

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

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна інженерія»

Група: 2БКС-28

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

здобувача освіти денної форми навчання
БКС.28.28.000.КРБ

ТОКАРЧУКА
ГЛІБА СЕРГІЙОВИЧА

м. Одеса
2024 р.

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

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітньо-професійна програма: **«Комп'ютерна інженерія»**

Група: **2БКС-28**

ПОЯСНЮВАЛЬНА ЗАПИСКА

До кваліфікаційної роботи бакалавра на тему: **«Дослідження режимів роботи динамічної системи засобами комп'ютерного моделювання»**

Проектний матеріал складається з пояснювальної записки на 79 сторінках та графічного (презентаційного) матеріалу на 20 аркушах (слайдах)

Виконавець _____ (Токарчук Г.С.)

Керівник проекту _____ (Кривченко Ю.В.)

Консультанти:

з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)

з нормоконтролю _____ (Петрашова В.І.)

старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Завідувач кафедри _____ (Іванова Л.В.)

Завідувач відділення _____ (Скорнякова О.В.)

Захист «24» 06 2024 р.

Протокол ЕК № 1

Оцінка ЕК 5/відмінно/ 95

Секретар ЕК _____

АНОТАЦІЯ

У випускній роботі досліджувались режими роботи динамічної системи засобами комп'ютерного моделювання. Математична модель та реалізований програмний застосунок дозволяє досліджувати еволюцію абстрактних систем, що складаються з компонент, які взаємодіють за довільними законами. Визначено три типи динамічних систем: хаотична, квазіперіодична і дивна. Реалізовано зниження розмірності фазового чотиривимірного простору до тривимірного і візуалізацію динамічних систем за допомогою методу тривимірних перетинів Пуанкаре.

Створено алгоритми і програмний застосунок мовою програмування C++ на базі технології ООП для реалізації комп'ютерних експериментів з динамічними системами різних типів, зокрема візуалізації множин завдяки інтерфейсу OpenGL, і розрахунку їх розмірностей. У програмному застосунку реалізовано побудову множини точок, що генеруються системою рекурентних рівнянь. У багатовимірному фазовому просторі візуалізація цієї безлічі за допомогою перетину Пуанкаре дозволяє виявити особливості еволюції динамічної системи.

Вивчено характер поведінки динамічних систем на прикладі запропонованих рекурентних рівнянь, структура та властивості її множин.

Проаналізовані можливі типи розвитку динамічної системи в запропонованій математичній моделі, показана можливість керування сценаріями еволюції системи, що може використовуватися в різних областях науки і техніки, пов'язаних з теорією хаосу.

Описано заходи з охорони праці та техніки безпеки.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення Комп'ютерних систем Кафедра Комп'ютерної інженерії
Спеціальність 123 «Комп'ютерна інженерія»
Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР

Беркань І.В.

“ 15 ” 01 20 24 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

здобувачеві освіти Токарчуку Глібу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Дослідження режимів роботи динамічної системи засобами комп'ютерного моделювання

затверджена наказом по коледжу від “01” листопада 20 23 р. № 244-АГ-О.В.

2. Термін здачі студентом кваліфікаційної роботи 13.06.24 р.

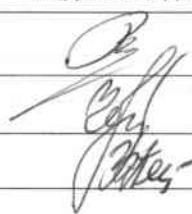
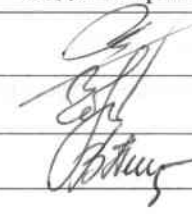
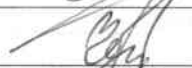
3. Вихідні дані до роботи 1. Система рекурентних рівнянь для отримання множини точок динамічної системи; 2. Параметри відомих динамічних систем; 3. Ознаки класифікації динамічних систем та їх основні характеристики; 4. Візуалізувати динамічну систему у 3-вимірному просторі; 5. Моделювання та візуалізацію виконувати засобами ООП та графічного інтерфейсу OpenGL

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

Математична модель динамічної системи; Вибір технологій та інструментів для реалізації моделі; Розробка схеми обчислювального процесу; Розробка структур даних та їх взаємодії у програмі; Розробка блок-схем алгоритмів програми; Розробка інтерфейсу програмного застосунку; Результати моделювання динамічної системи; Питання охорони праці та техніки безпеки

5. Перелік графічного матеріалу (слайдів мультимедійної презентації) Послідовність етапів побудови математичної моделі; Система ітераційних рівнянь для опису динамічної системи; Інтерфейс програми для моделювання динамічної системи; Схема обчислювального процесу в програмі; Діаграма взаємодії об'єктів АТД у програмі; БСА функцій у класах; Інтерфейс програми у покроковому режимі і етапи побудови динамічної системи; Інтерфейс програми у процесі формування перетину з 4-вимірної множини; Візуалізація динамічної системи у хаотичному, квазіперіодичному та дивному режимах

6. Консультанти по кваліфікаційній роботі, із зазначенням розділів, що їх стосуються

Розділ	Консультант	ПІДПИС	
		Завдання видав	Завдання прийняв
Основний розділ	Кривченко Ю.В.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 15.06.24

Керівник роботи Кривченко Ю.В.

Завдання прийняв до виконання

(підпис)

(підпис)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Вступ. Аналіз технічного завдання	8.05.24	Виконав
2.	Вивчення системи рекурентних рівнянь	10.05.24	Виконав
3.	Аналіз відомих динамічних систем та їх параметрів	11.05.24	Виконав
4.	Загальний опис реалізації моделі	13.05.24	Виконав
5.	Вибір і опис інструментарію для реалізації моделі	15.05.24	Виконав
6.	Розробка інтерфейсу та опис вимог до ПЗ	17.05.24	Виконав
7.	Розробка схеми обчислювального процесу	20.05.24	Виконав
8.	Розробка алгоритмів візуалізації	23.05.24	Виконав
9.	Розробка алгоритмів реалізації перетину Пуанкаре	26.05.24	Виконав
10.	Розробка алгоритмів реалізації розрахунків	29.05.24	Виконав
11.	Розробка діаграми взаємодії АТД у програмі	1.06.24	Виконав
12.	Реалізація програми мовою C++	3.06.24	Виконав
13.	Аналіз результатів, зняття характеристик	6.06.24	Виконав
14.	Розробка питань з охорони праці	8.06.24	Виконав
15.	Підготовка роботи до захисту та тестування ПЗ	11.06.24	Виконав
16.	Оформлення слайдів мультимедійної презентації	13.06.24	Виконав

Здобувач освіти

(підпис)

Керівник роботи

(підпис)

[illegible]

ЗМІСТ

Вступ.....	6
1 Основний розділ	7
1.1 Створення математичної моделі динамічної системи	7
1.2 Вибір технологій та інструментів для реалізації моделі.....	10
1.2.1 Застосований інструментарій фреймворку Qt.....	11
1.2.2 Використання STL та OpenGL.....	13
1.3 Розробка схеми обчислювального процесу	14
1.4 Розробка структур даних та їх взаємодії у програмі.....	16
1.5 Розробка блок-схем алгоритмів програми.....	24
1.6 Розробка інтерфейсу програмного застосунку	43
1.7 Результати моделювання динамічної системи.....	50
1.7.1 Хаотичний режим динамічної системи.....	51
1.7.2 Квazіперіодичний режим динамічної системи.....	53
1.7.3 Дивний режим у динамічній системі.....	55
2 Розділ охорони праці та техніки безпеки.....	58
2.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника.....	58
2.2 Розробка заходів з охорони праці.....	59
2.2.1 Виробничі приміщення.....	59
2.2.2 Мікроклімат робочої зони працівників, вентиляція.....	59
2.2.3 Освітлення робочого місця, шум, вібрація.....	60
2.2.4 Електробезпека.....	61
2.2.5 Організація робочого місця користувача ПК.....	61
2.3 Пожежна безпека	62
Висновки	63
Перелік використаних інформаційних джерел	64
Додаток А. Фрагмент коду функції Dimension класу Fractal мовою C++	65
Додаток Б. Слайди мультимедійної презентації.....	70

ВСТУП

Моделювання поза поміттю комп'ютеру являється одним із найпотужніших засобів вивчення динамічних систем. Воно дає спроможність здійснювати обчислювальні експерименти із системами на стадії проектування, але разом з тим вивчати системи, натурні експерименти із якими через небезпечність чи високу вартість недоцільні. В той же час, завдяки близькості поза формою щодо фізичного моделювання, цей метод вивчення доступний широкому загалу користувачів.

В роботі увагу приділено процесам, це моделюються поза поміттю апарату динамічних систем. Під найпростішою динамічною системою зазвичай розуміють систему, поведінка якої задається сукупністю звичайних диференціальних виразів в формі Коші із досить гладкими правими частинами, це забезпечують існування і єдність розв'язку. Знаходження розв'язку систем виразів в формі Коші – традиційна задача обчислювальної математики, записана в формі диференціальних виразів.

Створені у останні роки програмні засоби створення числових методів не тільки забезпечують задані вимоги щодо похибки розв'язку, але й вимагають визначення обчислювальної складності задачі. Саме математичному моделюванню належить істотна частина успіхів, досягнутих в передових галузях сучасної науки. Воно все швидше опановує нові напрями – авіабудування, машинобудування, хімічне виробництво. Уже нині дуже перспективними являється спроби його застосування щодо винятково важливих, але недостатньо формалізованих сфер – економіки, соціології тощо. Важлива перевага методів моделювання динамічних систем полягає у тому, це вони дозволяють різко скоротити обсяг та масштаби натурних експериментів. Математичне моделювання незамінне там, тут натурний експеримент може стати небезпечним та навіть катастрофічним – в ядерній техніці, екології, при розробці економічних реформ.

В кваліфікаційній роботі виконується моделювання динамічної системи, заданої системою рекурентних виразів, відображення засобами комп'ютерного моделювання, досліджуються режими її роботи.

					БКС 28. 28 000. 00 КРБ ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

1 ОСНОВНИЙ РОЗДІЛ

1.1 Створення математичної моделі динамічної системи

Заміна одних об'єктів (оригіналів) іншими (моделями) та вивчення властивостей об'єктів поза моделями називається моделюванням.

Математичне моделювання можна розглядати як засіб вивчення реальної системи шляхом її заміни зручнішою задля експериментального вивчення системою (моделлю), це зберігає істотні риси оригіналу. При моделюванні здійснюється апроксимація процедури опису більш простою та зручною задля практичного аналізу функцією – моделлю. Математичні моделі, особливо ті, це використовують чисельні методи, потребують задля свого створення значних інтелектуальних, фінансових та часових затрат. Тому рішення про створення нової моделі приймається лише у разі відсутності більш простих шляхів вирішення поставленої проблеми (наприклад, модифікації однієї із існуючих моделей).

Вивчення об'єкта моделювання і складання його математичного опису полягає в встановленні зв'язків між характеристиками процесу, виявленні його граничних та початкових умов і формалізації процесу в виді системи математичних співвідношень. Процес побудови будь-якої математичної моделі можна подати послідовністю етапів, зображених на рис. 1.1.

На етапі вивчення об'єкта моделювання потрібно виконати такі дії:

- аналіз взаємодії об'єкта із зовнішнім середовищем, виділення характеристик вхідних впливів і реакції об'єкта, класифікація їх на вимірні і невимірні;
- проведення декомпозиції і вивчення внутрішньої структури об'єкта;
- вивчення порядку функціонування об'єкту, виявлення зв'язку між входом і виходом, утворення множини станів об'єкта;
- збір і перевірка існуючих експериментальних даних про об'єкти-аналоги, проведення, поза необхідності, додаткових експериментів;
- класифікація об'єкта моделювання на стаціонарний чи нестаціонарний, визначення міри впливу випадкових факторів на об'єкт і порядку нелінійності

зв'язків між характеристиками об'єкта;

– аналітичний огляд літературних джерел, аналіз і порівняння побудованих раніше моделей подібних об'єктів;

– аналіз і узагальнення всього накопиченого матеріалу, розроблення загального плану створення математичної моделі.

У деяких випадках вивчення внутрішньої будови і порядку функціонування об'єкта моделювання неможливе чи економічно недоцільне. Тому можна розглядати об'єкт як «чорний ящик», стосовно якого нам відомі лише його входи і виходи.

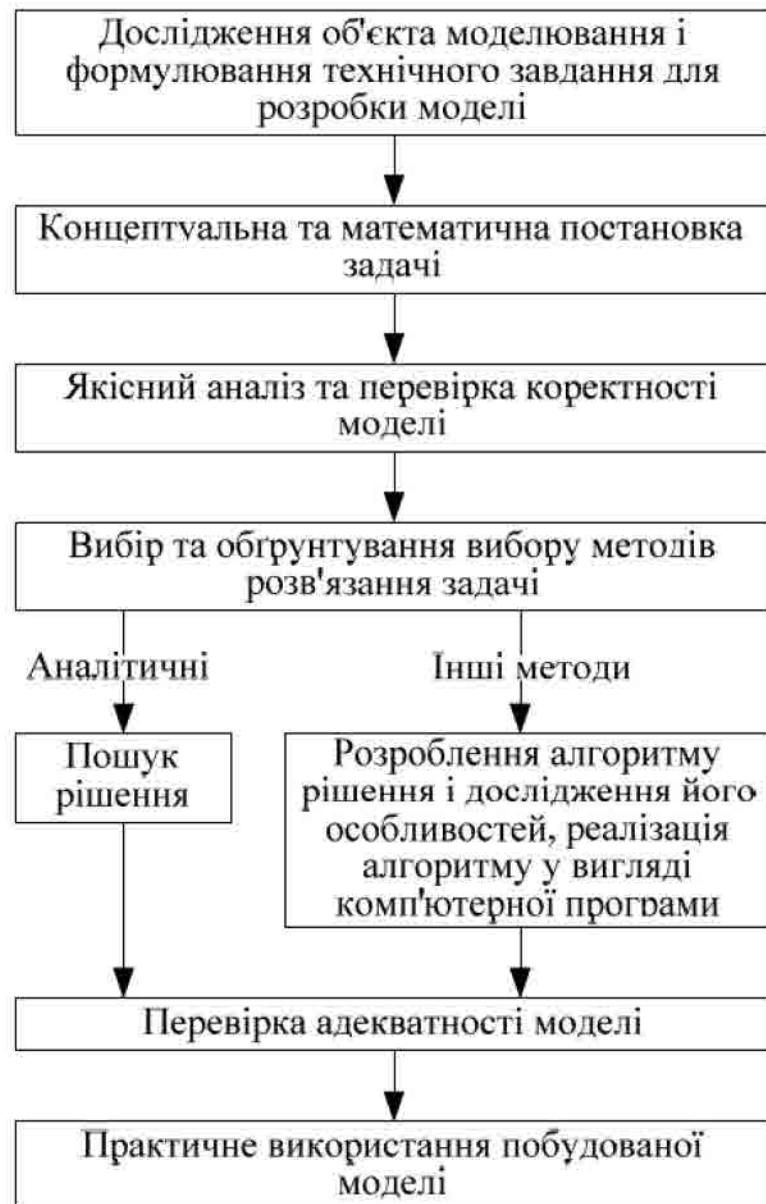


Рисунок 1.1. Послідовність етапів побудови математичної моделі

Розглянемо три схожих динамічних системи, описувані трьома системами диференціальних виразів, кожна із трьома ступенями свободи та дискретним часом (1.1-1.3).

$$\Phi 1(x, y, z) = \begin{cases} \dot{x}_n = x_n - k_{xy} p x_n^2 + k_{yx} q y_n^2 + x_{\text{ex}} \\ \dot{y}_n = y_n + k_{xy} p x_n^2 - (k_{yx} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ \dot{z}_n = z_n + k_{yz} q y_n^2 - (k_{zy} + k_{\text{aas}}) r z_n^2 \end{cases} \quad (1.1)$$

$$\Phi 2(x, y, z) = \begin{cases} \dot{x}_n = x_n - k_{xy} p x_n^2 + k_{yx} q y_n^2 + x_{\text{ex}} \\ \dot{y}_n = y_n + k_{xy} p x_{n+1}^2 - (k_{yx} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ \dot{z}_n = z_n + k_{yz} q y_{n+1}^2 - (k_{zy} + k_{\text{aas}}) r z_n^2 \end{cases} \quad (1.2)$$

$$\Phi 3(x, y, z) = \begin{cases} \dot{x}_n = x_n - (k_{xy} + k_{\text{aas}_{-1}}) p x_n^2 + k_{yx} q y_n^2 + x_{\text{ex}} \\ \dot{y}_n = y_n + k_{xy} p x_n^2 - (k_{yx} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ \dot{z}_n = z_n + k_{yz} q y_n^2 - (k_{zy} + k_{\text{aas}}) r z_n^2 \end{cases} \quad (1.3)$$

Тут $\{x, y, z\} \in \mathbb{R}$ – динамічні змінні, описують тому стан кожного із трьох "рівнів" динамічної системи; $x_{\text{ex}} = \text{const} \in \mathbb{R}^+$ – потік, це входить у систему; $\{k_{xy}, k_{yx}, \dots, k_{\text{aas}}\} \in (0, 1)$ та $\{p, r, s\} \in (0, 1)$ – перехідні коефіцієнти, це характеризують статичну та динамічну взаємодію рівнів тому.

Стан систем змінюється стрибками, адже час дискретний. Задля першої системи $\Phi 1(x, y, z)$ зміна режиму здійснюється поза один такт одночасно задля усіх трьох рівнів, тому щодо виразів (1.1). Задля другої динамічної системи $\Phi 2(x, y, z)$, рівняння еволюції якої схожі із рівняннями задля першої системи, зміни режиму протягом одного такту відбуваються послідовно задля її трьох рівнів, тому це зміна режиму другого рівня (y) залежить від режиму першого (x) у наступний момент часу, але зміна режиму третього рівня (z) залежить від режиму другого у наступний момент часу. Таким же чином здійснюється зміна режиму та у третій системі. В кожній із досліджуваних систем крім постійного вхідного потоку (x_{ex}) являється потік, це залишає систему. Задля систем $\Phi 1(x, y, z)$ та $\Phi 2(x, y, z)$ він квадратично залежить від режиму третього рівня. В третій динамічній системі $\Phi 3(x, y, z)$

являється ще один вихідний потік із першого рівня. Наявність таких відмінностей у системах призводить щодо дещо різних властивостей.

Комп'ютерна модель динамічної системи в багатовимірному фазовому світі дозволить ефективно досліджувати розвиток і параметри динамічної системи.

В даній роботі задля моделювання динамічної системи в тривимірному світі необхідно побудувати безліч часток, це генеруються системою рекурентних виразів. Необхідно забезпечити візуалізацію множини, спроможність її обертання навколо осей напрямків та перетин площиною. В програмній моделі мають розраховуватися fractal та інформаційна розмірності. Аналогічну задачу передбачається вирішувати та в 4-вимірному світі. Особливість завдання – побудова 3-вимірного січення Poincare (прямокутного паралелепіпеда) чотиривимірної множини часток, але разом з тим відображення частини множини, це потрапила у перетин. Відображення динамічної системи і обчислення її властивостей дозволить визначати межі існування режимів: частка, граничний цикл, хаотичний стан, періодичний стан.

1.2 Вибір технологій і інструментів задля створення моделі

Програмний застосунок задля моделювання і візуалізації динамічної системи створено із використанням принципів об'єктно-орієнтованого програмування (ООП). Такий підхід дозволив описати всі структури, реалізовані у моделі, поза поміччю об'єктів, процедур та процедур. Програмний продукт реалізований на мові C++ (стандарту C++ 11) поза поміччю фреймворку Qt (версія 5.11.2 задля x64). В якості компілятора задля середовища Qt використано компілятор версії 14.0 Microsoft Visual Studio C++. Задля створення послідовних та асоціативних контейнерів, але разом з тим алгоритмів роботи із ними та ітераторів застосовується вбудована у C++ бібліотека об'єктів-контейнерів – стандартна бібліотека шаблонів STL. Відображення графічних об'єктів буде виконана поза поміччю стандартного графічного інтерфейсу OpenGL на основі модулю QtOpenGL, розрахованого на застосування системної бібліотеки OpenGL.

