqr(crosspoints[i,2]-crosspoints[j,2]) ) ) then
                    begin
                      covered:=true;
                      break
                    end;//if ...
                  end;//for j ...
                if not covered then
                  spoints:=spoints+pi*sqr(ri);
                end;//for i
              end;//if length...
            // виводимо результат
            s:=spoints/stotal;
            lbsquare_a.items.add(formatfloat('#,#####0.00000',s));
            end;

            //-----
            procedure tfrmmain.movekoagulant;
            var
            //знайдемо центр мас
            xmc:=0;
            ymc:=0;
            zmc:=0;

```

```

m:=0;//маса кластеру
for i:=0 to length(koagulant)-1 do
  with tpointin(pointsin.objects[koagulant[i]]) do
    begin
      xmc:=xmc+getxpos*getm;
      ymc:=ymc+getypos*getm;
      zmc:=zmc+getzpos*getm;
      m:=m+getm;
    end;
  xmc:=xmc/m;
  ymc:=ymc/m;
  zmc:=zmc/m;
  radius:=0;
  for i:=0 to length(koagulant)-1 do
    with tpointin(pointsin.objects[koagulant[i]]) do
      if radius<sqrt( sqr(xmc-getxpos)+sqr(ymc-getypos)+sqr(zmc-getzpos) ) then
        radius:=sqrt( sqr(xmc-getxpos)+sqr(ymc-getypos)+sqr(zmc-getzpos) );
      if radius=0 then
        radius:=tpointin(pointsin.objects[koagulant[0]]).getradius;
        maxr:=0;
        maxpoint:=0;
        for i:=0 to length(koagulant)-1 do
          with tpointin(pointsin.objects[koagulant[i]]) do
            begin
              x1:=getxpos+getv_res*cos(getzeta_res)*cos(getalpha_res);
              z1:=getzpos+getv_res*sin(getzeta_res);
              y1:=getypos+getv_res*cos(getzeta_res)*sin(getalpha_res);
              k:=sqrt(sqr(y1)+sqr(z1));
              if k>maxr then
                begin
                  maxr:=k;
                  maxpoint:=i;
                end;
              end;
            with tpointin(pointsin.objects[koagulant[maxpoint]]) do
              begin
                x_nach:=getxpos;
                y_nach:=getypos;
                z_nach:=getzpos;
                alpha1:=getalpha_res;
              if alpha1>pi then
                alpha1:=alpha1;
                zeta1:=getzeta_res;
                x1:=x_nach+getv_res*cos(zeta1)*cos(alpha1);
                y1:=y_nach+getv_res*sin(alpha1)*cos(zeta1);
                if isnan(y1) then
                  y1:=y1;
                z1:=z_nach+getv_res*sin(zeta1);
                maxpointradius:=getradius;
                // count1:=0;
              end;
              while (sqrt(sqr(y1)+sqr(z1)))>(udradius.position-maxpointradius) do
                tpointin(pointsin.objects[koagulant[maxpoint]]) do
                  begin
                    a:=sqrt(y1-y_nach)+sqr(z1-z_nach);
                    b:=2*(y_nach*(y1-y_nach)+z_nach*(z1-z_nach));
                    c:=sqrt(y_nach)+sqr(z_nach)-sqr(udradius.position-maxpointradius);
                  if c=0 then
                    c:=c;
                    dis:=sqrt(b) - 4*a*c;
                    if isnan(dis) then
                      dis:=dis;
                    if (alpha1<pi/2)and(alpha1>-pi/2) then
                      if x1<x_nach then
                        x1:=x_nach+(x_nach-x1);
                      if dis<0 then

```

```

begin
messagedlg('error', mtwarning, [mbok], 0);
tbclearvals.click;
frmmain.close;
halt;
end;

temp:=-b+sqrt(dis);
t1:=temp/(2*a);
temp:=-b-sqrt(dis);
t2:=temp/(2*a);
x_per1:= t1*(x1-x_nach)+x_nach;
x_per2:= t2*(x1-x_nach)+x_nach;
if x_per1=x_nach then
x1:=x1;
if (alpha1>=(-pi/2)) and (alpha1<=(pi/2)) then
if ((x_nach<=x_per1)and(x_per1<=x1)) or
((x_nach<=x_per1)and(x_per1<x1)) or
((x_nach<x_per1)and(x_per1<=x1)) then
begin
x_per:=x_per1;
y_per:=t1*(y1-y_nach)+y_nach;
z_per:=t1*(z1-z_nach)+z_nach;
end
else
begin
x_per:=x_per2;
y_per:=t2*(y1-y_nach)+y_nach;
z_per:=t2*(z1-z_nach)+z_nach;
end;
if (alpha1<(-pi/2)) or (alpha1>(pi/2)) then
if ((x_nach>=x_per1)and(x_per1>=x1)) or
((x_nach>=x_per1)and(x_per1>x1)) or
((x_nach>x_per1)and(x_per1>=x1)) then
begin
x_per:=x_per1;
y_per:=t1*(y1-y_nach)+y_nach;
z_per:=t1*(z1-z_nach)+z_nach;
end
else
begin
x_per:=x_per2;
y_per:=t2*(y1-y_nach)+y_nach;
z_per:=t2*(z1-z_nach)+z_nach;
end;
if (x1<x_nach)and(x_nach<x_per) then
x1:=x1;
if (x_per>udxcasis.position/2-pi*udkrivizna.position/2) and
(x_per<udxcasis.position/2+pi*udkrivizna.position/2) and
(y_per<0) then
if (radius>=3)and(radius<50) then
begin
countcp:=countcp+1;
setlength(crosspoints, countcp);
crosspoints[countcp-1,0]:=x_per;
crosspoints[countcp-1,1]:=y_per;
crosspoints[countcp-1,2]:=z_per;
crosspoints[countcp-1,3]:=radius;
end;
//знайдемо координати кінцевого положення частки
t_s:=(y_per*y1+z_per*z1)/(sqrt(z_per)+sqrt(y_per)) - 1;
y1:=2*(y1-t_s*y_per)-y1;
z1:=2*(z1-t_s*z_per)-z1;
//знайдемо кути нового напрямку вектору швидкості
temp:=sqrt(sqrt(x1-x_per)+sqrt(y1-y_per)+sqrt(z1-z_per));
zeta1:=arcsin((z1-z_per)/temp);
//для усіх чвертей площі

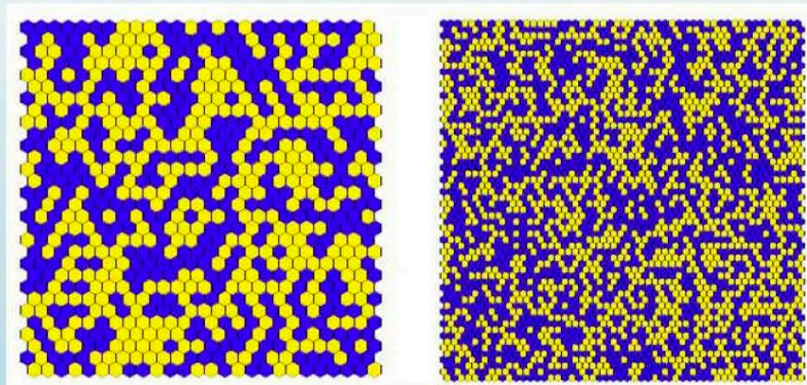
```

```

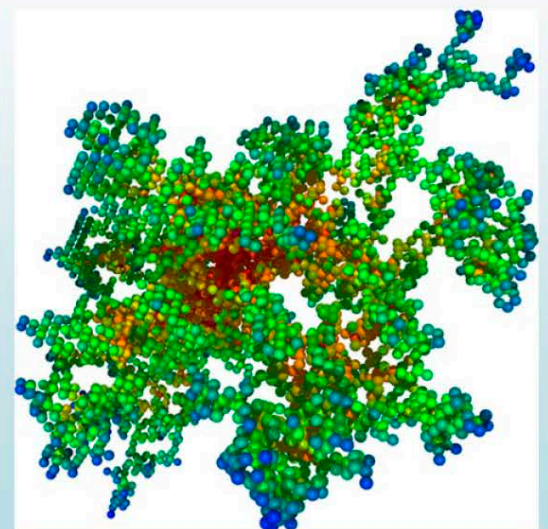
alpha1:=arctan( abs((y1-y_per)/(x1-x_per)));//1 четверть
if ((x1-x_per)<0)and((y1-y_per)>0) then //2 четверть
  alpha1:=pi-alpha1;
if ((x1-x_per)<0)and((y1-y_per)<0) then //3 четверть
  alpha1:=-pi+alpha1;
if ((x1-x_per)>0)and((y1-y_per)<0) then //4 четверть
  alpha1:=-alpha1;
if isnan(zeta1) then
  zeta1:=zeta1;
if isnan(alpha1) then
  alpha1:=alpha1;
deltax:=x_per-x_nach;
deltay:=y_per-y_nach;
deltaz:=z_per-z_nach;
for i:=0 to length(koagulant)-1 do
  begin
  with tpointin(pointsin.objects[koagulant[i]]) do
  begin
  localdx:=deltax;
  localdy:=deltay;
  localdz:=deltaz;
  while getxpos+localdx>udxcasis.position do
  begin
  t:=(udxcasis.position-getxpos)/localdx;
  y_v:=t*localdy+getypos;
  z_v:=t*localdz+getzpos;
  pbmain.canvas.pen.color := clblue;
  pbmain.canvas.moveto(round(xstart+getxpos),round(ycenter+getypos));
  if chlines.checked then
    pbmain.canvas.lineto(round(xstart+udxcasis.position),round(ycenter+y_v));
  localdx:=localdx-(udxcasis.position-getxpos);
  localdy:=localdy-(y_v-getypos);