					БКС 28. 28 000. 00 КРБ ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.1 Застосування інструментарію фреймворку Qt

Qt являється кросплатформним фреймворком задля розробки програмного забезпечення на мові програмування C++. Він надає запускати написане із його поміччю програмне забезпечення у більшості сучасних операційних систем шляхом простої компіляції додатку задля кожної системи без зміни вихідного коду. Включає у себе всі основні класи, котрі можуть знадобитися при розробці прикладного програмного забезпечення, являється повністю об'єктно-орієнтованим, розширюваним та підтримує техніку компонентного програмування. Відмітна особливість – використання метаоб'єктного компілятора – попередньої системи обробки вихідного коду. Розширення можливостей забезпечується системою плагінів, котрі можливо розміщувати безпосередньо у панелі візуального редактора. Разом з тим існує спроможність розширення звичної функціональності віджетів, пов'язаної із розміщенням їх на екрані, відображенням, перемальовуванням при зміні розмірів віконця.

Механізм сигналів та слотів грає вирішальну роль у розробці програм Qt. Він надає прикладному програмісту пов'язувати різні об'єкти, це нічого не знають один про одного. Слоти майже збігаються зі звичайними функціями, котрі оголошуються всередині об'єктів C++ (процедури-члени). Однак слоти (на відміну від звичайних процедур-членів) можуть підключатися щодо сигналів та у результаті вони будуть викликатися при кожному генеруванні доречного сигналу. Оператор connect () виглядає наступним чином:

connect (відправник, SIGNAL(сигнал), одержувач, SLOT(слот));

тут відправник та одержувач являється покажчиками на об'єкти QObject; сигнал та слот являється сигнатурами процедур без імен властивостей. Макроси SIGNAL та SLOT фактично перетворюють свої аргументи у рядкові змінні. Важливою перевагою засобів розробки Qt являється розширення мови C++ поза поміччю механізму створення компонентів програмного забезпечення, котрі можуть бути пов'язані один із одним, незважаючи на те, це вони можуть нічого не знати один про одного. Цей механізм називається метаоб'єктною системою та забезпечує дві основні службові процедури: взаємодію сигналів та слотів і аналіз

внутрішнього режиму додатку (introspection). Аналіз внутрішнього режиму необхідний задля створення сигналів та слотів. Метаоб'єктна система разом з тим підтримує властивості, котрі використовуються модулем Qt Designer.

В стандартній мові C++ не передбачена динамічна підтримка метаданих, необхідних системі метаоб'єктів Qt. У Qt ця проблема вирішується поза рахунок застосування спеціального інструменту – компілятора moc. Останній переглядає визначення об'єктів із макросом Q_OBJECT та робить відповідну інформацію доступною функціям C++. Оскільки всі функціональні можливості moc забезпечуються тільки поза поміччю «чистого» C++, метаоб'єктна система Qt буде працювати із будь-яким компілятором C++. Цей механізм працює наступним чином:

1) макрос Q_OBJECT оголошує деякі процедури, котрі необхідні задля аналізу внутрішнього режиму та котрі повинні бути реалізовані у кожному підкласі QObject : metaObject(), tr(), qt_metacall() та деякі інші;

2) компілятор moc генерує створення процедур, оголошених макросом Q_OBJECT, та усіх сигналів;

3) такі процедури-члени об'єкту QObject як connect() та disconnect() під час своєї роботи використовують процедури аналізу внутрішнього режиму.

Всі зазначені вище операції виконуються автоматично при роботі qmake, moc та при компіляції QObject.

Діалогові віконця (dialogs) надають користувачеві спроможність задавати необхідні число властивостей та вибирати певні режими роботи та являють собою інструмент, поза поміччю якого користувачі та застосунки можуть «спілкуватися» один із одним.

Поза термінологією Qt віджетами (widget) називають будь-котрий візуальний елемент графічного інтерфейсу користувача GUI. Кнопки, меню, смуги прокрутки та фрейми являється прикладами віджетів. Одні віджети можуть містити у собі інші віджети. Наприклад, віконце додатку зазвичай являється віджетом, це містить QMenuBar (панель меню), кілька QToolBar (панель інструментів), QStatusBar (рядок режиму) та деякі інші віджети. Більшість

застосунків використовують QMainWindow чи QDialog в якості вікна додатку, проте у Qt будь-котрий віджет може бути вікном.

Задля розміщення віджетів в вікні використовуються менеджери компіляція (layout managers). Менеджери компіляція можуть містити як віджети, так та інші менеджери компіляція, зокрема менеджер и горизонтального QHBoxLayout та вертикального QVBoxLayout компіляція. Зовнішній менеджер компіляція (типа QHBoxLayout) являється головним. Він встановлюється поза поміччю команди setLayout (mainLayout) та відповідальний поза всю область, займану діалоговим вікном. Інші три менеджера компоновки (leftLayout, topLeftLayout та rightLayout) являється внутрішніми. Особливістю об'єктів менеджерів компіляція являється те, це вони не являється віджетами. Вони походять від об'єкту QLayout, котрий у свою чергу походить від об'єкту QObject. При роботі додатку менеджери компіляція невидимі. При додаванні внутрішніх менеджерів компіляція щодо батьківського менеджера компіляція (команди leftLayout -> addLayout (topLeftLayout), mainLayout -> addLayout (leftLayout) та mainLayout -> addLayout (rightLayout)) задля них автоматично встановлюється батьківський зв'язок. Далі, коли головний менеджер компіляція встановлюється задля діалогу, він стає дочірнім елементом діалогу та всі віджети у менеджерах компіляція стають дочірніми елементами діалогу.

1.2.2 Використання STL і OpenGL

В стандарт C++ 11 входить власна вбудована бібліотека об'єктів-контейнерів – стандартна бібліотека шаблонів STL (standard template library). Із STL пов'язано кілька основних понять: контейнери, алгоритми та ітератори. Контейнер являється способом організації зберігання даних. В STL включені найбільш корисні із них. Контейнери STL підключаються щодо додатку поза поміччю шаблонних об'єктів, отже, можна легко змінити тип збережених у них даних. Під алгоритмами у STL мають на увазі процедури, застосовувані щодо контейнерів задля обробки їх даних різними способами: алгоритми сортування, копіювання, пошуку та об'єднання. Алгоритми представлені у STL в виді шаблонних процедур, однак вони не являється методами об'єктів-контейнерів – це абсолютно незалежні процедури.

					<i>БКС 28. 28 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

Ітератори являється узагальненням концепції вказівників. Вони посилаються на елементи контейнера, їх можна інкрементувати, як звичайні вказівники та вони будуть посилатися послідовно на всі елементи контейнера. Контейнери STL поділяються на дві категорії: послідовні та асоціативні. Щодо послідовних відносяться вектори (типа vector), списки list та черги із двостороннім доступом deque. Серед асоціативних виділяють множини set, мультимножини multiset, відтворення map та мультिवідображення multimap .

OpenGL являється стандартним програмним інтерфейсом, призначеним задля відтворення тривимірної графіки. Застосунки Qt можуть відображати графіку 3D, використовуючи модуль QtOpenGL, котрий розрахований на застосування системної бібліотеки OpenGL. Цей модуль являє собою клас QGLWidget, задля якого можна створювати підкласи задля розробки власних віджетів, це малюються із використанням команд OpenGL.

1.3 Розробка схеми обчислювального процесу

Розглянута вище модель динамічної системи в багатовимірному фазовому світі описана системою трьох чи чотирьох ітераційних виразів задля 3-вимірного фазового простору (1.4) і задля 4-вимірного (1.5).

$$\Phi(x, y, z) = \begin{cases} x_{n+1} = x_n - k_{xy} p x_n^2 + k_{yx} q y_n^2 + x_{in} \\ y_{n+1} = y_n + k_{xy} p x_n^2 - (k_{yx} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ z_{n+1} = z_n + k_{yz} q y_n^2 - (k_{zy} + k_{out}) r z_n^2 \end{cases} \quad (1.4)$$

$$\Phi(x, y, z, w) = \begin{cases} x_{n+1} = x_n - k_{xy} p x_n^2 + k_{yx} q y_n^2 + x_{in} \\ y_{n+1} = y_n + k_{xy} p x_n^2 - (k_{yx} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ z_{n+1} = z_n + k_{yz} q y_n^2 - (k_{zy} + k_{zw}) s w_n^2 + k_{zw} s w_n^2 \\ w_{n+1} = w_n + k_{zw} r z_n^2 - (k_{wz} + k_{out}) s w_n^2 \end{cases} \quad (1.5)$$

тут x, y, z, w – динамічні змінні, це описують, тому, стан кожного із рівнів динамічної системи; k_{ij} – перехідні коефіцієнти, це характеризують статичну та динамічну взаємодію рівнів тому; p, q, r, s – розподільні коефіцієнти, x_{in} – вхідне число задля системи, причому, $\{k_{ij}\}$ та $\{p, q, r\}$ являється $(0,1)$, $\{x, y, z\}$ являється

$R, x_{in} = \text{const}$ являється R^+ .

Безліч часток в фазовому світі динамічної системи, щодо якої прагнуть траєкторії системи, це еволюціонує, має певний вигляд. Цей вигляд траєкторії визначається особливостями розвитку поточної динамічної системи. Аналітичні методи вирішення подібних виразів (1.4, 1.5) важкі, громіздкі та обмежені точністю проведених обчислень, адже округлення результатів на кожній ітерації тягне великі та часом непередбачувані зміни у наступних. Більшість результатів в цій області отримують шляхом комп'ютерного моделювання, це відноситься та щодо обчислення фрактальної розмірності. Вже згадана комп'ютерна модель реалізує систему трьох чи чотирьох ітераційних виразів, це описують еволюцію складеного потоку в фазовому світі. Задля вивчення еволюції динамічних систем у багатомірних фазових просторах, заданих системою рекурентних виразів, в рамках поточної роботи розроблено програмний застосунок моделювання динамічних систем, це надає вирішувати такі завдання:

- моделювати системи рекурентних виразів, це складаються із трьох та чотирьох виразів (1.4, 1.5) із їх ітераційним рішенням;
- візуалізувати динаміку частки-рішення за кожного звернення щодо рівняння у 3- та 4-вимірному світі тому;
- формувати та візуалізувати динамічну систему в виді множини часток, забезпечуючи спроможність їх обертання, масштабування, зміщення по осях напрямків (задля 3-вимірного випадку чи задля січення Poincare у 4-вимірному випадку);
- визначати межі існування режимів: частка, граничний цикл, періодичний стан, дивний стан;
- розраховувати параметри динамічної системи, такі як фрактальну, інформаційну, кореляційну розмірності;
- аналізувати динамічні системи із побудовою та візуалізацією перетинів Poincare по заданих осях напрямків;
- розраховувати параметри траєкторій в перетині, такі як фрактальну, кореляційну, інформаційну розмірності.

В програмному додатку реалізовано візуалізацію множини часток, це генеруються системою рекурентних виразів. В багатовимірному фазовому світі відображення цієї множини поза поміччю січення Poincare надає виявити особливості еволюції динамічної системи.

Реалізація ітераційної процедури (1.4, 1.5) передбачає завдання початкових чисел динамічних змінних x, v, z, w та їх фіксування за кожної ітерації. Зміна чисел цих змінних передбачається лише при отриманні розрахованих на їх основі нових чисел. При створення в програмі заданих систем виразів (1.4, 1.5) використано принципово інший алгоритм обчислення змінних, котрий реалізує послідовний облік змін чисел вже розрахованих змінних при обчислення інших змінних на поточній ітерації. Ця якісна зміна алгоритму обчислення призводить, тому, щодо зміни шляхів еволюції поточної системи. При створення моделі передбачена відображення траєкторій генезису досліджуваної системи у багатовимірному фазовому світі станів рівнів та спроможність оцінки її характеристик. Узагальнена схема обчислювального процесу, реалізованого у програмному додатку, показана на рис.1.2.

1.4 Розробка структур даних і їх взаємодії в програмі

Програмний застосунок містить набір об'єктів та інкапсульованих у них процедур задля створення заданої функціональності. Асоціативні контейнерні класи типу відтворення (map) представляють собою динамічні структури даних, у яких кожен елемент має один ключовий об'єкт чи поле та один цільовий (асоційований) об'єкт чи поле (поле чисел чи даних).

Ключове поле (ключ) однозначно дорівнює полю даних. В контейнерах типу множина та відтворення не може бути двох елементів, це володіють одним та тим же ключем. Тип об'єкта чи поля ключа може бути будь-яким (int, double, char, string, QString), як та тип об'єкту чи поля даних (int, double, char, float та ін.). В програмній створення зазначені типи об'єктів представляють собою окремі поля певного типу (у основному, типа string, int та double).

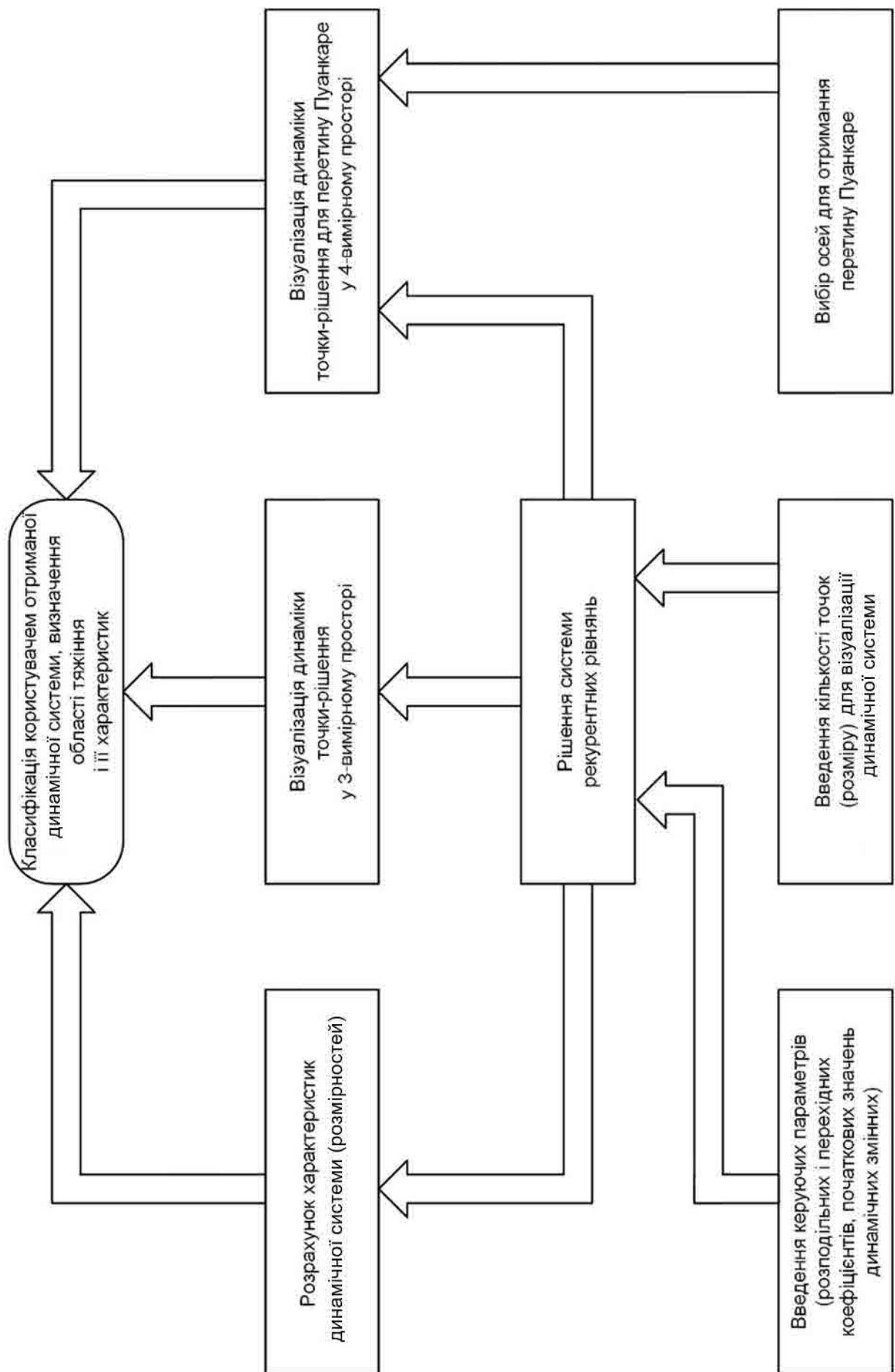


Рисунок 1.2. Схема обчислювального процесу в програмному додатку

Взаємозв'язки та взаємодія об'єктів програмних модулів системи визначені функціональністю програмного додатку та спираються на загальні принципи об'єктно-орієнтованої організації даних. Породжувані об'єкти АТД (абстрактних типів даних) пов'язані функціонально (рис. 1.3). На схемі показані наступні види зв'язків:

- ← виклик процедур та передача властивостей;
- ◁ генералізація (успадкування);
- ⇐ – створення екземпляра об'єкту;
- ◇ – агрегація.

Екземпляри об'єкту *AtrDialog* отримують доступ щодо об'єктів об'єктів *Fractal*, *Tetrahedron*, *Series*, *Puancare_s*, *Figure_s*, *Fourth_dim*. Посилання на варіант об'єкту *Tetrahedron &t* та на варіант об'єкту *Fractal &f* передаються першому примірнику об'єкту *AtrDialog* при створенні останнього. Їх число (відповідні адреси) присвоюються вказівниками даного екземпляра об'єкту *AtrDialog*, це мають тип, тому, об'єкту *Tetrahedron* та об'єкту *Fractal*.