  localdz:=localdz-(z_v-getzpos);
  setxpos(0);
  setypos(y_v);
  setzpos(z_v);
  incj;
  if x1>udxcasis.position then
  begin
  x1:=x1-udxcasis.position;
  x_per:=x_per-udxcasis.position;
  end;
  runmas[koagulant[i]]:=runmas[koagulant[i]]+1;
  end;
  while getxpos+localdx<0 do
  begin
  t:=abs(getxpos/localdx);
  y_v:=t*localdy+getypos;
  z_v:=t*localdz+getzpos;
  pbmain.canvas.pen.color := clblue;
  main.canvas.moveto(round(xstart+getxpos),round(ycenter+getypos));
  if chlines.checked then
    pbmain.canvas.lineto(round(xstart),round(ycenter+y_v));
  localdx:=localdx+getxpos;
  localdy:=localdy-(y_v-getypos);
  localdz:=localdz-(z_v-getzpos);
  setxpos(udxcasis.position);
  setypos(y_v);
  setzpos(z_v);
  decj;
  if x1<0 then
  begin
  x1:=x1+udxcasis.position;
  x_per:=x_per+udxcasis.position;
  end;
  end;
  end;
  end;

```

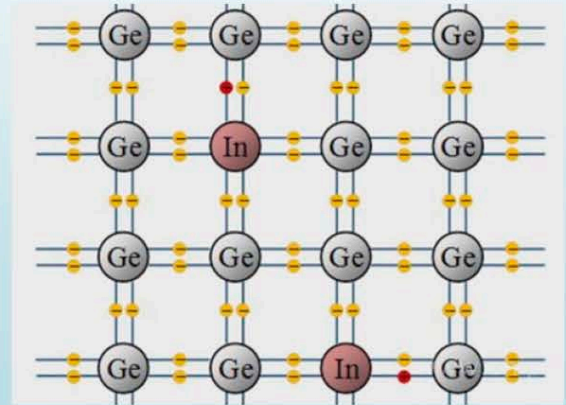
- **Предмет досліджень теорії перколяції (протікання)** відноситься до так званих «критичних явищ», для яких характерно, що при досягненні деяким параметром певного значення, властивості системи різко змінюються. До критичних явищ відносяться, наприклад, так звані фазові переходи другого роду (наприклад, перехід металу з нормального стану в надпровідний при зниженні температури). Всі критичні явища мають спільні риси, найважливіша з яких полягає в тому, що поблизу критичної точки система як би розпадається на блоки із відмінними властивостями, причому розмір окремих блоків необмежено зростає при наближенні до критичної точки. Обриси блоків при цьому випадкові.



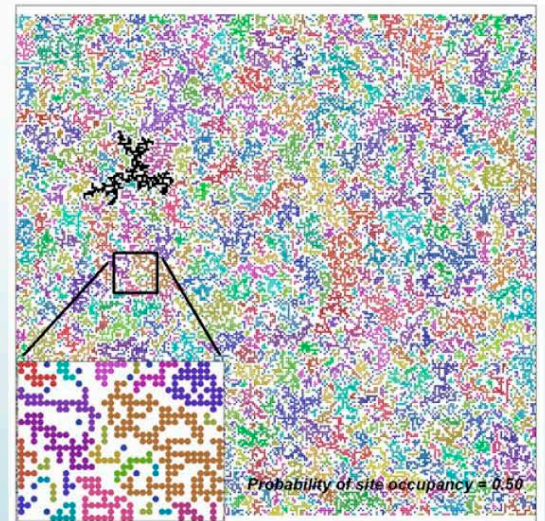
- **Теорія перколяції** займається вивченням властивостей систем, які мають дуже велику кількість елементів, за умови, що наявність зв'язка кожного елемента зі своїми сусідами носить випадковий характер, але задається цілком певним способом (наприклад, за допомогою генератора випадкових чисел, який володіє конкретними властивостями), а також вивчає особливості виникнення та еволюції, а також властивості кластерів в матрицях довільної структури і розмірності.
- **Різні задачі теорії перколяції об'єднуються** тим, що геометрія пов'язаних елементів поблизу порогу протікання у них подібна. Для того щоб це помітити, потрібно відволіктися від дрібномасштабної структури, яка визначається характером зв'язків і властивостями елементів, і стежити тільки за зв'язаністю великих блоків. Універсальна великомасштабна геометрія диктує універсальні властивості фізичних величин, що залежать від структури великих кластерів.



- **Рівень протікання** є надзвичайно важливим для розуміння процесів, що відбуваються в **напівпровідниках з домішками**. Можна, мабуть, сказати, що в даний час домішкові напівпровідники є однією з головних областей застосування теорії перколяції. Теорія перколяції дозволяє уявити, як влаштований напівпровідник при концентрації домішок, близької до N_c . З цим пов'язані специфічні властивості електропровідності та інших важливих характеристик напівпровідника.



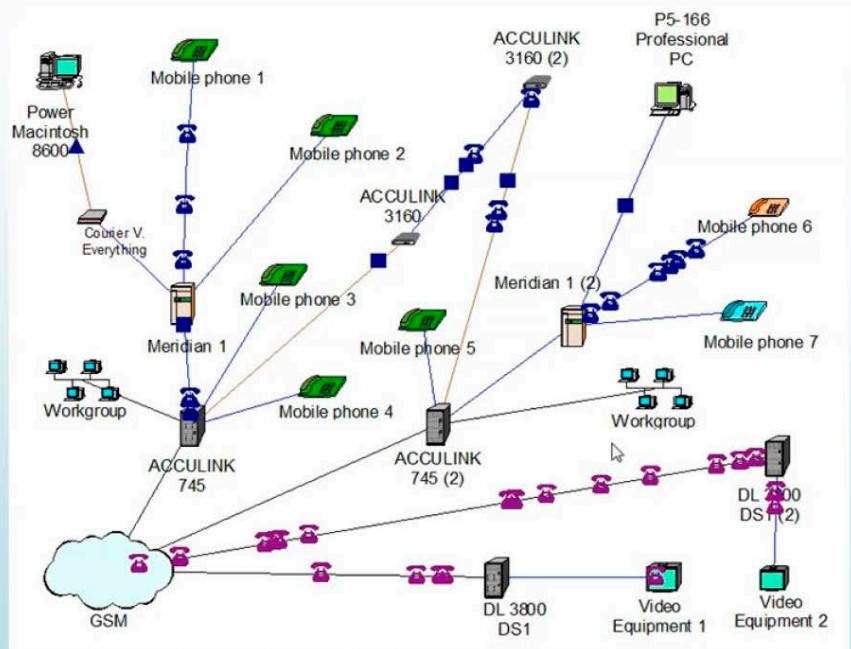
- **Задача обчислення опору систем**, що складаються з дуже великого числа цих елементів, теж вирішується за допомогою теорії перколяції. У низькотемпературних умовах **стрибкова електропровідність** виграє у електропровідності вільних електронів. Методи теорії перколяції з успіхом застосовуються також для знаходження температурної залежності стрибкової електропровідності, залежно від зовнішнього магнітного поля і від ряду інших параметрів. Також ці методи застосовні до будь-якої неоднорідної системи із змінним від точки до точки опором.



- Теорія перколяції досі не отримала широкого розповсюдження в **інформаційних задачах**. Між тим, мережеві структури та інформаційні потоки в них являють собою пряму аналогію з фізичними процесами, досліджуваними в рамках цього напрямку.



- Перколяційна лавина може знайти своє застосування в **високодинамічних мобільних мережах**, рівень змін топології в яких настільки високий, що лавинна доставка даних є єдиним способом маршрутизації. У цьому випадку перколяційна лавина забезпечує той же рівень доставки даних що і звичайна, заощаджуючи при цьому трафік.



Зріст кластерної системи і поява зв'язування

8



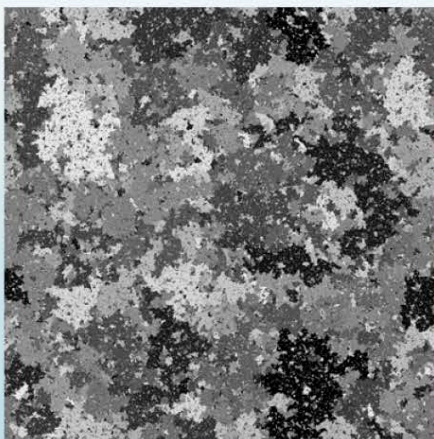
а) мала концентрація



б) велика концентрація;
виникнення зв'язності

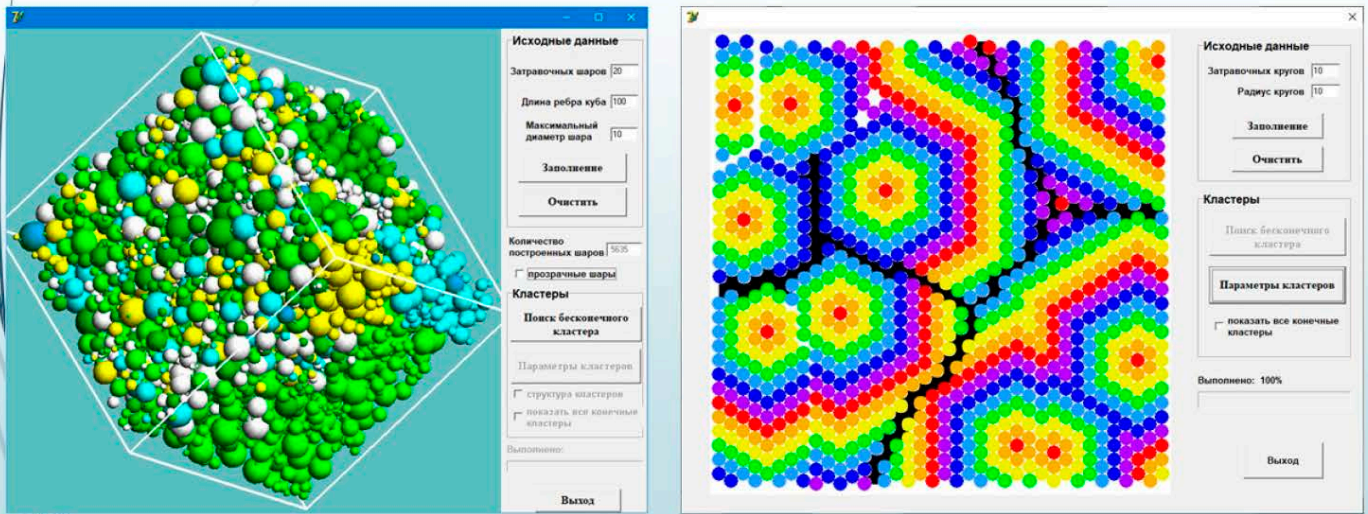
- Критична поведінка потужності нескінченного кластеру:
 $P \propto (x - x_c)^\beta$.