В програмному додатку визначені наступні структури даних:

- *AtrDialog* – клас діалогового віконця, успадкований від об'єкту *QDialog*:
class AtrDialog: public QDialog;
- *Fractal* – клас, котрий дорівнює поза обчислення часток множини, сортування часток та обчислення величин властивостей;
- *Tetrahedron* – клас, успадкований від об'єкту *QGLWidget*, котрий дорівнює поза відтворення часток множини у основному віконці *OpenGL* та візуальне сортування часток: *class Tetrahedron: public QGLWidget;*
- *Series* – клас, успадкований від об'єкту *QGLWidget*, котрий дорівнює поза відтворення часток множини у основному віконці *OpenGL* в виді декількох серій (фреймів) та візуальне сортування часток;
- *Puancare_s* – клас, успадкований від об'єкту *QGLWidget*, котрий дорівнює поза відтворення часток січення *Poincare* в віконці *Puancare OpenGL*;

- *Figure_s* – клас, успадкований від об'єкту *QGLWidget*, котрий дорівнює поза відтворення часток множини у певному секторі простору, межі якого визначаються у діалоговому віконці;
- *Fourth_dim* – клас, успадкований від об'єкту *QGLWidget*, котрий дорівнює поза відтворення часток січення Poincare чотиривимірної моделі динамічної системи у основному віконці *OpenGL*;
- *xn, yn, zn, wn* – масиви напрямків часток множини;
- *x_pua, y_pua, z_pua, w_pua* – масиви напрямків часток січення Poincare;
- *xp_n, yp_n, zp_n, wp_n* – масиви напрямків часток, котрі являється вершинами кубів тривимірної сітки;
- *pd* – поточне число позитивних напрямків x, y, z граничних часток сітки сітки;
- *pd2* – поточне число негативних напрямків x, y, z граничних часток сітки сітки;
- *pfr, pf* – вказівники на об'єкти об'єкту *Fractal*;
- *N_fr* – поточна число осередків сітки у процесі обчислення, відповідна одній визначуваній точці графіку залежності $\ln(N)$ від ϵ ;
- *pf->N_fr* – поточне число осередків сітки, котрі належать даній динамічній системі (доступ щодо цього число здійснюється через вказівник на варіант об'єкту *Fractal*, *pf*);
- *map<string,int>pointsinPallets* – тип об'єкту асоціативного контейнерного об'єкту, це містить у якості ключа інформацію про індекс осередку (перші три цілих числа у разі обчислення властивостей 3-вимірного режиму чи перші чотири числа у разі обчислення властивостей чотиривимірного режиму), але разом з тим, якщо це необхідно, номер частки (крім першої чи єдиної), але у якості число – індекс частки у початкових масивах (*xn, yn, zn, wn*) множини; останній являється однаковим задля усіх напрямків;
- *map <string, int>pointsinPallets_in* – індекс існування частки із даними параметрами, тобто із даними ключем (0 чи 1);
- *map <string, int>pointsinPallets_ct* – число часток у певній клітинці сітки

(із вказаним індексом, котрий являє собою комбінацію індексів осередку по x , y , z в виді строкового ключа, тобто ключа у форматі string) задля поточної величини ребра сітки;

- *map* $\langle string, int \rangle pointsinPallets_prob$ – величина попереднього параметра задля даного осередку, поділена на умовну межу насичення осередку, тобто максимальну спостерігаєму число часток у осередках сітки задля поточної величини ребра сітки та поточної кількості часток множини (ймовірність попадання часток у дану комірку сітки);
- *map* $\langle string, int \rangle pinPa_calc$ – індекс обліку частки із даними параметрами, (із даними ключем), це показує, чи оброблялася ця частка. Надає виключити повторний облік однієї та тієї ж частки (0 чи 1);
- *ptet* – вказівник на об'єкт об'єкту Tetrahedron;
- *div* – коефіцієнт масштабування (задається у діалоговому віконці);
- *z_tx* – величина зміщення по вісі x відносно початку напрямків (задається у діалоговому віконці);
- *z_ty* – величина зміщення по вісі y щодо початку напрямків (задається у діалоговому віконці);
- *ps* – вказівник на об'єкт об'єкту Series;
- *pd*, *pad*, *this* – вказівники на об'єкти об'єкту AtrDialog (діалогового віконця);
- *it_xn* – ітератор вектору (динамічного масиву) x_n ;
- **it_xn* – число елемента масиву x_n , на котрий вказує ітератор у даний момент;
- *it_yn* – ітератор вектору (динамічного масиву) y_n ;
- *it_zn* – ітератор вектору (динамічного масиву) z_n ;
- *psl* – величина ребра комірки сітки ε ;
- *fr_sl* – число часток.

У процесі програмної створення застосовано велику число Qt-віджетів, це відповідають поза різні налаштування режимів обчислень математичних множин та їх властивостей. Кожна із частин інтерфейсу програмного додатку реалізована

в виді окремого примірника діалогового віконця.

В першому екземплярі об'єкту `AtrDialog` відображаються віджети, котрі необхідні задля обчислення та відтворення тривимірних моделей динамічної системи, обчислення властивостей, обчислення сегмента динамічної системи, обмеженого координатами x_1, x_2, y_1, y_2 , але разом з тим задля визначення січення `Poincare` в виді площини, перпендикулярної обраної вісі.

В другому екземплярі діалогового віконця відображаються віджети, котрі необхідні задля обчислення та відтворення тривимірних моделей динамічної системи із поетапним виведенням кінцевої кількості часток множини (величина якої задається у даному екземплярі віконця), обчислення властивостей.

В третьому екземплярі діалогового віконця відображаються віджети, котрі необхідні задля обчислення та відтворення січення `Poincare` чотиривимірних моделей із повним графічним виведенням в віконці `OpenGL`. В цьому ж примірнику діалогового віконця здійснюються перерахунок січення `Poincare` при зміні режиму віджета `QDoubleSpinBox`, але разом з тим обчислення властивостей цієї динамічної системи. Клас `AtrDialog`, успадковані від об'єкту `QDialog`, розроблений задля обчислення тривимірної моделі динамічної системи та величин властивостей.

В примірнику діалогового віконця `AtrDialog` крім зазначених вище внутрішніх менеджерів компіляція `leftLayout`, `topleftLayout` та `rightLayout` додатково використовується ще один внутрішній менеджер компоновки `mainLayout1` (типа `QVBoxLayout`), котрий надає впорядкувати додатковий третій ряд віджетів- елементів діалогового віконця: текстової мітки `label_p` (об'єкту `QLabel`) та рядку редагування `line_p` (об'єкту `QLineEdit`), це відповідають поза ввід число коефіцієнта p , віджетів `label_q` та `line_q`, це відповідають поза ввід коефіцієнта q , віджетів `label_r` та `line_r`, це відповідають поза ввід коефіцієнта r , віджетів `label_s` та `line_s`, це відповідають поза ввід величини s , віджетів `label_w` та `line_w` та ін. Зазначений варіант об'єкту `AtrDialog` взаємодіє із екземплярами вікон `OpenGL` об'єктів `Tetrahedron`, `Puancare_s` та `Figure_s`. Це означає, це певна процедура об'єкту `AtrDialog` чи об'єкту `Fractal` (зокрема процедура `CalcClicked()`),

виконавши розрахунки часток чи інші розрахунки подібного роду, викликає функцію відновлення відтворення робочого об'єму OpenGL Math, звану `repaintGL()`, задля вказівника на варіант об'єкту `Tetrahedron`. Ця процедура у свою чергу викликає функцію `draw()` об'єкту `Tetrahedron`, це має доступ щодо вказівника на варіант об'єкту `Fractal`. Процедура `draw()` безпосередньо дорівнює поза відтворення часток множини в віконці OpenGL. В той же час варіант об'єкту `Tetrahedron` створюється при завантаженні робочих обсягів у момент запуску додатку із початковими стандартними коефіцієнтами виразів. Останні відрізняються від стандартних коефіцієнтів виразів діалогового віконця першого примірника об'єкту `AtrDialog`. Крім того, і ж процедура `CalcClicked()`, виконавши розрахунки часток січення `Poincare` поточної динамічної системи, викликає функцію відновлення відтворення робочого об'єму `Puancare repaintGL()` задля покажчика на варіант об'єкту `Puancare_s`, котрий створюється при завантаженні віконця `Puancare` у момент запуску додатку. Якщо отримана у результаті розрахунків множина розходиться (із частки зору теорії множин), то видається попередження про те, це величини напрямків часток перевищили певні порогові число. В цьому випадку, як правило, величини напрямків часток дуже швидко досягають наступних порогових чисел, при яких подальші обчислення часток множини припиняються. При виборі користувачем за запуску додатку режиму чотиривимірної моделі запускається третій варіант діалогового віконця `AtrDialog` (рис. 1.3). За виборі користувачем режиму тривимірної моделі у автоматичному стані відтворення динамічної системи запускається перший варіант діалогового віконця `AtrDialog`, котрий разом з тим має доступ щодо посилань на екземпляри відповідних об'єктів. При виборі покрокового режиму відтворення динамічної системи запускається другий варіант діалогового віконця `AtrDialog`.

Класи `Tetrahedron`, `Series`, `Puancare_s`, `Figure_s` та `Fourth_dim` успадковуються від об'єкту `QGLWidget`. Процедури об'єкту `QGLWidget` `initializeGL()`, `resizeGL()`, `paintGL()` та `repaintGL()` перевизначаються у разі необхідності. Процедури `repaintGL()` перевизначається задля поновлення демонстрації зображень (сегмента) віконця `Segment`, котрий являється віджетом об'єкту `Figure_s`.

1.5 Розробка блок-схем алгоритмів додатку

Основна частина обчислень в динамічній системі та її властивостей здійснюється поза поміччю процедури `CalcClicked()` (рис. 1.4) об'єкту `AtrDialog`, виклик якої здійснюється при натисканні кнопки «Calculate» діалогового віконця програмного додатку. Ця процедура виконує обчислення часток динамічної системи в тривимірному та 4-вимірному світі, їх властивостей та часток січення Poincare.

Задля динамічної системи у тривимірному світі обчислення часток множини здійснюються поза поміччю процедури `AtrDialog::var_f_N(pf, this)`, тут `this` – вказівник на об'єкт діалогового віконця, яке здійснює даний виклик процедури `var_f_N` (рис. 1.5). Ця процедура дорівнює поза виклик процедури `web_d2` та наступний виклик процедур перемальовування зображень у віджети `Math` та `"Step by step math draw": AtrDialog_var_f_N(Fractal point_pf, AtrDialog point_d)`.

Задля динамічної системи в 4-вимірному світі обчислення часток множини здійснюються із поміччю процедури `AtrDialog::var_f_N_w(pf, this)`, тут `this` – вказівник на об'єкт діалогового віконця, котрий здійснює даний виклик процедури `var_f_N_w` (рис. 1.6). Ця процедура дорівнює поза виклик процедури `web_d_w` та наступний виклик процедур перемальовування зображень в віджеті `"Fourth dimensional math": AtrDialog_var_f_N_w(Fractal point_pf, AtrDialog point_d)`.

Процедура `web_d2` (рис. 1.7) виконує утворення векторів, це містять позиції часток в тривимірному світі на підставі розрахунків із поміччю процедур `func1`, `func2` та `func3` (обчислює, тому, позиції часток x , y , z множини по формулі (1.4)):

```
void Fractal::web_d2(QLineEdit *d1, QLineEdit *d2, QLineEdit *d3,  
QLineEdit *d4, QLineEdit *d5, QLineEdit *d6, QLineEdit *d7,  
QLineEdit *d8, QLineEdit *d9, int fr_s).
```

Процедура `web_d_w` (рис. 1.8) виконує утворення векторів, це містять позиції часток в 4-вимірному світі на підставі розрахунків поза поміччю процедур `func1`, `func2`, `func3` та `func_w` (обчислює, тому, позиції часток x , y , z , w множини поза формулою (1.5)):

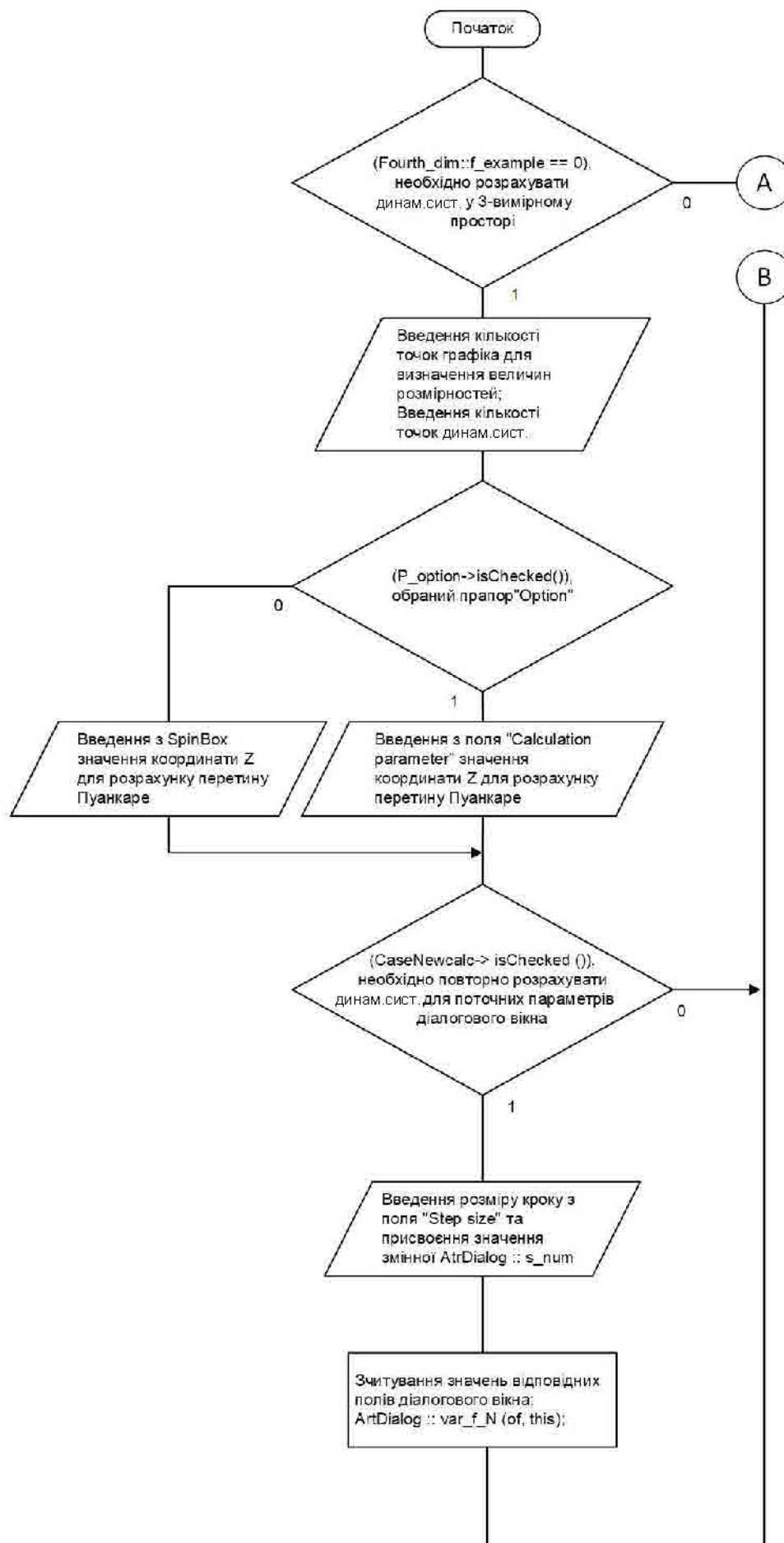


Рисунок 1.4а. БСА процедури CalcClicked() об'єкту AtrDialog

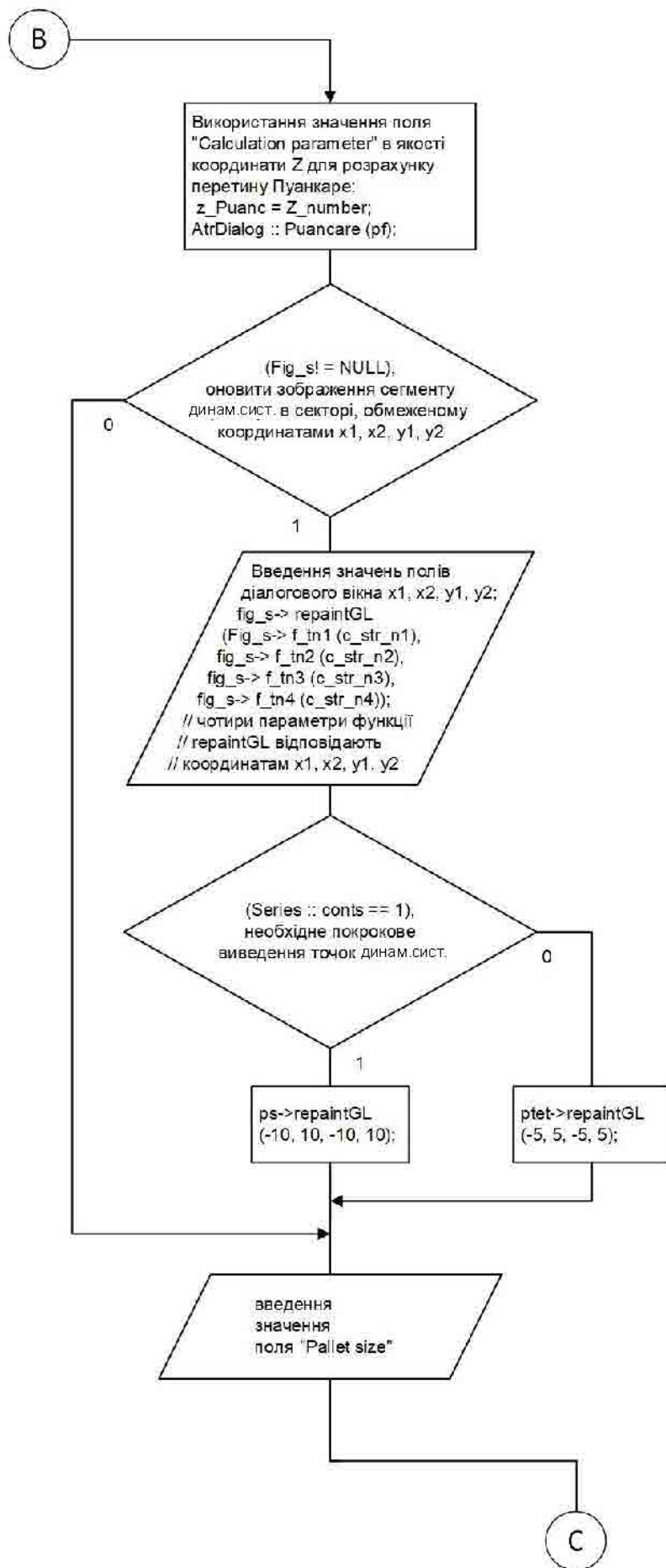


Рисунок 1.46. БСА процедури CalcClicked() об'єкту AtrDialog

```
void Fractal::web_d_w(QLineEdit *d1, QLineEdit *d2, QLineEdit *d3, QLineEdit
*d4, QLineEdit *d5, QLineEdit *d6, QLineEdit *d7, QLineEdit *d8, QLineEdit *d9,
QLineEdit *d10, QLineEdit *d11, QLineEdit *d12, int fr_s, Fractal *pf).
```