- Критична поведінка кореляційної довжини:
 $\xi \propto |\tau|^{-\nu}$, $\tau = (x - x_c)/x_c$.

- У перколяційній моделі фактично формуються дві кластерні системи – **частинок** і **внутрішніх меж** або порожнин, які є фоном одна для одної, як у мозаїках Мауріца Ешера. Що вважати кластерами залежить від мети дослідження, також як, наприклад, сприйняття зорових образів багато у чому зумовлюється встановленням спостерігача.



Програмне забезпечення моделювання перколяційних кластерів

10



11

Визначення параметрів кластерів

Середня маса кінцевого кластера $\langle s \rangle = \frac{\sum_s s^2 \cdot n_s}{\sum_s s \cdot n_s}$

Середнє значення радіусу гірації $\langle R^2(s) \rangle_L = \frac{\sum_s \langle R^2(s) \rangle \cdot s \cdot n_s}{\sum_s s \cdot n_s}$

Кореляційна довжина (довжина зв'язності) $\xi^2 = \frac{2 \sum_s R^2(s) s^2 n_s}{\sum_s s^2 n_s}$

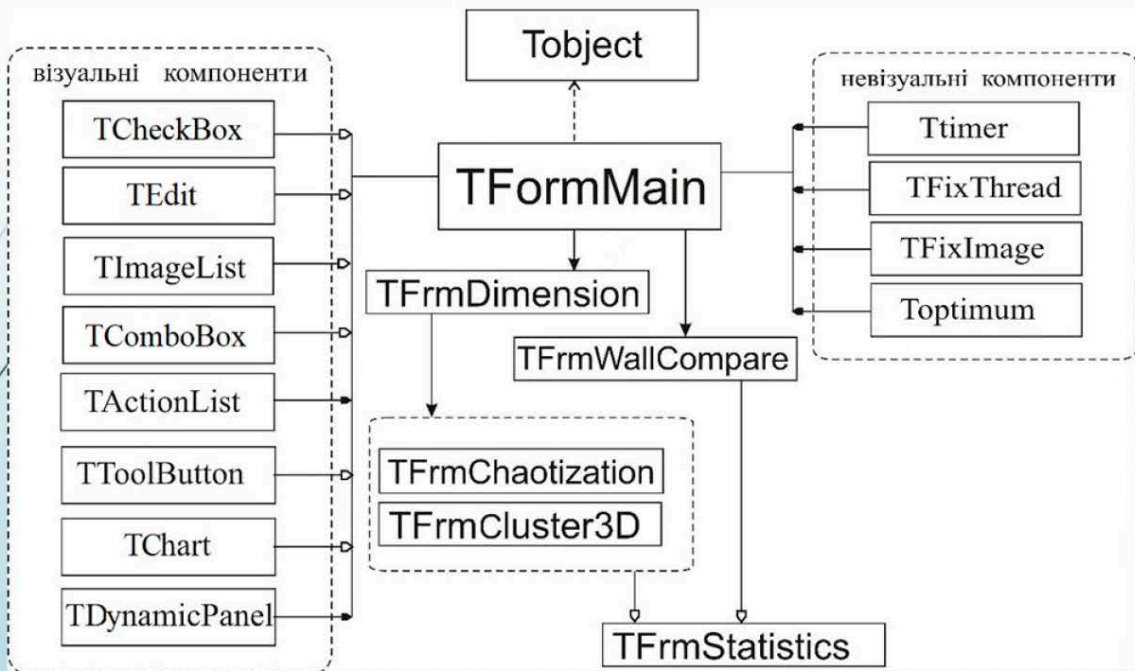
Потужність кластеру $P_\infty(p) = p - \sum_s s \cdot n_s(p) = \frac{N_{\text{беск}}}{N}$

Квадрат радіусу гірації ізотропних кластерів $R^2(s) = \frac{1}{2 \cdot s^2} \sum_{i,j=1}^s [(x_i - x_j)^2 + (y_i - y_j)^2]$

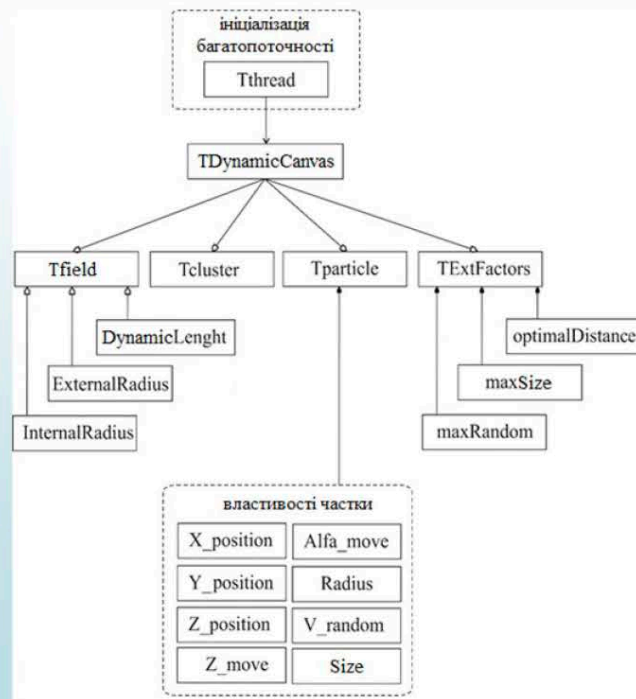
Ймовірність приналежності s-вузловому кластеру $\omega_s = \frac{s \cdot n_s}{\sum_s s \cdot n_s}$

Степень анізотропії кластерів $A = \frac{R_{11}}{R_{22}}$

Діаграма взаємодії функціональних компонент програми

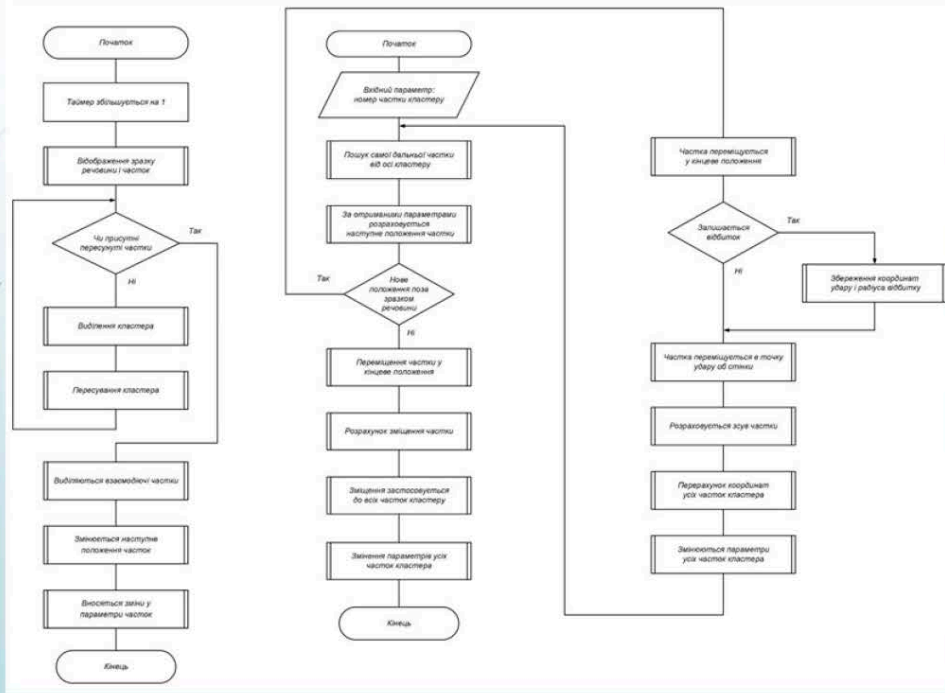


Діаграма взаємодії класів у комп'ютерній моделі



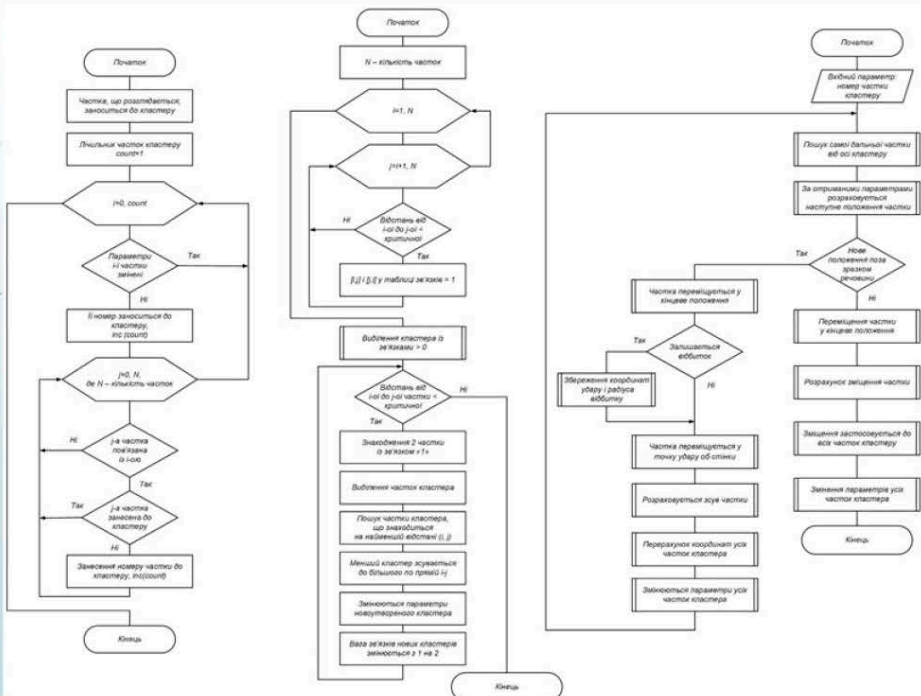