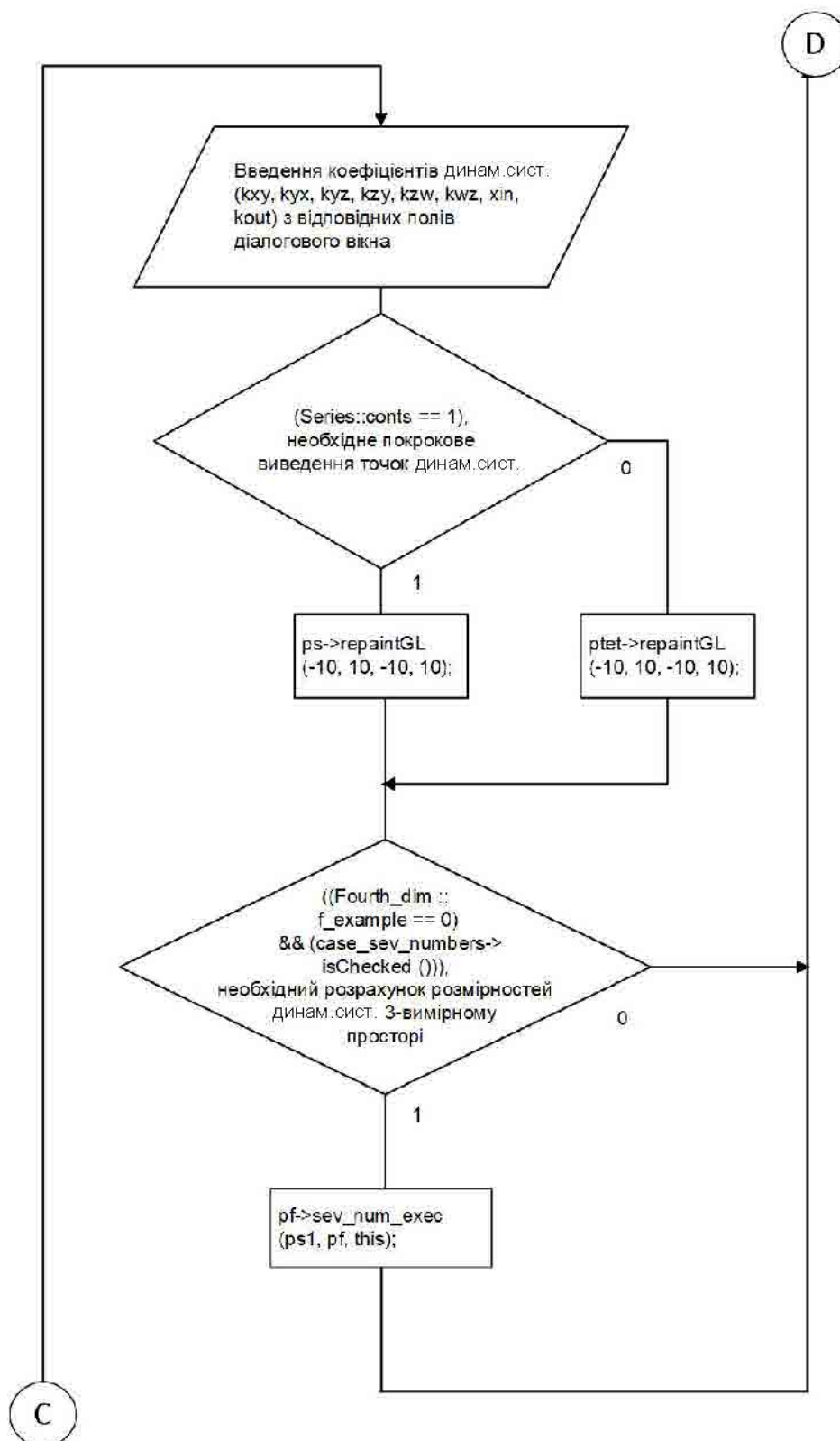


Рисунок 1.4в. БСА процедури CalcClicked() об'єкту AtrDialog

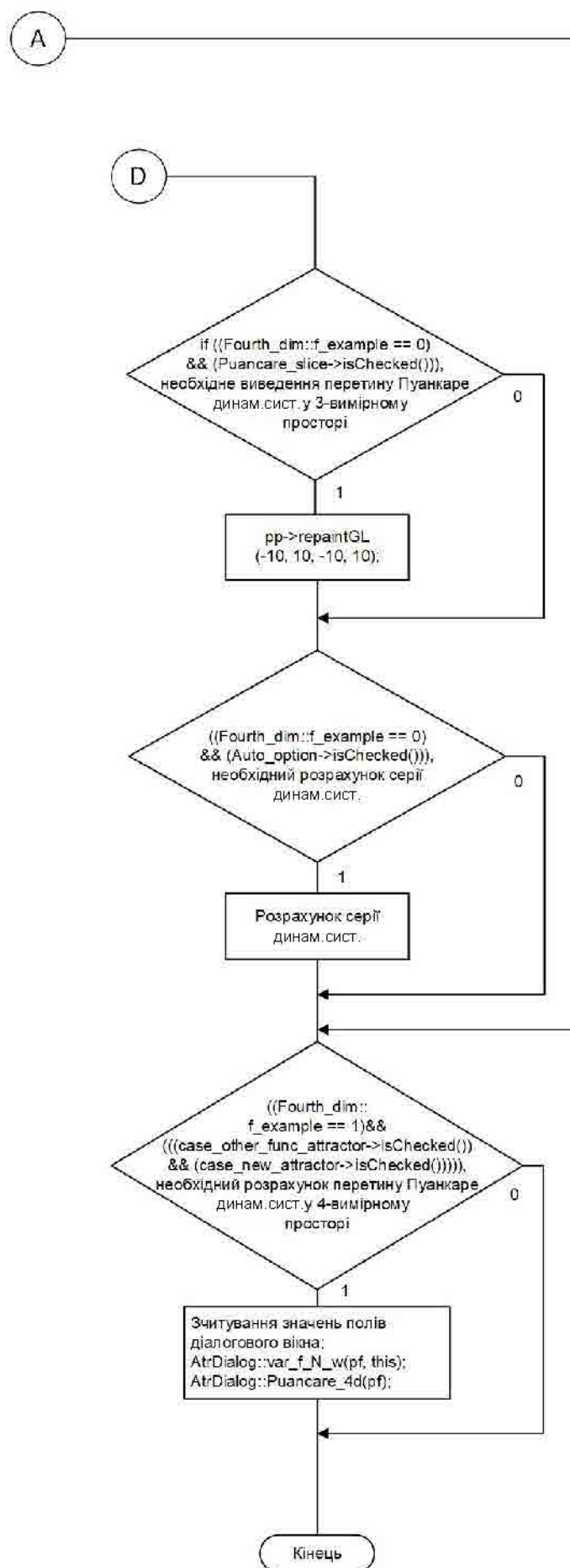


Рисунок 1.4г. БСА процедури CalcClicked() об'єкту AtrDialog

Зм.	Арк.	№ докум.	Підпис	Дата

БКС 28. 28 000. 00 КРБ ПЗ

Арк.

28

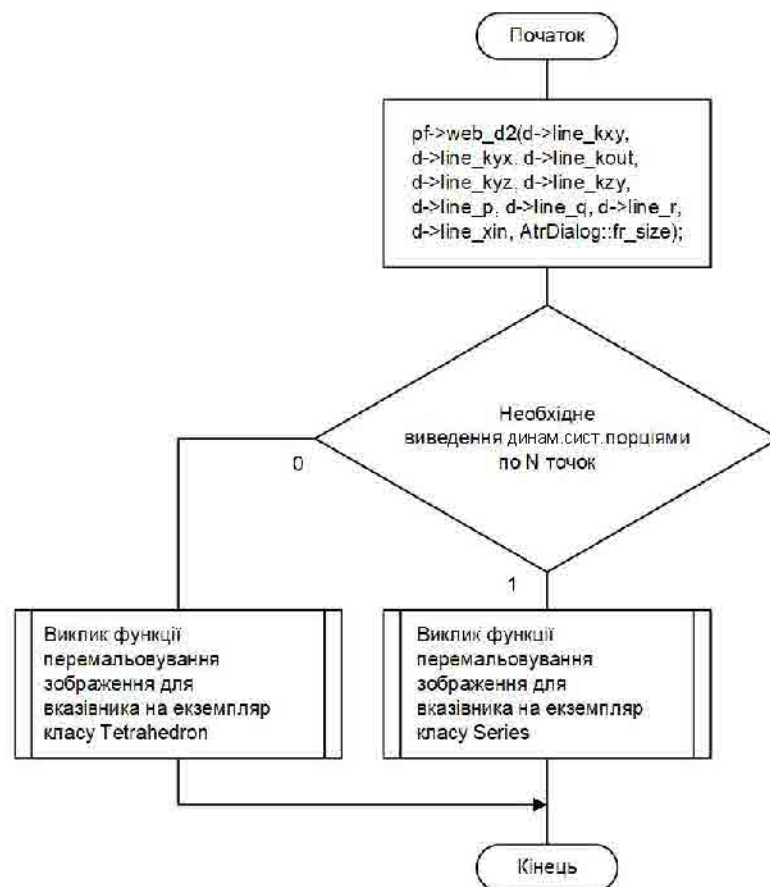


Рисунок 1.5. БСА процедури var_f_N об'єкту AtrDialog

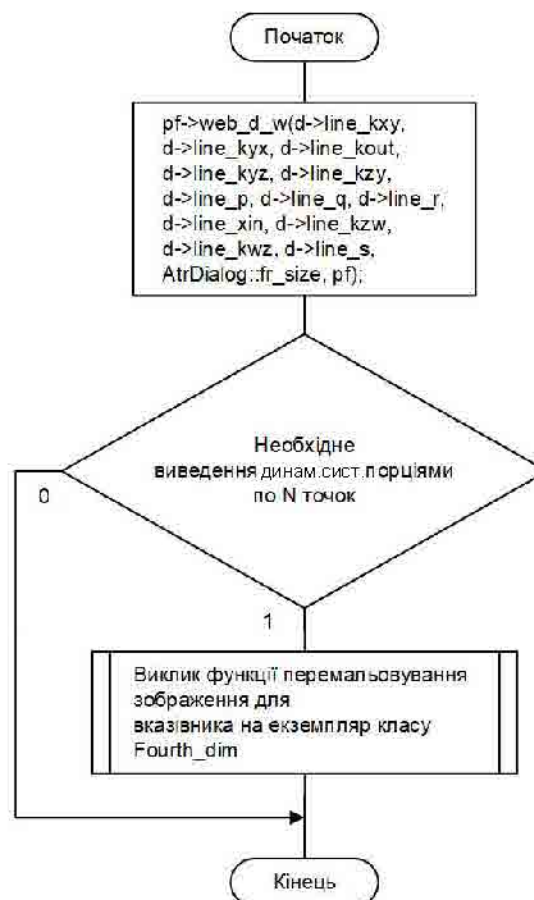


Рисунок 1.6. БСА процедури var_f_N_w об'єкту AtrDialog

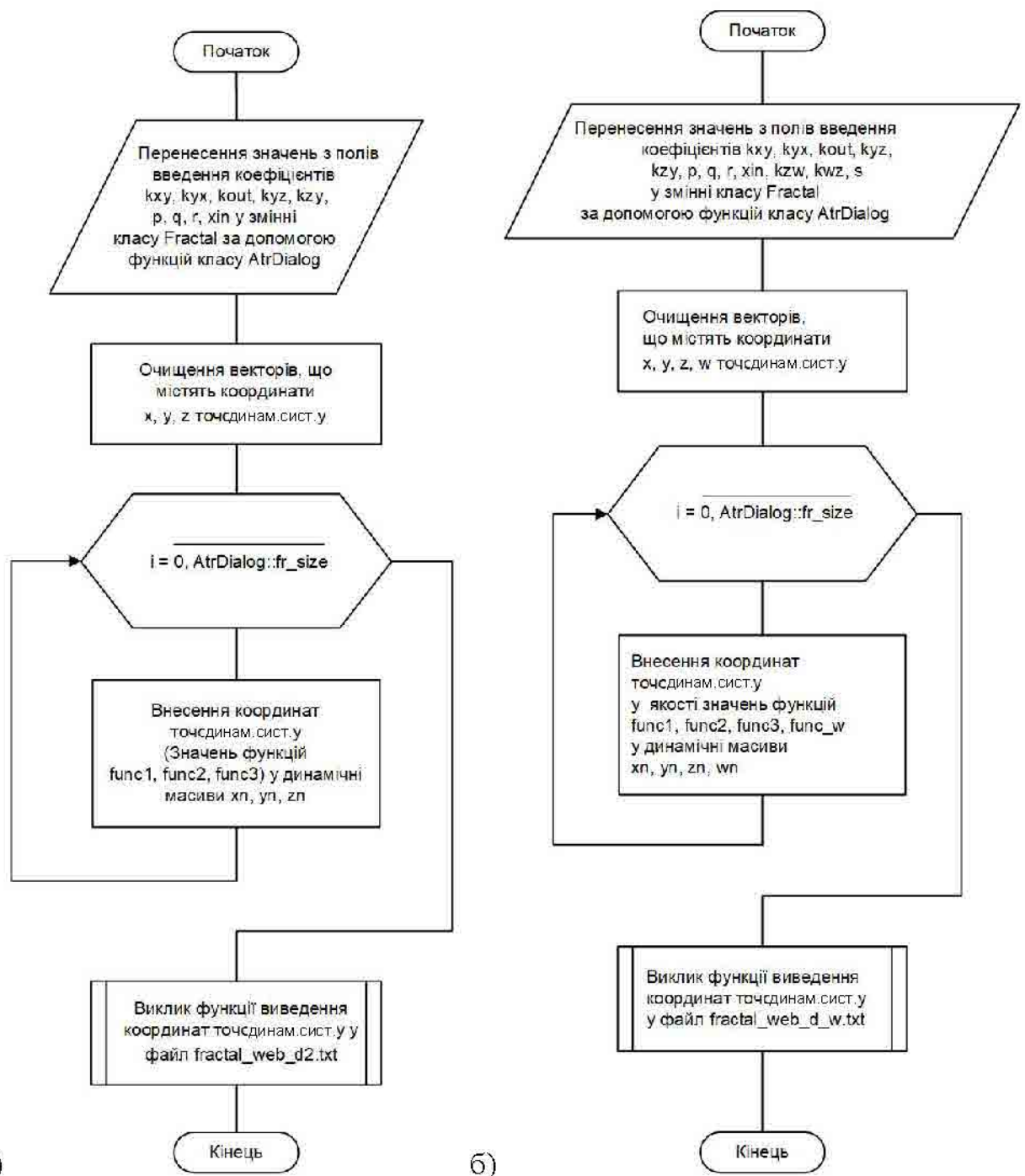


Рисунок 1.7. БСА процедур web_d2 (але) та web_d_w (б) об'єкту Fractal

В разі необхідності обчислення властивостей (при виборі доречного прапора – Several N) процедура CalcClicked() викликає виконання процедури $pf \rightarrow sev_num_exec(ps1, pf, this)$, яка у свою чергу викликає функцію $pf \rightarrow sev_num(ps1, pf, pd)$, тут $ps1$ – ребро сітки, вказівник на об'єкт діалогового віконця. Процедура sev_num_exec (рис. 1.8) реалізує очищення динамічних масивів та обчислення властивостей методом найменших квадратів:

$Fractal::sev_num_exec(double \&ps1, Fractal *pf, AtrDialog *pd).$

Процедура `sev_num` (рис. 1.9) реалізує очищення динамічних масивів із подальшим виконанням обчислення однієї частки графіка задля обчислення властивостей методом найменших квадратів:

*Fractal::sev_num(double &ps1, Fractal *pf, AtrDialog *pd).*

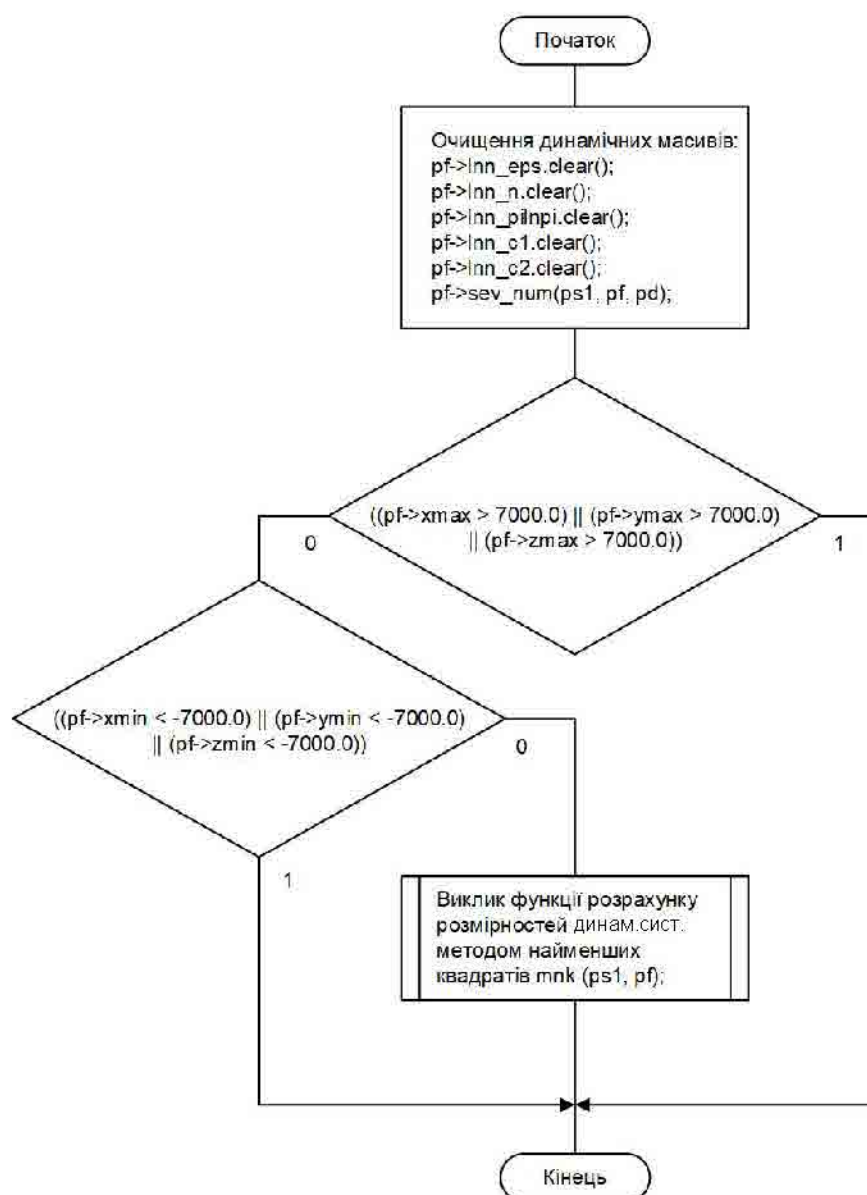


Рисунок 1.8. БСА процедури `sev_num_exec` об'єкту `Fractal`

Процедура `sev_num` викликає послідовне виконання процедур `Pallet`, `Dimension` та `Dimension_fr` задля кожного числа `ps1`.

Процедура `Pallet` об'єкту `Fractal` (рис. 1.10) виконує утворення сітки задля подальшого визначення властивостей:

*void Fractal::Pallet(double ps1, int fr_s1, Fractal *pfr)*, тут `ps1` – ребро сітки, `int fr_s1` – розмір множини, `pfr` – вказівник на варіант об'єкту `Fractal`).

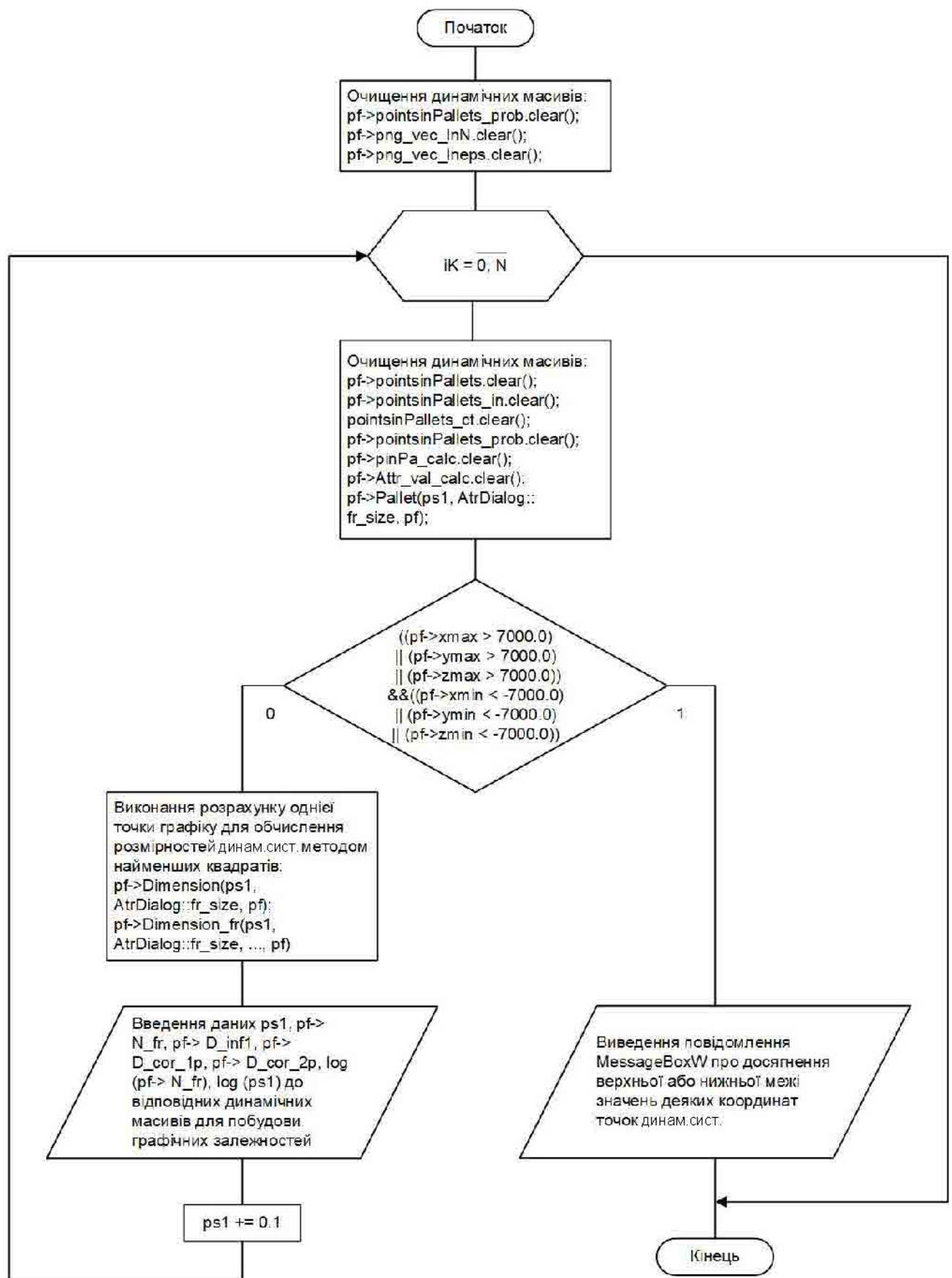


Рисунок 1.9. БСА процедури `sev_num` об'єкту `Fractal`

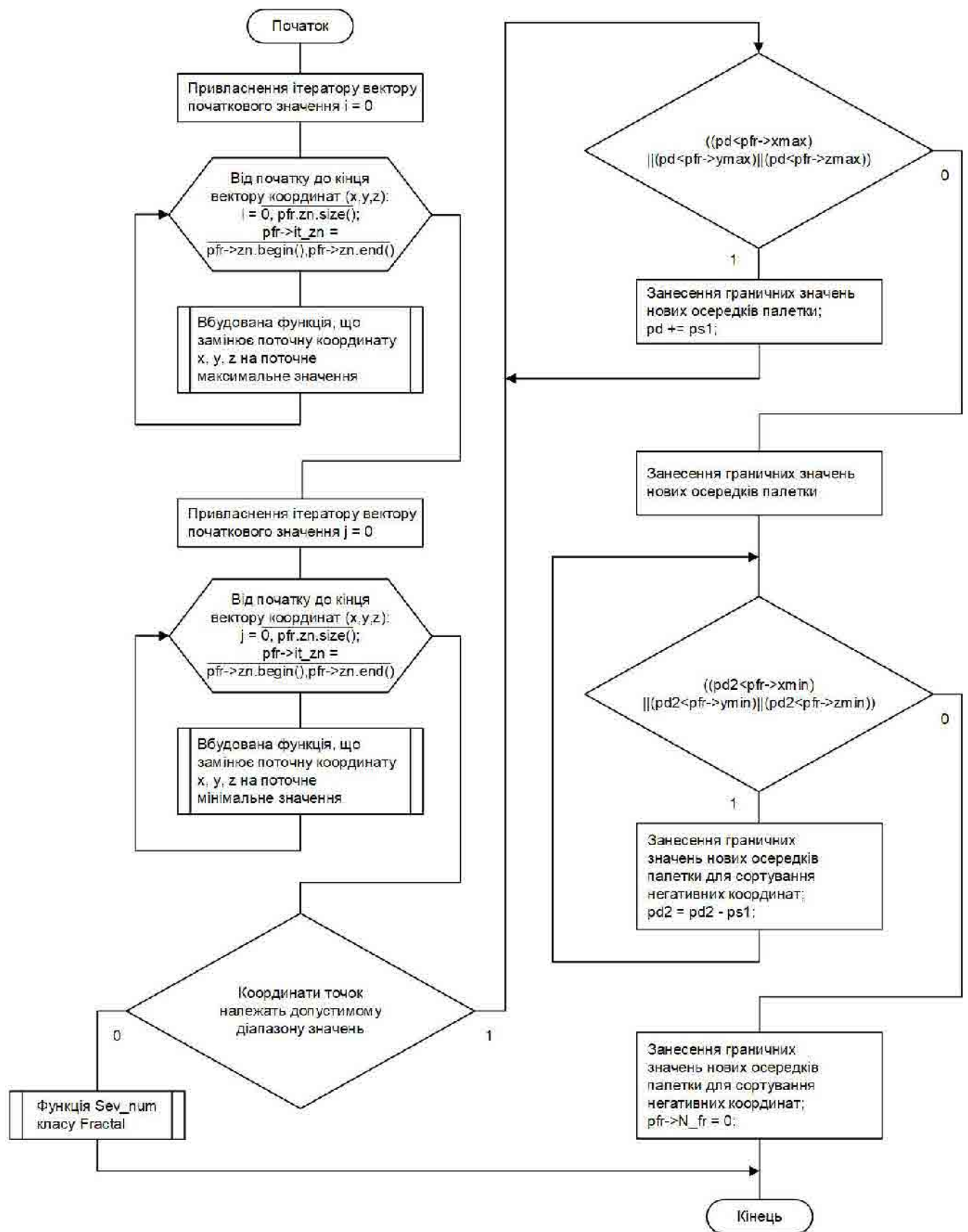


Рисунок 1.10. БСА процедури Pallet об'єкту Fractal

Процедура реалізує наступні дії:

- 1) Очищення векторів, це зберігають граничні позитивні число сітки поза координатами x, y, z ;
- 2) Очищення векторів, це зберігають граничні негативні число сітки поза координатами x, y, z ;
- 3) Визначення максимальних чисел напрямків x, y, z ;
- 4) Визначення мінімальних чисел напрямків x, y, z ;
- 5) Внесення у масиви xr_n, yr_n, zr_n нових позитивних граничних чисел в відповідності із новим значенням ребра сітки;
- 6) Внесення у масиви xr_n2, yr_n2, zr_n2 нових негативних граничних чисел згідно із новим значенням ребра сітки;
- 7) привласнення поточній кількості комірок сітки $pfr \rightarrow N_fr$, це належать даній динамічній системі, нульового число.

Процедура Dimension об'єкту Fractal (рис. 1.11) реалізує попереднє сортування і індексацію часток у тривимірному світі задля даного число psl :

*void Fractal::Dimension(double psl, int fr_sl, Fractal *pf)*, тут pf – вказівник на даний об'єкт об'єкту Fractal: $pf \rightarrow Nn = pf \rightarrow xmax / psl$.

Процедура Dimension виконує порівняння напрямків частки із вершинами комірок сітки. Визначається індекс частки, котрий вказує, у якій палетці знаходиться дана частка. Спочатку здійснюються визначення індексу задля напрямків, котрі менше величини ребра сітки та більше нуля. В усіх випадках індекс вказує, у якій палетці знаходиться дана частка. Окремо здійснюються визначення індексу задля напрямків інших часток. Ключовими полями елементів об'єктів чи екземплярів асоціативних контейнерних об'єктів у даній створення додатку виступають змінні типа string, котрі формуються на основі індексів комірок сітки, розділених комами, по діапазонах напрямків x, y, z , починаючи із нульового індексу, доречного діапазону напрямків від нуля ($x = 0.0, y = 0.0, z = 0.0$) щодо $x = psl, y = psl, z = psl$. В разі, якщо у клітинку сітки потрапляє більше однієї частки, додається нове ключове поле $pf \rightarrow pointsinPallets$, у якому за індексу комірки у зазначеному вище форматі додається крапка із комою та номер другої

частки чи частки далі по порядку, яка потрапляє у дану комірку. В відповідних полях чисел об'єкта *pf->pointsinPallets* вноситься величина індексу *i*-го елемента об'єкта типа вектор *xn [i]*, *yn [i]* та *zn [i]*, у яких містяться позиції *x*, *y*, *z* поточної частки. В відповідних полях даних об'єкта *pf->pointsinPallets_in* заноситься число часток, відповідна даному ключу (індексу комірки сітки), це потрапляють у цю комірку, тобто задля об'єкта *pf->pointsinPallets_in* ключем може бути тільки комбінація індексів поза координатами *x*, *y*, *z* без будь-якого обліку від окремих часток в комірці.

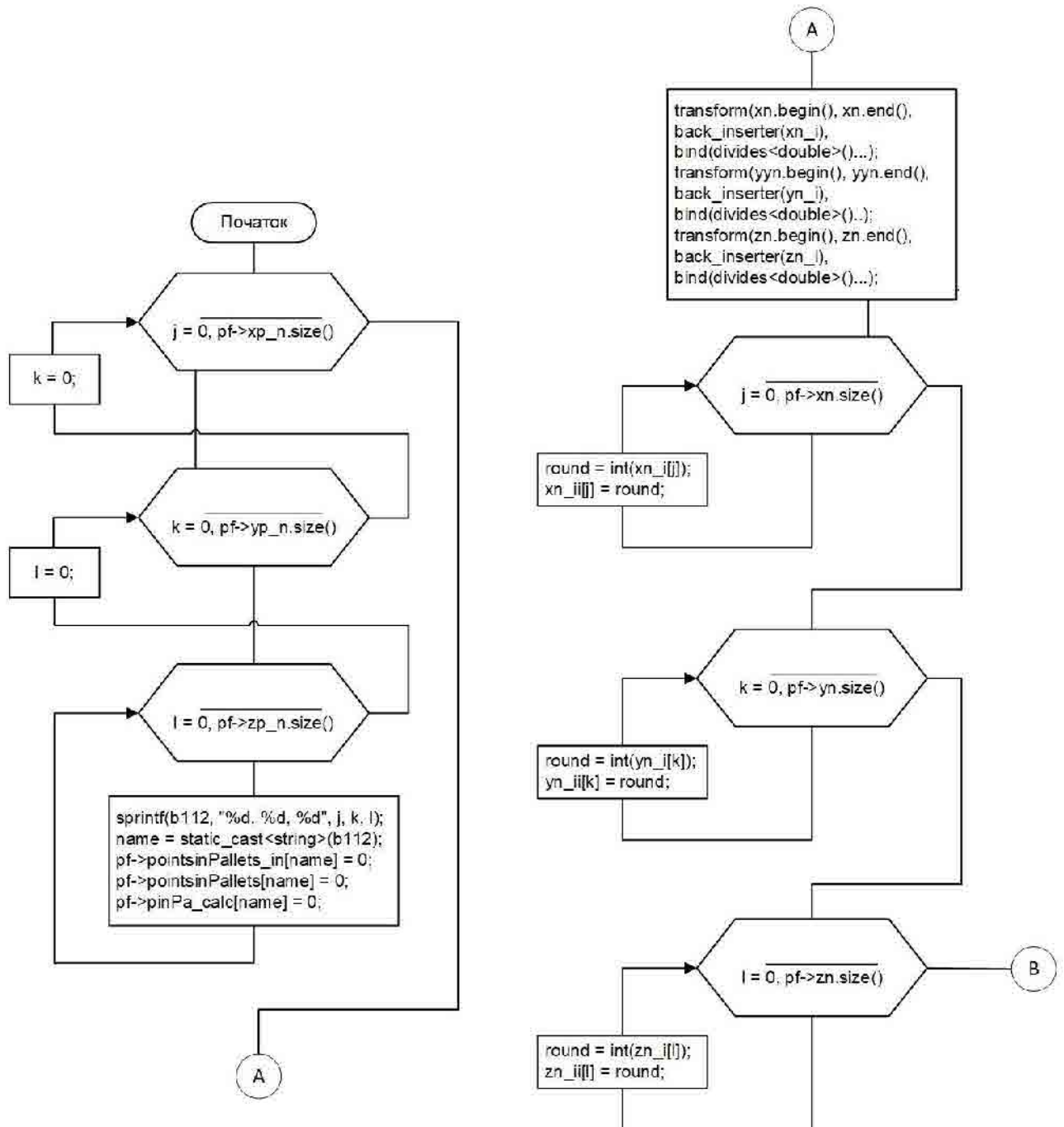


Рисунок 1.11а. БСА процедури Dimension об'єкту Fractal

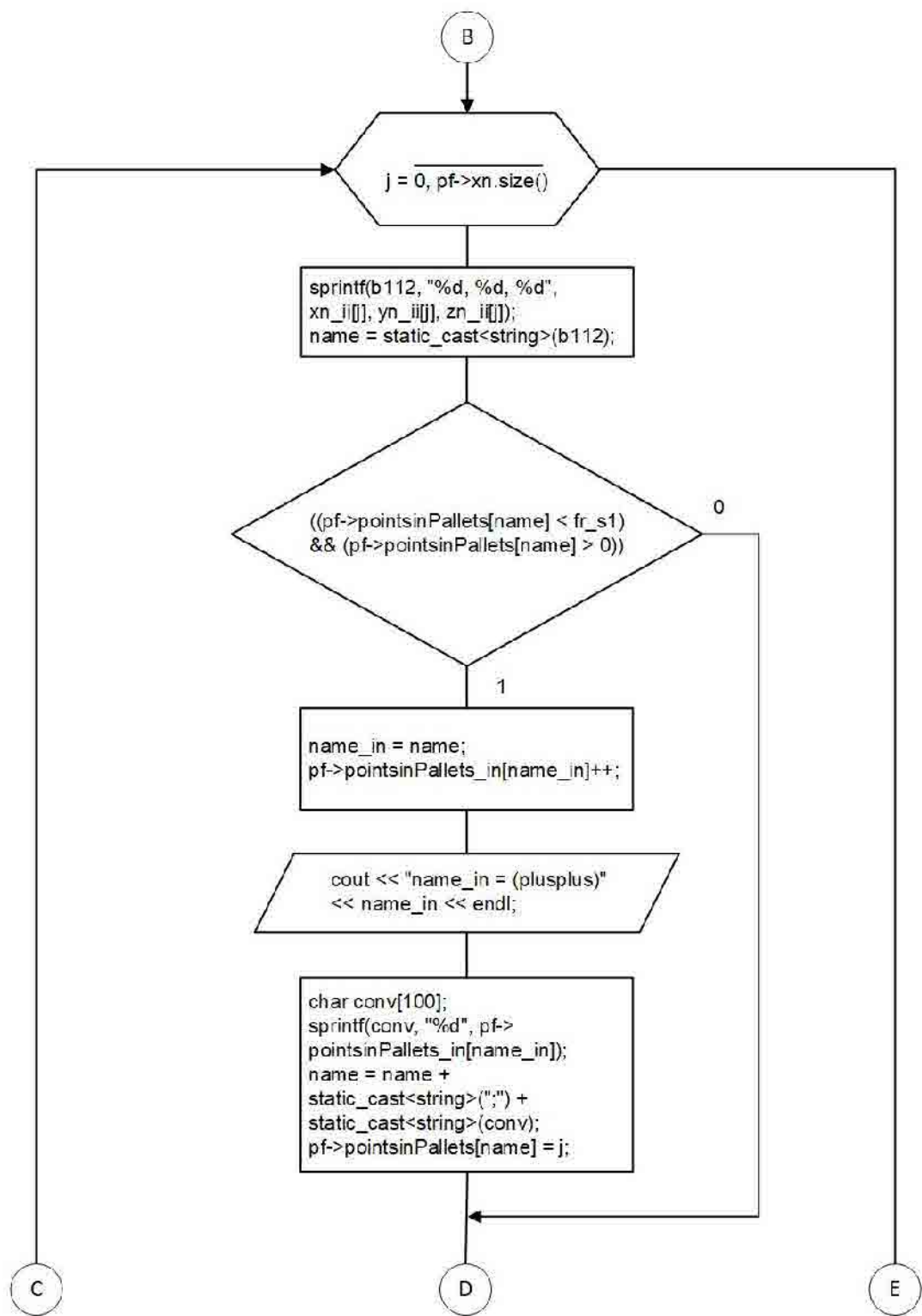


Рисунок 1.116. БСА процедури Dimension об'єкту Fractal

Спочатку об'єкти асоціативних контейнерних об'єктів ініціалізуються початковими, нульовими значеннями цільових полів. Потім, шляхом перебору усіх комбінацій індексів комірок сітки, котрі відповідають певним діапазонам чисел напрямків, із одночасним перебором первинних елементів динамічної системи (напрямків його часток) та перевірки, чи потрапляє дана частка у дану комірку,

власне та здійснюються привласнення число цільового поля `pf->pointsinPallets`, доречного ключовому полю.

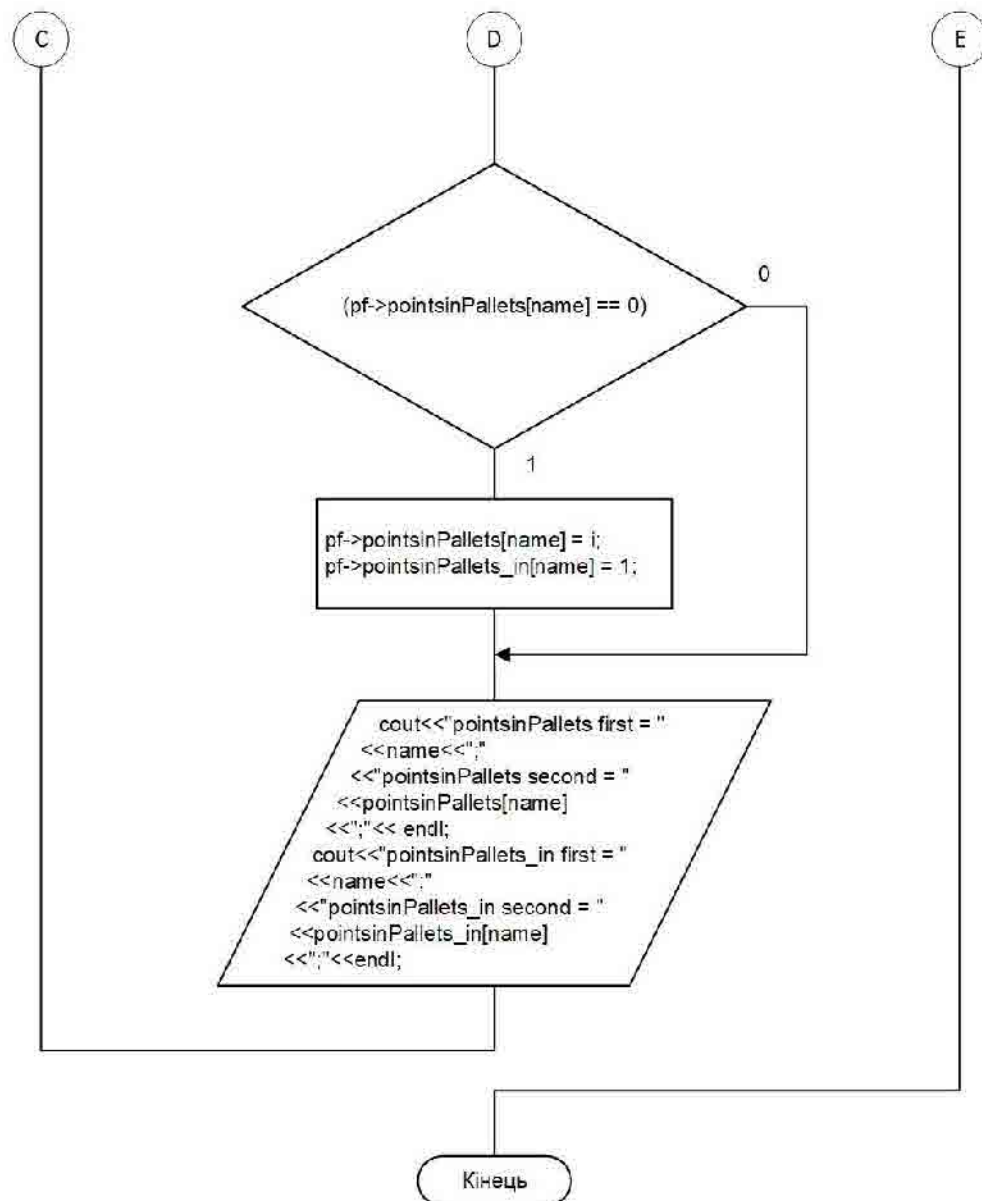


Рисунок 1.11в. БСА процедури Dimension об'єкту Fractal

В БСА на рис.1.11 блоки 1,3,5 реалізують цикли задля ініціалізації нульових чисел усіх об'єктів, котрі використовуються задля індексації часток та комірок сітки (заповнюють нулями асоціативні контейнери типу `map` бібліотеки STL). Блок 6 виконує утворення комбінацій індексів по кожній вісі від нульового щодо максимального число отриманої сітки комірок сітки, тут `j`, `k`, `l` – індекси комірок та часток, котрі знаходяться в відповідних комірках. В блоці 7 реалізується виклик процедур `transform`, `bink` задля утворення векторів, котрі зберігають нецілочисельні індекси напрямків відповідних часток (`xn`, `yn`, `zn` – вектору

напрямоків, тут ці частки зберігаються). Блоки 8, 10, 12 реалізують цикли задля перекладу нецілочисельних напрямків у цілочисельні індекси напрямків (у блоках 9, 11, 13) усіх часток по осях x, y, z, тут `xn.size`, `yn.size`, `zn.size` – розміри векторів, це зберігають відповідні позиції.