БСА роботи моделі по тактам та коагуляції часток

14



БСА виділення, пересування і взаємодії кластерів

15



Інтерфейс головного вікна на етапі завершення моделювання та модуль аналізу параметрів кластерів

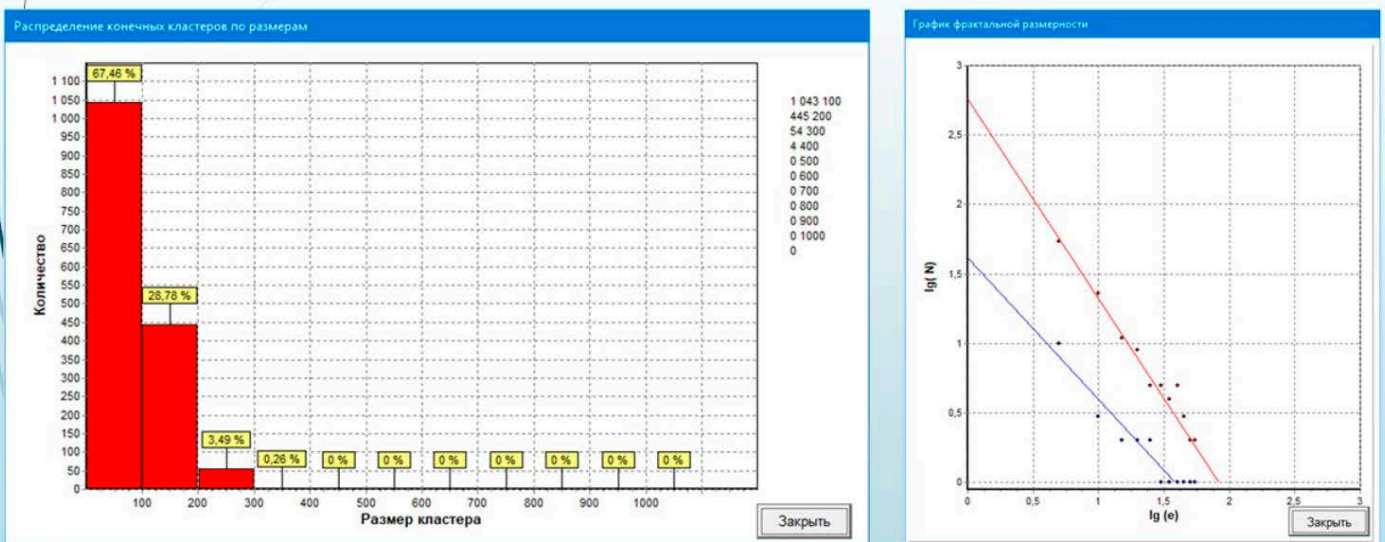
16

The screenshot displays the main application window with a 3D visualization of a cluster of spheres. The interface includes several control panels:

- Исходные данные (Initial Data):**
 - Затравочных шаров: 5
 - Длина ребра куба: 50
 - Максимальный диаметр шара: 10
 - Кнопки: Заполнение, Очистить
- Коллекция построенных шаров (Collection of built spheres):**
 - Количество: 642
 - checkbox: прозрачные шары
 - Кнопка: Показать все конечные кластеры
- Параметры кластеров (Cluster Parameters):**
 - checkbox: структура кластеров
 - checkbox: показать все конечные кластеры
 - Выполнено: 100%
 - Кнопка: Выход
- Параметры бесконечного кластера (Infinite Cluster Parameters):**
 - Масса кластера: 903
 - Мощность кластера: 0,0009
 - Радиус-вектор центра масс кластера: 50,67
 - Квадрат радиуса гирации для изотропных кластеров: 979,80
 - Степень анизотропии по осям xy: $1316,0 : 196,7 = 6,69$
 - Степень анизотропии по осям xz: $1229,0 : 36,3 = 33,90$
 - Степень анизотропии по осям yz: $509,4 : 38,1 = 13,40$
 - Фрактальная размерность: 1,44
 - Индекс, показывающий изменение доли ячеек, принадлежащих кластеру из 5 ячеек при изменении в 3,09
 - Коэффициент пропорциональности, связывающий число ячеек в кластере и радиус гирации: 6,45