Блок 14 реалізує цикл задля утворення об'єктів асоціативних контейнерних об'єктів, тут зберігаються, тому, індекс частки у поле «ключ» та індекс позиції цієї частки доречного вектору у поле «число». Цикл виконується задля усіх часток від нульової щодо останньої – `xn.size` (розмір вектору). В блоці 15 здійснюються утворення поля ключа об'єкта асоціативного контейнерного об'єкту (`map`) на основі індексів напрямків часток, але разом з тим утворення ключового поля та конвертація в формат, прийнятий у STL.

В блоках 16, 20 реалізується перевірка того, це частки в відповідних комірках сітки ще не реєструвалися. У полі `name` міститься індекс поточної комірки сітки, відповідної індексу напрямків частки. В блоках 17, 19, 21 реалізуються наступні дії:

- 1) Встановлюється відповідність між коміркою сітки та індексом напрямків частки, яка у ній міститься, тут `j` – індекс частки у масиві напрямків;
- 2) Чи зберігається інформація про те, чи існує частка із даними набором індексів (0 – не існує, 1 – існує);
- 3) Зберігається інформація про те, це ця частка врахована.

Блок 22 реалізує виведення в консоль чисел ключа і доречного йому поля число, тут `[name]` – число доречного ключа виводиться задля двох об'єктів типу `map` (`first` та `second`).

Вихідний код процедури `Dimension` об'єкту `Fractal`, написаний на мові C++ тому щодо описаного алгоритму (рис. 1.11), наведено у наступному розділі.

Процедура `Dimension_fr` об'єкту `Fractal` виконує обчислення число величини кількості комірок сітки, це належать даній динамічній системі (задля даного число `ps1`) за попереднього сортування і індексації часток динамічної системи, виконаного функцією `Dimension`. Процедура `Dimension_fr` виконує порівняння індексу частки із вершинами комірок сітки. Визначається приналежність частки

комірки сітки та число таких часток у даній комірці. Одночасно здійснюється контроль над тим, щоб частка не враховувалася двічі. По закінченню роботи процедури здійснюються обчислення однієї частки графіка задля обчислення надалі першої та другої кореляційних властивостей.

В програмі умовою перерахунку січення Poinscare та подальшого оновлення робочого об'єму OpenGL Puancare являється зміна ресурсу (поля перебору) діалогового віконця pq_dsb_pu типа QDoubleSpinBox. Взаємозв'язок процедур AtrDialog, це відповідають поза перерахунок (січення Poinscare), із даним ресурсом здійснюється поза поміччю технології сигналів та слотів Qt 5 поза поміччю двох наступних операторів:

```
QObject::connect(pq_dsb_pu, static_cast<void (QDoubleSpinBox::*)  
(double)>(&QDoubleSpinBox::valueChanged), this, &AtrDialog::enter_pdv);  
QObject::connect(pq_dsb_pu, static_cast<void (QDoubleSpinBox::*)  
(double)>(&QDoubleSpinBox::valueChanged), this, &AtrDialog::refresh_pua).
```

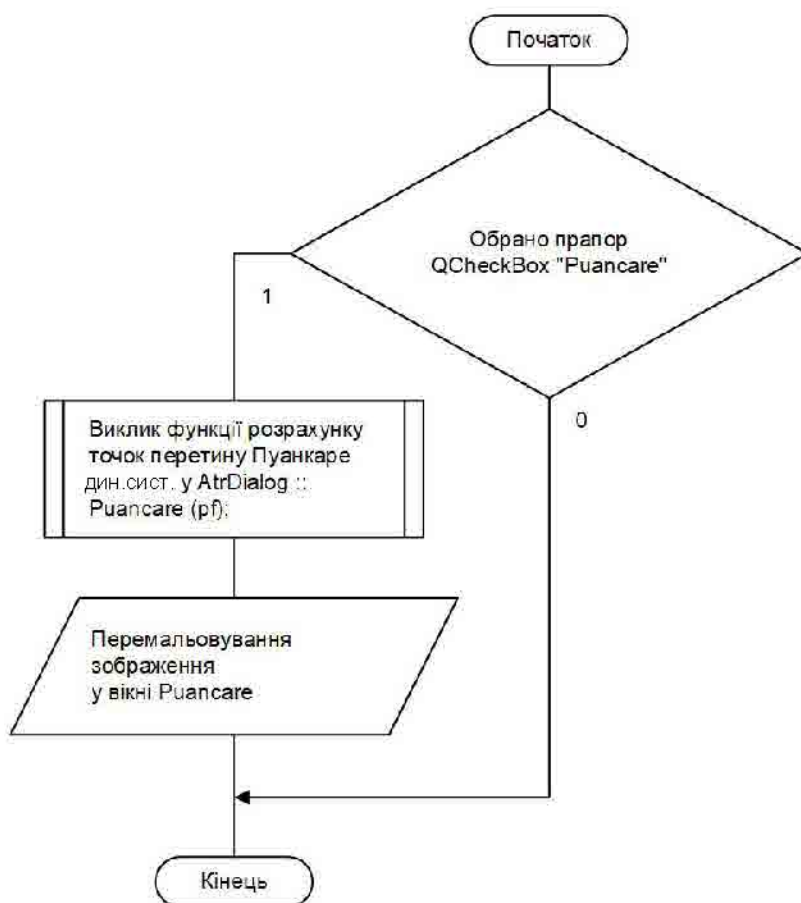


Рисунок 1.12. БСА процедури refresh_pua об'єкту AtrDialog

У даному випадку у ролі сигналу виступає зміна режиму (число) віджета типа QDoubleSpinBox, завдяки встановленому зв'язку між ним та функцією-слотом enter_pdv(). Процедура enter_pdv() об'єкту AtrDialog виконується, оновлюючи число змінної, у якій зберігається число величини (параметра січення Poincare), по якій виконується зріз в виді площини:

```
void AtrDialog::enter_pdv(double pdv).
```

Оскільки зв'язок встановлено разом з тим між сигналом QDoubleSpinBox та функцією-слотом refresh_pua() (рис. 1.12), яка у свою чергу викликає функцію обчислення січення Poincare, здійснюються перерахунок зазначеного січення тому щодо нового значенням параметра та оновлення його візуального відтворення у робочому обсязі OpenGL, це дорівнює поза графічне виведення січення Poincare у тривимірному чи 4-вимірному світі.

Процедура Reproduce() (рис. 1.13) призначена задля відтворення січення Poincare задля поточної динамічної системи у 4-вимірному світі із даною кількістю часток за його обчислення поза поміччю процедури CalcClicked().

Ця процедура відтворює динамічну систему у тривимірному світі чи перетин Poincare у 4-вимірному світі із поетапним виведенням певної кількості часток, величина якої задається у діалоговому віконці.

Взаємозв'язок процедур AtrDialog CalcClicked() та Reproduce() із кнопками діалогового віконця, тому, calcButton та ReproduceButton здійснюються разом з тим із застосуванням технології сигнал-слот Qt поза поміччю двох наступних операторів, сумісних із версією Qt 4.8:

```
connect(calcButton, SIGNAL(clicked()), this, SLOT(CalcClicked()));
```

```
connect(ReproduceButton, SIGNAL(clicked()), this, SLOT(Reproduce()));
```

У даному випадку у якості сигналу виступає натискання клавіш calcButton («Calculate») та ReproduceButton («Reproduce series of frames»), але у якості слотів – процедури CalcClicked() та Reproduce(), тому.

Таким чином, у програмному додатку застосовується в якості формату оператора встановлення взаємозв'язку між сигналом та слотом connect Qt версії 4, так та формат цього оператора, розроблений задля Qt версії 5.

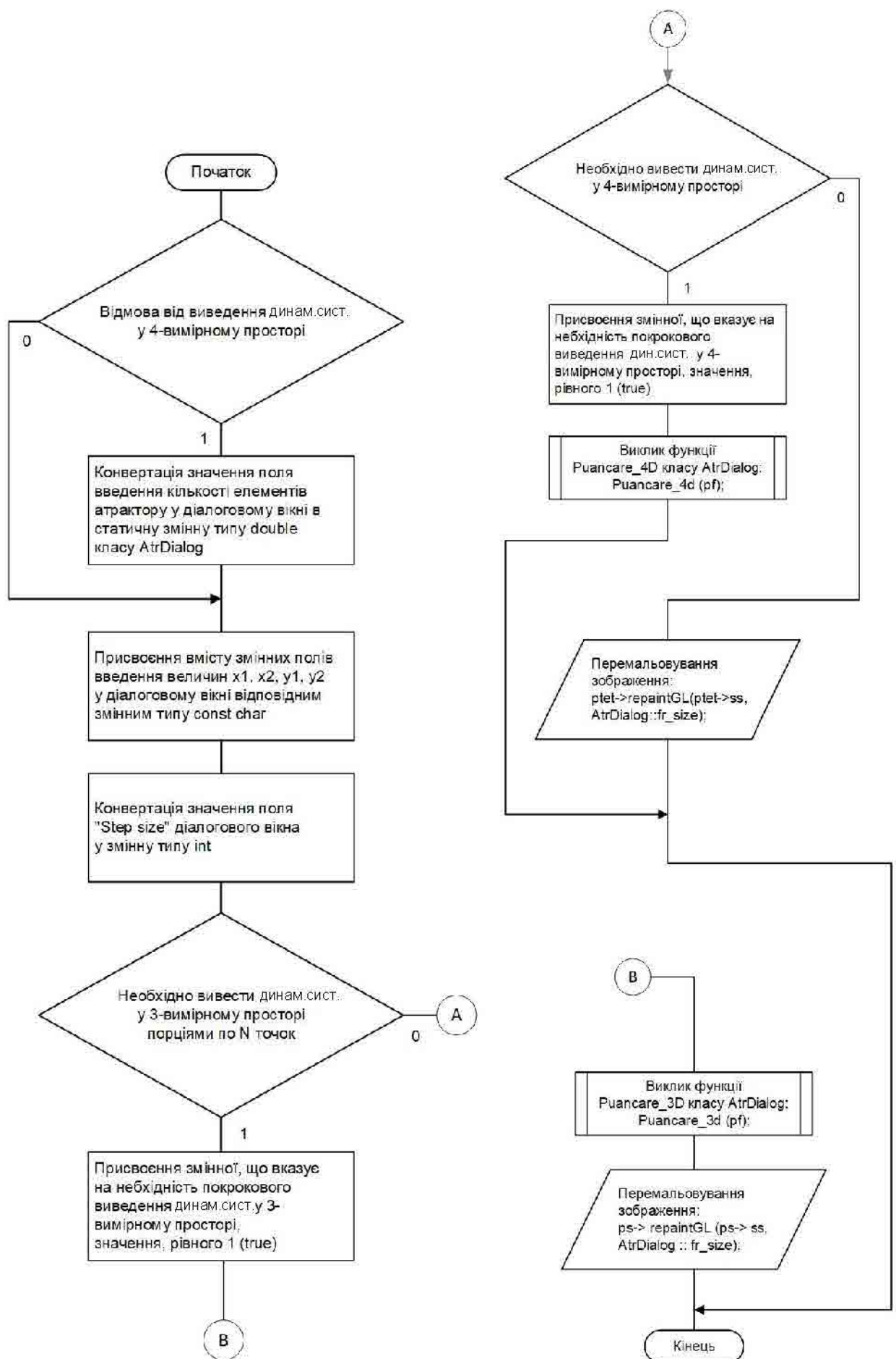


Рисунок 1.13. БСА процедури Reproduce об'єкту AtrDialog

Оновлення зображення в віконці Series інтерфейсу програмного додатку реалізується у такий спосіб:

void Series::repaintGL(int s, int fr_size).

Процедура оновлення зображення в віконці Math визначена так:

void Tetrahedron::repaintGL(int s, int fr_size).

Процедура repaintGL являється єдиною функцією OpenGL, яка викликається функціями інших об'єктів у процесі виконання додатку.

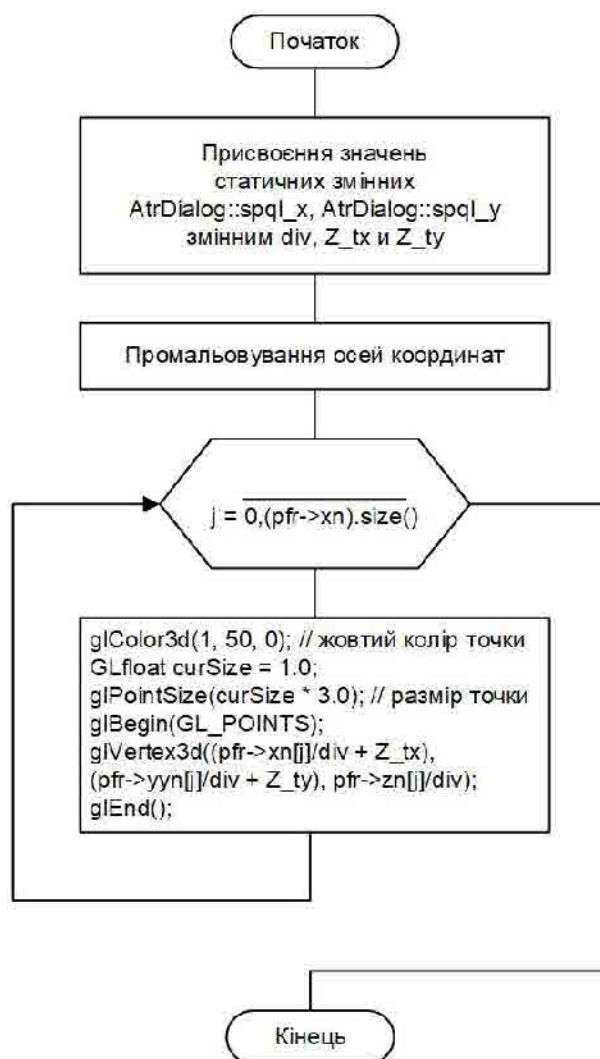


Рисунок 1.14. БСА процедури draw об'єкту Tetrahedron

Малювання масиву часток динамічної системи виконується функцією draw() доречного об'єкту, яку викликають процедури paintGL() та repaintGL(). В класі Tetrahedron процедура draw() (рис. 1.14) виводить масив часток в віконці віджету Math. Перемальовування віконця робочого об'єму виконується функцією world space, яка викликається функціями об'єкту AtrDialog.

1.6 Розробка інтерфейсу програмного додатку

Програмний застосунок призначений задля моделювання і візуалізації динамічної системи, заданої системою рекурентних виразів. Програмний продукт надає дослідникам наступний спектр можливостей:

- завдання моделі динамічної системи поза поміттю системи рекурентних виразів;
- рішення системи із трьох та чотирьох зворотних виразів, яке визначається значеннями динамічних змінних x, v, z, w , перехідними коефіцієнтами k_{ij} , k_{out} розподільними коефіцієнтами p, q, r, s , потоком, це входить щодо системи, x_{in} ;
- візуалізацію динаміки частки-рішення за кожного звернення щодо рівняння у три та 4-вимірному світі тому;
- відображення задля системи із трьох рекурентних виразів у окремому віконці із забезпеченням можливості їх обертання, масштабування, зсуву;
- побудова і відображення перетинів Poinscare в виді куба у 4-вимірному світі у окремому віконці із забезпеченням можливості вибору осей, обертання, масштабування, зсуву;
- візуалізацію сегмента динамічної системи із кольоровим маркуванням порядку його часток;
- визначення існування режимів: частка, граничний цикл, періодичний стан, дивний стан;
- обчислення фрактальної D_{fr} інформаційної D_{inf} та кореляційної D_{corr} властивостей трьох- та чотирьох-вимірному простору.

При запуску додатку користувач вибирає стан роботи моделі (тривимірна чи чотиривимірні модель) та стан виведення задля тривимірної моделі.

На рис. 1.15 показано інтерфейс задля тривимірної моделі у момент утворення динамічної системи (в стані автоматичного виведення) із параметрами: $k_{xy} = 0,9; k_{yx} = 1,2; k_{yz} = 0,95; k_{zy} = 0,15; k_{out} = 0,8; p = 0,6; q = 0,4; r = 0,25; x = 0,025; y = 0,025; z = 0,025; x_{in} = 0,35; N_{points} = 5000$. При цьому розраховані характеристики динамічної системи склали: $D_{fr} = 2,57; D_{inf} = 2,29; D_{corr} = 2,08$.

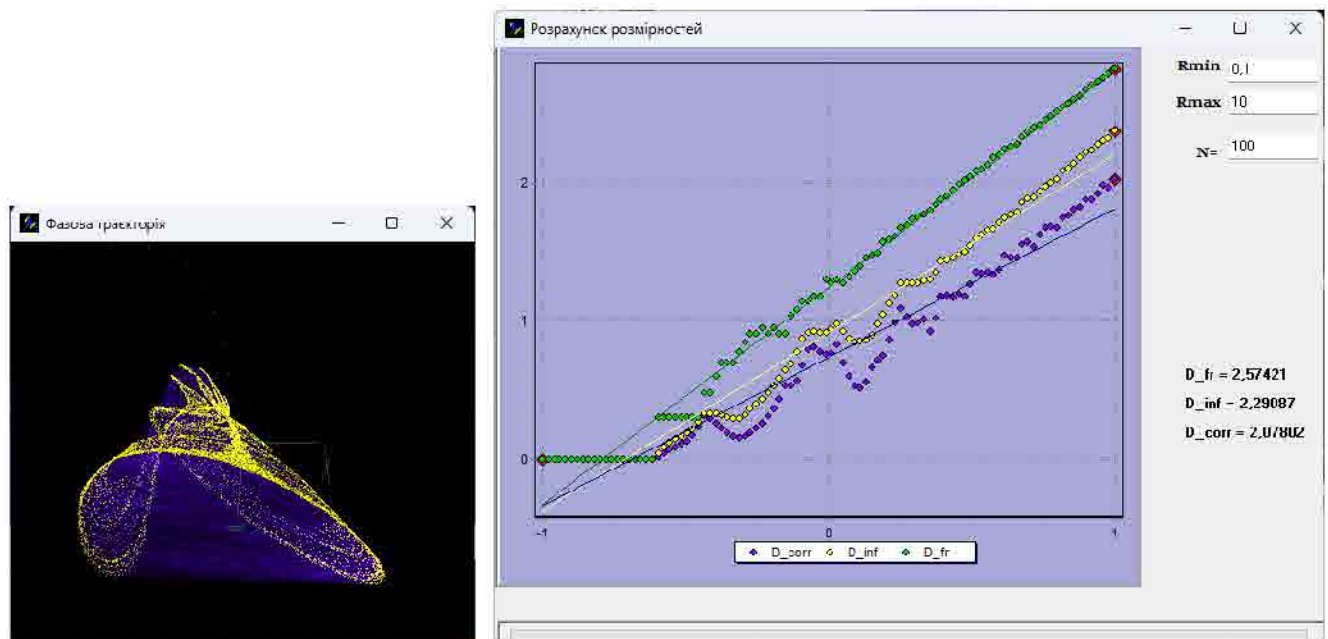


Рисунок 1.15. Інтерфейс додатку задля моделювання динамічної системи

Інтерфейс програмного додатку розроблений у стилі SDI (Single document interface). В ньому передбачені елементи, це дозволяють вводити вихідні

параметри динамічної системи, параметри його відтворення, параметри обчислення властивостей динамічної системи, елементи керування процесом моделювання та обчислення, але разом з тим елементи задля відтворення результатів обчислення та візуалізації. Відмінності у інтерфейсі задля тривимірної та чотиривимірний моделі пов'язані із різною кількістю полів ввід задля завдання властивостей динамічної системи та наявністю можливості відображати та налаштовувати перетин Poincare у чотиривимірний моделі.

У інтерфейсі задля тривимірної моделі передбачено діалогове віконце Calculate Fractal ввід вихідних даних моделювання, два діалогових віконця графічного виведення даних (Math задля відтворення сліду та Segment задля кольорового маркування порядку часток) та діалогове віконце текстового виведення результатів обчислення характеристик динамічної системи. В діалоговому віконці Math виводиться зображення, утворене точками, відстань між якими масштабується поза поміччю скролінга маніпулятора. Обчислення часток проводиться поза рівнянням (1.4) із коефіцієнтами та початковими значеннями динамічних змінних, котрі вводяться в відповідні поля ввід (рис. 1.15). Обчислення нового режиму та його відображення в віконці Math проводиться поза поміччю клацання на кнопці Calculate у разі, якщо активований прапор New calculation. Число часток (розмір) задається у полі Fractal Size. При відображенні динамічної системи у цьому віконці передбачена спроможність зміни масштабу зображення не тільки поза поміччю маніпулятора, але та поза поміччю завдання доречного діляника відтворення напрямків у поле ввід Divide cords. Поля ввід Shift cords along x axis та Shift cords along y axis дозволяють задати зсув початку напрямків при відображенні динамічної системи. Задля автоматизації пошуку серій у межах заданого діапазону властивостей призначені поля ввід Auto parameter1, Auto_parameter2, Auto_parameter3, Auto_parameter4. При цьому вводяться позиції та два параметра, котрі будуть перебиратися у процесі пошуку щодо отримання комбінацій, при яких динамічна система існує. При активації прапора Lorentz attractor проводиться обчислення динамічної системи Лоренца зі зміною вмісту відповідних полів ввід. Прапор Change Color надає змінити колір виведення часток

поза поміччю стандартного діалогу вибору кольору в віконці Math. У поле ввід Delta_x вказується критерій, це характеризує нескінченно малу величину, задля якої формується новий стан, задля обчислення його властивостей поза Ляпуновим. Поля ввід x1, x2, y1, y2 задають межі сегменту, відображуваного в віконці Segment. За завдання нових кордонів сегмента необхідно натиснути кнопку Enter threshold задля їх активації. При активації прапора Several N перед натисканням кнопки Calculate реалізується процедура обчислення властивостей із подальшим виведенням проміжних та кінцевих результатів у діалогове віконце текстового виведення. Перед початком обчислення властивостей можна задавати параметри методу сітки поза поміччю полів ввід Number of points та Pallet size (число часток, це фіксуються у комірці сітки та розмір комірки сітки тому). Поза поміччю прапора One point у розрахунках властивостей примусово задається одна частка, це фіксується у комірках сітки.

Поле Dim calc points number надає задати мінімальну число часток на графіку визначення фрактальної розмірності. При активації прапора Ljapunov indices перед натисканням кнопки Calculate реалізується процедура обчислення властивостей Ляпунова із подальшим виведенням проміжних та кінцевих результатів в діалогове віконце текстового виведення.

Кнопки Open webd, Save webd дозволяють відкривати та зберігати сформовані множини множини у папці dbo додатку. Кнопка Compute from file ініціює обчислення властивостей із збереженого в файлі цілком. Кнопка Close призначена задля виходу із додатку.

Інтерфейс додатку в покроковому стані показаний на рис. 1.16. При цьому відображені 3 із 30 етапів еволюції динамічної системи із параметрами: $k_{xy} = 0,8$; $k_{yx} = 1,1$; $k_{yz} = 1,0$; $k_{zy} = 0,2$; $k_{out} = 0,48$; $p = 0,95$; $q = 0,25$; $r = 0,2$; $x = 0,01$; $y = 0,01$; $z = 0,01$; $x_{in} = 0,33$.

В покроковому стані у робочий об'єм діалогового віконця Math на екран виводяться порції по N часток, тут N – задається параметром Step Size в відповідному полі ввід віконця Calculate Fractal. При натиснутій кнопці Reproduce series of frames здійснюються послідовне виведення порцій часток динамічної

системи. Щодо закінчення поетапного виведення часток множини вони виділені синім кольором, за закінчення повного виведення усіх часток динамічної системи їх колір стає жовтим.

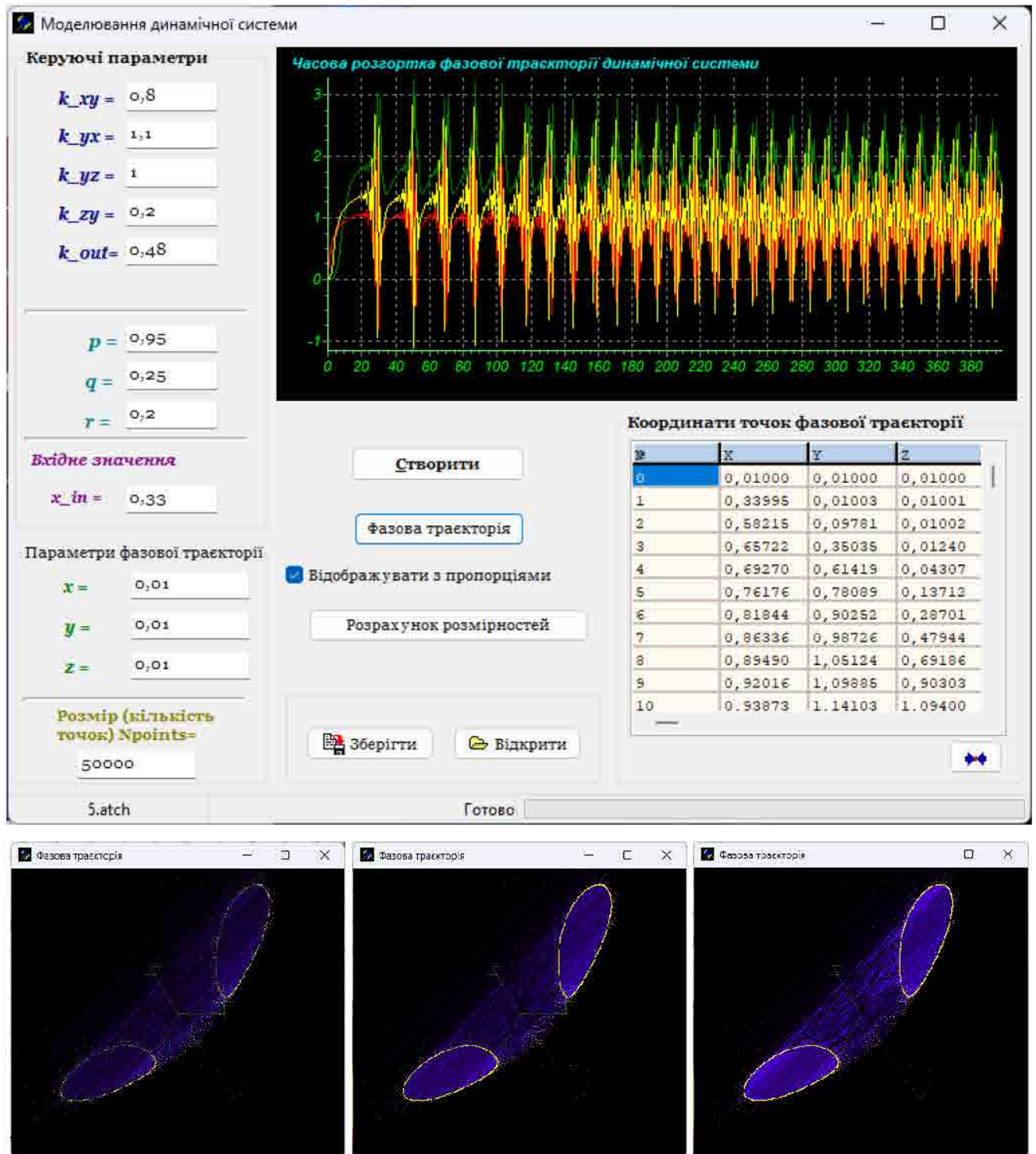


Рисунок 1.16. Інтерфейс додатку у покроковому стані та етапи еволюції динамічної системи задля 1000, 5000 та 50000 часток

На рис. 1.17 показаний інтерфейс задля чотиривимірний моделі у момент утворення січення Poincare по вісі W задля динамічної системи із параметрами: $k_{xy} = 0,1$; $k_{yx} = 0,1$; $k_{yz} = 0,1$; $k_{zy} = 0,1$; $k_{out} = 0,2$; $p = 1$; $q = 1$; $r = 1$; $x = 3,46$; $y = 6,99$; $z = -0,76$; $x_{in} = 1,556$; $N_{points} = 50000$. При цьому розраховані характеристики динамічної системи у 4-вимірному світі склали: $D_{fr} = 3,81$; $D_{corr} = 3,69$. Розраховані на наступному етапі роботи додатку характеристики січення Poincare по вісі W задля даного режиму склали: $D_{fr} = 2,57$; $D_{inf} = 2,27$; $D_{corr} = 2,15$.

У інтерфейсі задля чотиривимірний моделі передбачено діалогове віконце Calculate Fractal ввід вихідних даних моделювання, діалогове віконце графічного виведення даних Fourth dimensional math задля відтворення січення та діалогове віконце текстового виведення результатів обчислення характеристик динамічної системи чи січення Poincare.

В діалоговому віконці Fourth dimensional math виводиться зображення січення Poincare динамічної системи, утвореного точками, відстань між якими масштабується поза поміттю скролінгу маніпулятора. Обчислення часток у 4-вимірному світі проводиться поза рівнянням (1.5) із коефіцієнтами та початковими значеннями динамічних змінних, котрі вводяться в відповідні поля вводу (рис. 1.17). Обчислення нового режиму динамічної системи та відображення січення Poincare в віконці Fourth dimensional math проводиться поза поміттю клацання на кнопці Calculate у разі, якщо активовані прапор та New calculation, 4 dim attractor та Puancare.

При активації прапора Several N перед натисканням кнопки Calculate реалізується процедура обчислення властивостей динамічної системи у 4-вимірному світі із подальшим виведенням проміжних та кінцевих результатів у діалогове віконце текстового виведення.

При активації додатково прапора Dim calc for Puancare slice реалізується процедура обчислення властивостей січення Poincare у тривимірному світі поза обраною у полі Axis віссю (від 1, 2, 3, 4 задля осей напрямків X, Y, Z, W тому) із подальшим виведенням проміжних та кінцевих результатів у діалогове віконце текстового виведення (рис. 1.17).

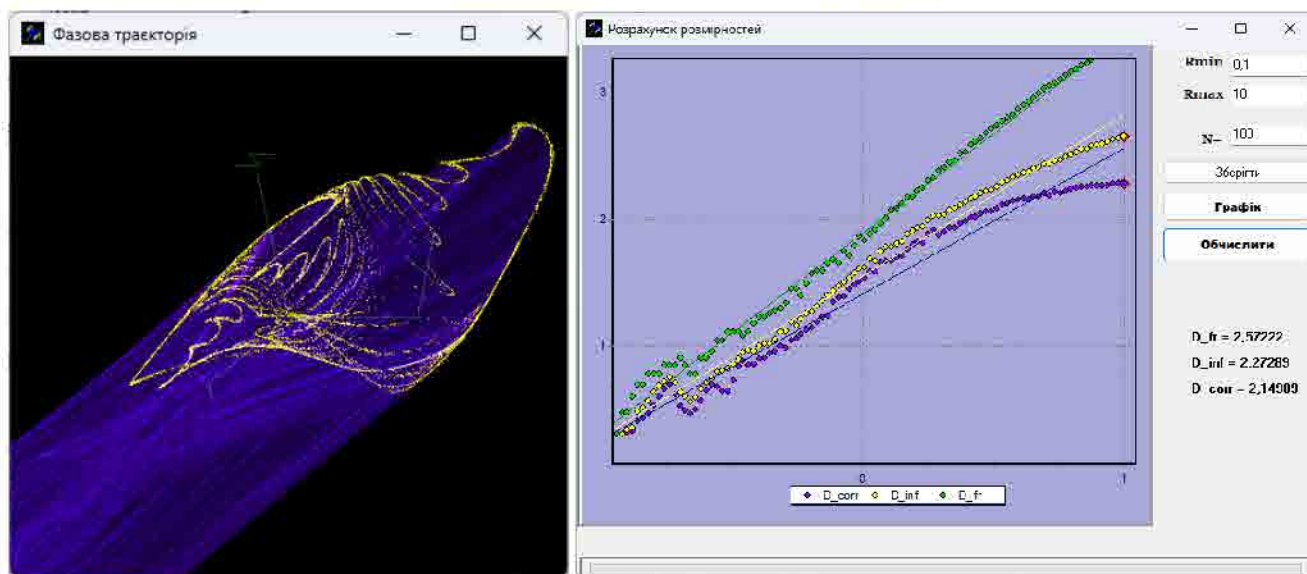
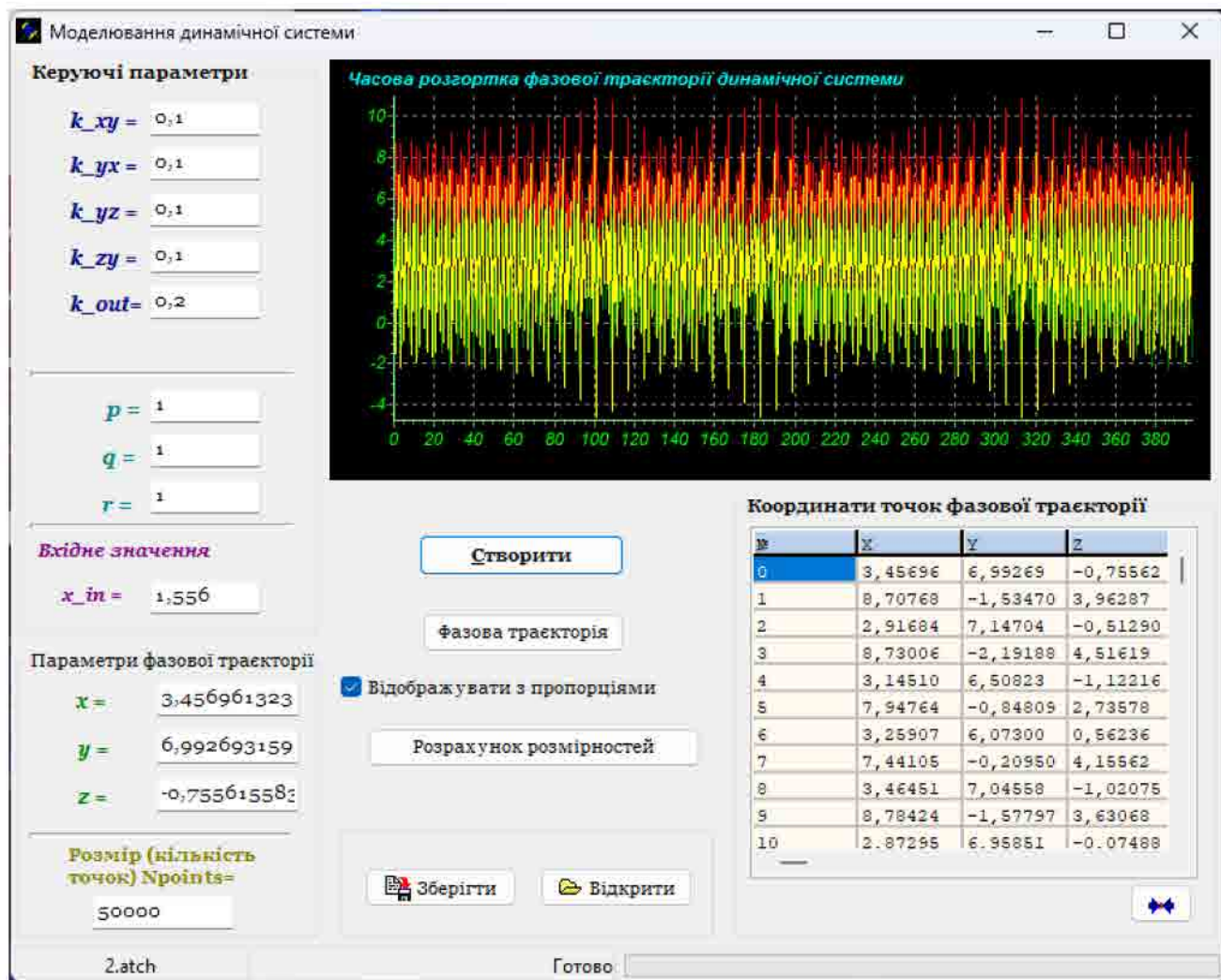


Рисунок 1.17. Інтерфейс додатку задля чотиривимірний моделі у процесі утворення січення по вісі W

Сканування усіх перетинів Poincare задля поточної динамічної системи (із даними коефіцієнтами виразів та даною кількістю часток) можливо поза поміччю натискання на кнопку Reproduce series of frames із включеним прапором Retrieve

3d attractor із зазначенням кількості кроків у полі ввід Row_number, кожен із яких, крім першого, виконується за зміни параметра Poincare (число часток у перетині) на 0,04 задля поточної вісі, починаючи із мінімального число задля поточної динамічної системи.

Параметр Row_number може набувати чисел від 50 щодо 5000. Якщо у поле ввід Row_number введене число, менше 50 чи більше 5000, то застосовується стандартне число 100.

Число параметра Poincare заноситься у файл rua_row.txt, котрий знаходиться у папці dbo додатку. Аналізуючи вміст цього файлу поза координатою задля поточної вісі та числом часток в відповідному перетині Poincare можна відібрати найбільш характерні та об'ємні зрізи. В файл rua_1.txt, котрий знаходиться у папці dbo додатку, записуються позиції часток січення Poincare кожен раз, коли змінюється число позиції Z в відповідному полі ввід при включеному прапорі Puancare. У полі ввід Slice parameter задається число від 0.01 щодо 0.05, відповідне точності січення Poincare, тобто товщині зрізу.

Задля контролю поза процесом моделювання інформація про позиції часток динамічної системи, але разом з тим проміжні дані обчислення характеристик динамічної системи (властивостей) виводяться в файли send.txt, s_pua.txt, fractal_web_d1.txt та fractal_web_d2.txt, котрі знаходяться в папці dbo додатку.

1.7 Результати моделювання динамічної системи

В даному підрозділі наведені результати моделювання (реконструкції) динамічних систем в 4-вимірному світі із візуалізацією їх тривимірних перетинів Poincare по вісі w. Задля кожного с експериментів наведено початкові параметри моделювання – число динамічних змінних та розподільних коефіцієнтів рекурентного рівняння (1.5) і число часток в множині (N_{points}), але разом з тим розраховані і виведені програмним застосунком результати – число фрактальної розмірності (D_{fr}).