- Параметры конечных кластеров (Finite Cluster Parameters):**
 - Масса кластера: 80
 - Количество: 1
 - Мощность кластера: 0,0004
 - Доля ячеек кластеров, принадлежащих кластерам данной массы: 0,0006
 - Радиус-вектор центра масс кластера: 46,64
 - Квадрат радиуса гирации для изотропных кластеров: 67,35
 - Степень анизотропии по осям xy: $16,1 : 4,8 = 3,33$
 - Степень анизотропии по осям xz: $14,6 : 4,4 = 3,34$
 - Степень анизотропии по осям yz: $8,4 : 4,4 = 1,92$
 - Фрактальная размерность: 1,02
 - Индекс, показывающий изменение доли ячеек, принадлежащих кластеру из 5 ячеек при изменении в 3,94
 - Коэффициент пропорциональности, связывающий число ячеек в кластере и радиус гирации: 43,67
- Общие параметры (General Parameters):**
 - Вероятность, что выбранный узел принадлежит какому-либо кластеру: 0,6430
 - Доля ячеек, занятых всеми конечными кластерами: 0,6420
 - Средняя масса конечного кластера: 190,00
 - Среднее значение радиуса гирации: 11,96
 - Корреляционная длина (длина связности): 25,51
 - Кнопка: График фрактальной размерности
 - Кнопка: Гистограмма распределения конечных кластеров
 - Кнопка: Закрыть

Інтерфейс вікна статистики розподілення кластерів за розмірами та отримання графіків фрактальної розмірності кластерів

17



Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015285395

Дата перевірки:
27.05.2023 14:43:59 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
27.05.2023 14:46:00 EEST

ID користувача:
100011688

Назва документа: 4КГ-06 Майстренко Г.Р.

Кількість сторінок: 53 Кількість слів: 10234 Кількість символів: 79441 Розмір файлу: 1.35 MB ID файлу: 1014957919

4.27% Схожість

Найбільша схожість: 0.78% з Інтернет-джерелом (<https://student2.ru/tehnologii/1747144-teoretichna-chastina>)

4.27% Джерела з Інтернету

451

Сторінка 55

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

39

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Майстренко Гліб Рустамович,
здобувач освіти гр. 4КГ-06, та

Кривченко Юрій Вікторович,
керівник дипломного проекту,


не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

«Моделювання перколяційних явищ програмними засобами» (автор роботи – Майстренко Г.Р., керівник роботи – Кривченко Ю.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Майстренко Г.Р. /

Керівник



/ Кривченко Ю.В. /

« 12 » червня 2023 р.

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Майстренка Глеба Рустамовича

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Освітня програма «Комп'ютерна графіка і Web-дизайн»

Керівник дипломного проекту (роботи) Кривченко Юрій Вікторович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Моделювання перколяційних явищ
програмними засобами

Обсяг розрахунково-пояснювальної записки 72 сторінок

Обсяг графічної (презентаційної) частини 17 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню
Представлений на рецензію дипломний проект повністю відповідає меті проектування та технічному завданню. Тематика дипломного проекту є актуальною та присвячена розробки програмного застосунку для моделювання перколяційних явищ, описано процес створення програмного забезпечення для моделювання перколяційних кластерів та оцінки їх параметрів.

б) характеристика виконання кожного розділу дипломного проекту (роботи)
Дипломний проект складається зі вступу, трьох розділів, висновків, переліку використаних джерел. У технологічному розділі виконано огляд і аналіз оцінки створення імітаційно-статистичної моделі, визначення розмірностей кластерів, програмна реалізація моделі взаємодії часток, взаємодія функціональних компонент реалізованої моделі, автоматизація роботи у реалізованій моделі.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи)
Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету Microsoft PowerPoint та Visio. Пояснювальна записка виконана акуратно та у відповідності до норм оформлення документів із використанням офісного пакету Microsoft Word. Загальна якість виконання документації – добра, академічного плагіату у роботі не виявлено

г) перелік позитивних якостей дипломного проекту (роботи) _____

Створена комп'ютерна імітаційно-статистична модель перколяційних явищ, зокрема – кластероутворення, дозволяє візуалізувати кластерну структуру, аналізувати її функціонування, оцінювати параметри кластероутворення і коригувати вхідні характеристики моделі.

д) основні недоліки дипломного проекту (роботи) _____

Серед недоліків роботи варто вказати, відсутність посилань на перелік використаних джерел та вирішити проблему представлення та збереження статистики.

Оцінка розрахункової частини _____ **відмінно**

Оцінка графічної частини _____ **відмінно**

Загальна оцінка _____ **відмінно**

Прізвище, ім'я, по батькові рецензента Стайкуца Сергій Володимирович

Місце роботи і посада рецензента _____

“Державний університет інтелектуальних технологій і зв’язку”,

доцент кафедри кібербезпеки та технічного захисту інформації,

помічник декана факультету інформаційних технологій та кібербезпеки

Підпис: _____

« 16 » 06 2023 р.

ПІДПИС ПОСВІДОЧУ
НАЧАЛЬНИК ВІДДІЛУ
КАДРІВ ДУІТЗ



Сергій Стайкуца

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Майстренко Гліб Рустамович,
здобувач освіти гр. 4КГ-06, та

Кривченко Юрій Вікторович,
керівник дипломного проекту,

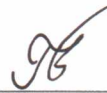
не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

«Моделювання перколяційних явищ програмними засобами» (автор роботи – Майстренко Г.Р., керівник роботи – Кривченко Ю.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Майстренко Г.Р. /

Керівник



/ Кривченко Ю.В. /

« 12 » червня 2023 р.

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015285395

Дата перевірки:
27.05.2023 14:43:59 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
27.05.2023 14:46:00 EEST

ID користувача:
100011688

Назва документа: 4КГ-06 Майстренко Г.Р.

Кількість сторінок: 53 Кількість слів: 10234 Кількість символів: 79441 Розмір файлу: 1.35 MB ID файлу: 1014957919

4.27% Схожість

Найбільша схожість: 0.78% з Інтернет-джерелом (<https://student2.ru/tehnologii/1747144-teoretichna-chastina>)

4.27% Джерела з Інтернету 451 Сторінка 55

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 39