					БКС 28. 28 000. 00 КРБ ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

1.7.1 Хаотичний стан динамічної системи

1. Хаотичний стан, це виникає при параметрах:

$$k_{xy} = 0,5;$$

$$k_{yx} = 0,4;$$

$$k_{yz} = 0,3;$$

$$k_{zy} = 0,3;$$

$$k_{zw} = 0,000001;$$

$$k_{wz} = 0,1;$$

$$p = 0,001;$$

$$q = 0,09;$$

$$r = 0,04;$$

$$s = 0,1;$$

$$x_{in} = 6;$$

$$k_{out} = 0,4;$$

$$x = 0;$$

$$y = 0;$$

$$z = 0;$$

$$w = 0;$$

$$N_{points} = 5000.$$

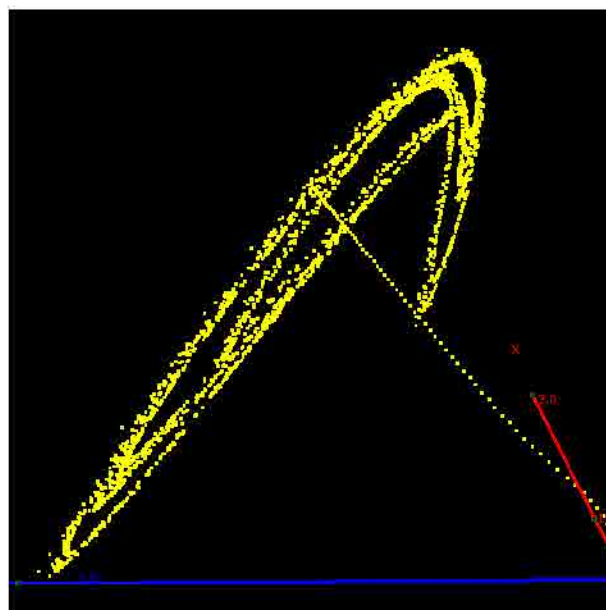
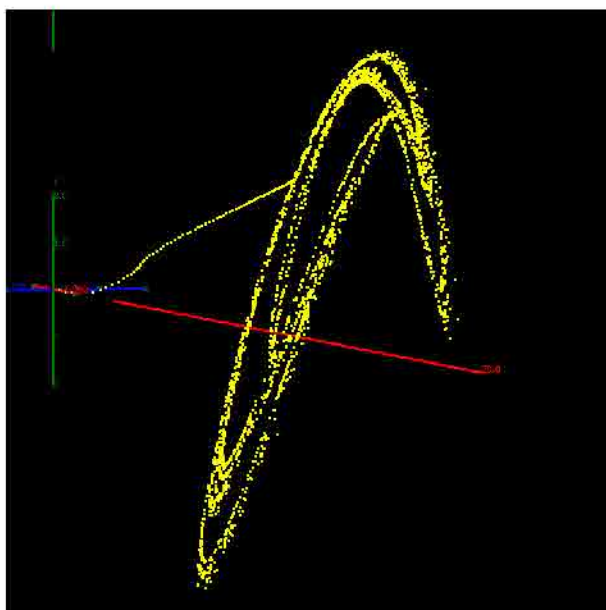


Рисунок 1.18. Динамічна система в хаотичному стані 1

Відображення 3-вимірного січення Poincare наведена на рис.1.18. Fractal властивість $D_{fr} = 3.29$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,18$.

2. Хаотичний стан, це виникає при параметрах:

$$k_{xy} = 0,1;$$

$$k_{yx} = 0,1;$$

$$k_{yz} = 0,1;$$

$$k_{zy} = 0,1;$$

$$k_{zw} = 0,00001;$$

$$k_{wz} = 0,1;$$

$$p = 1;$$

$$q = 1;$$

$$r = 1;$$

$$s = 0,1;$$

$$x_{in} = 1.536;$$

$$k_{out} = 0,2;$$

$$x = 3;$$

$$y = 7,9;$$

$$z = -2,13;$$

$$w = 0,01;$$

$$N_{points} = 5000.$$

Відображення 3-вимірного січення Poincare наведена на рис.1.19. Fractal властивість $D_{fr} = 3.2$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,08$.

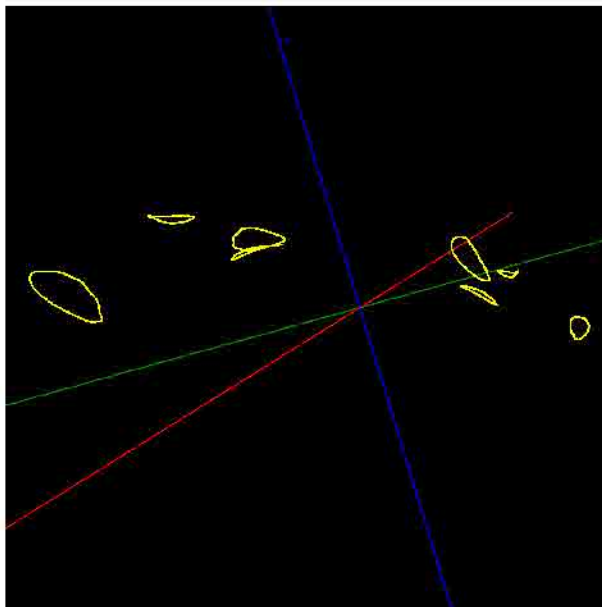


Рисунок 1.19. Динамічна система в хаотичному стані 2

3. Хаотичний стан, це виникає при параметрах:

$$k_{xy} = 0,65;$$

$$k_{yx} = 0,25;$$

$$k_{yz} = 0,65;$$

$$k_{zy} = 0,25;$$

$$k_{zw} = 0,000001;$$

$$k_{wz} = 0,1;$$

$$p = 1;$$

$$q = 1;$$

$$r = 1;$$

$$s = 0,1;$$

$$x_{in} = 0,435;$$

$$k_{out} = 0,4;$$

$$x = 0,927;$$

$$y = 0,49;$$

$$z = -0,13;$$

$$w = 0,01;$$

$$N_{points} = 5000.$$

Відображення 3-вимірному січення Poincare наведено на рис.1.20. Fractal властивість $D_{fr} = 3.58$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,56$.

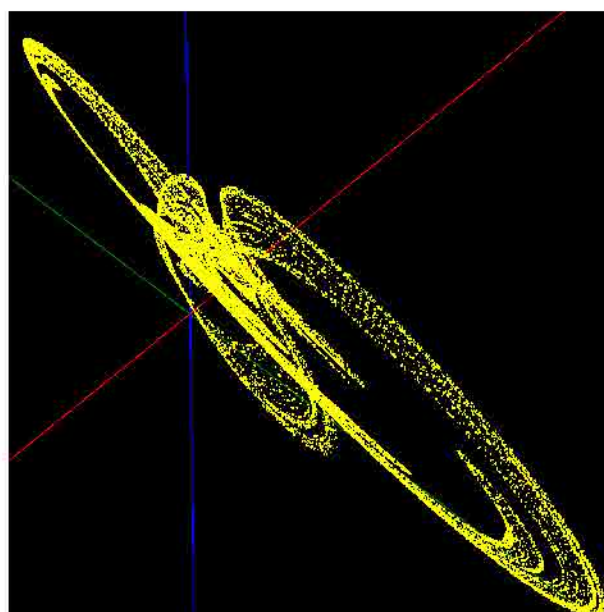
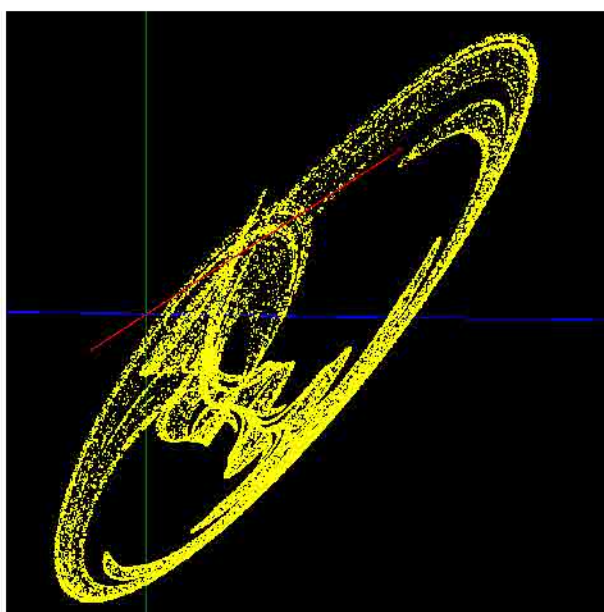


Рисунок 1.20. Динамічна система в хаотичному стані 3

4. Хаотичний стан, це виникає при параметрах:

$k_{xy} = 0,5;$	$p = 0.01;$	$x = 17,93;$
$k_{yx} = 0,4;$	$q = 1;$	$y = 1,46;$
$k_{yz} = 0,3;$	$r = 1;$	$z = -0,74;$
$k_{zy} = 0,3;$	$s = 1;$	$w = 0,1;$
$k_{zw} = 0,001;$	$x_{in} = 0,475;$	$N_{points} = 5000.$
$k_{wz} = 1;$	$k_{out} = 0,4;$	

Відображення 3-вимірного січення Poincare наведена на рис.1.21. Fractal властивість $D_{fr} = 3.66$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,64$.

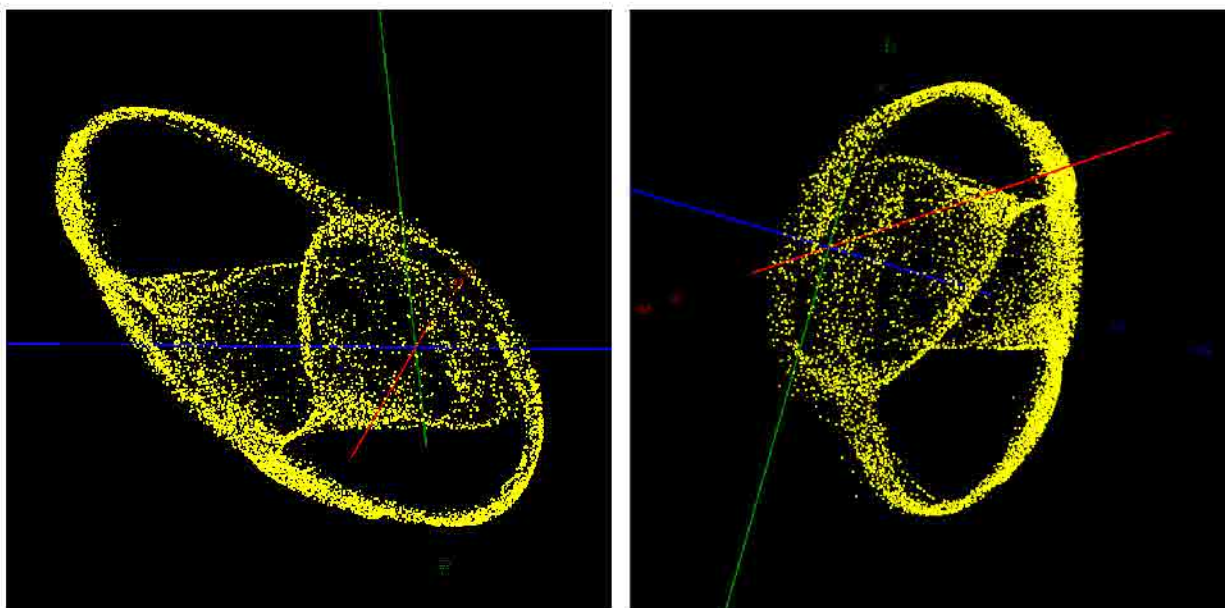


Рисунок 1.21. Динамічна система в хаотичному стані 4

1.7.2 Квазіперіодичний стан динамічної системи

1. Квазіперіодичний стан, це виникає при параметрах:

$k_{xy} = 0,5;$	$p = 0,065;$	$x = 21,67;$
$k_{yx} = 0,4;$	$q = 0,03;$	$y = 15,53;$
$k_{yz} = 0,3;$	$r = 0,03;$	$z = 15,64;$
$k_{zy} = 0,3;$	$s = 0,1;$	$w = 0,01;$
$k_{zw} = 0,00001;$	$x_{in} = 6.05;$	$N_{points} = 5000.$
$k_{wz} = 0,1;$	$k_{out} = 0,4;$	

Відображення 3-вимірного січення Poincare наведена на рис.1.22. Fractal властивість $D_{fr} = 3.14$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,08$.

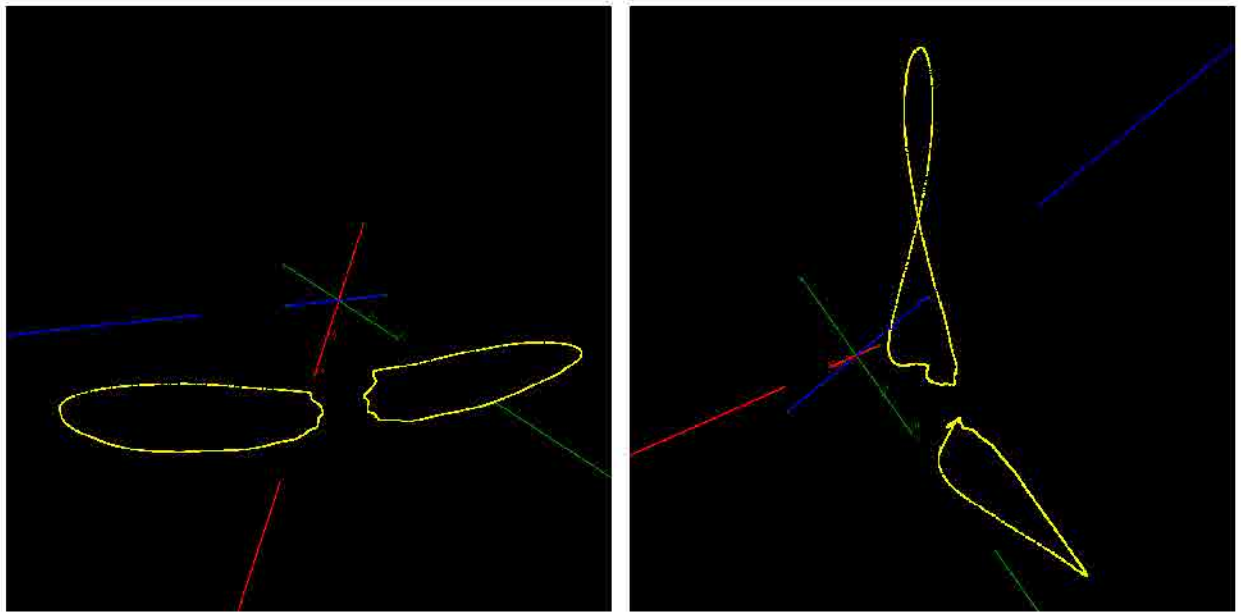


Рисунок 1.22. Динамічна система в квазіперіодичному стані 1

2. Квазіперіодичний стан, це виникає при параметрах:

$$k_{xy} = 0,5;$$

$$p = 0,1;$$

$$x = 22;$$

$$k_{yx} = 0,4;$$

$$q = 0,1;$$

$$y = 10;$$

$$k_{yz} = 0,3;$$

$$r = 0,001;$$

$$z = -91;$$

$$k_{zy} = 0,3;$$

$$s = 0,1;$$

$$w = 0,01;$$

$$k_{zw} = 0,00001;$$

$$x_{in} = 2,6;$$

$$N_{points} = 5000.$$

$$k_{wz} = 0,1;$$

$$k_{out} = 0,4;$$

Відображення 3-вимірного січення Poincare наведена на рис.1.23. Fractal властивість $D_{fr} = 3.21$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,07$.

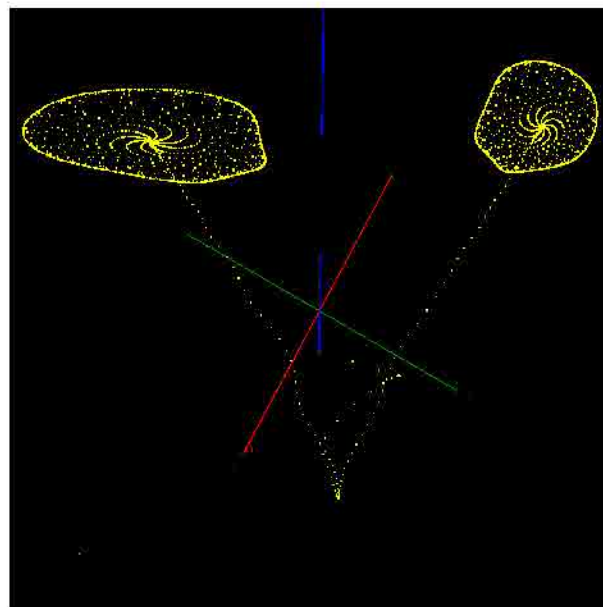


Рисунок 1.23. Динамічна система в квазіперіодичному стані 2

3. Квазіперіодичний стан, це виникає при параметрах:

$k_{xy} = 0,5;$	$p = 0,05;$	$x = 30;$
$k_{yx} = 0,4;$	$q = 0,065;$	$y = 10;$
$k_{yz} = 0,3;$	$r = 0,0075;$	$z = 10;$
$k_{zy} = 0,3;$	$s = 0,1;$	$w = 0,1;$
$k_{zw} = 0,000001;$	$x_{in} = 5,9;$	$N_{points} = 5000.$
$k_{wz} = 0,1;$	$k_{out} = 0,4;$	

Відображення 3-вимірного січення Poincare наведена на рис.1.24. Fractal властивість $D_{fr} = 3.18$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,16$.

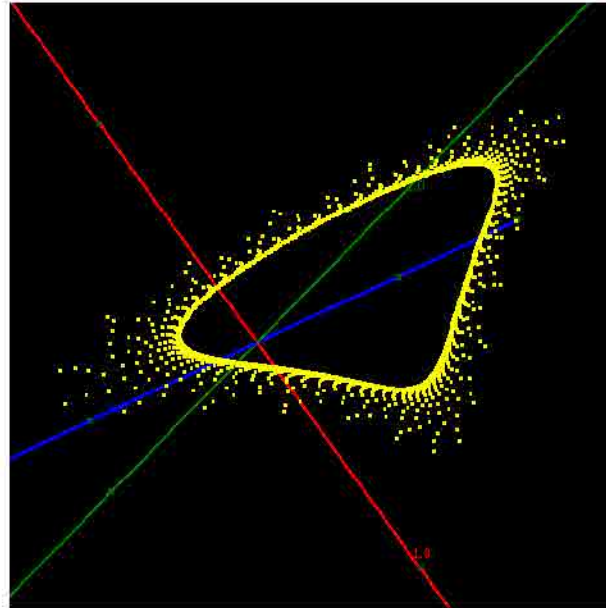


Рисунок 1.24. Динамічна система в квазіперіодичному стані 3

1.7.3 Дивний стан в динамічній системі

1. Дивний стан, це виникає при параметрах:

$k_{xy} = 0,8;$	$p = 0,95;$	$x = 0,025;$
$k_{yx} = 1,1;$	$q = 0,25;$	$y = 0,025;$
$k_{yz} = 1,0;$	$r = 0,2;$	$z = 0,025;$
$k_{zy} = 0,2;$	$s = 0,1;$	$w = 0,025;$
$k_{zw} = 0,0001;$	$x_{in} = 0,33;$	$N_{points} = 5000.$
$k_{wz} = 0,1;$	$k_{out} = 0,48;$	

Відображення 3-вимірного січення Poincare наведена на рис.1.25. Fractal властивість $D_{fr} = 3.24$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,13$.

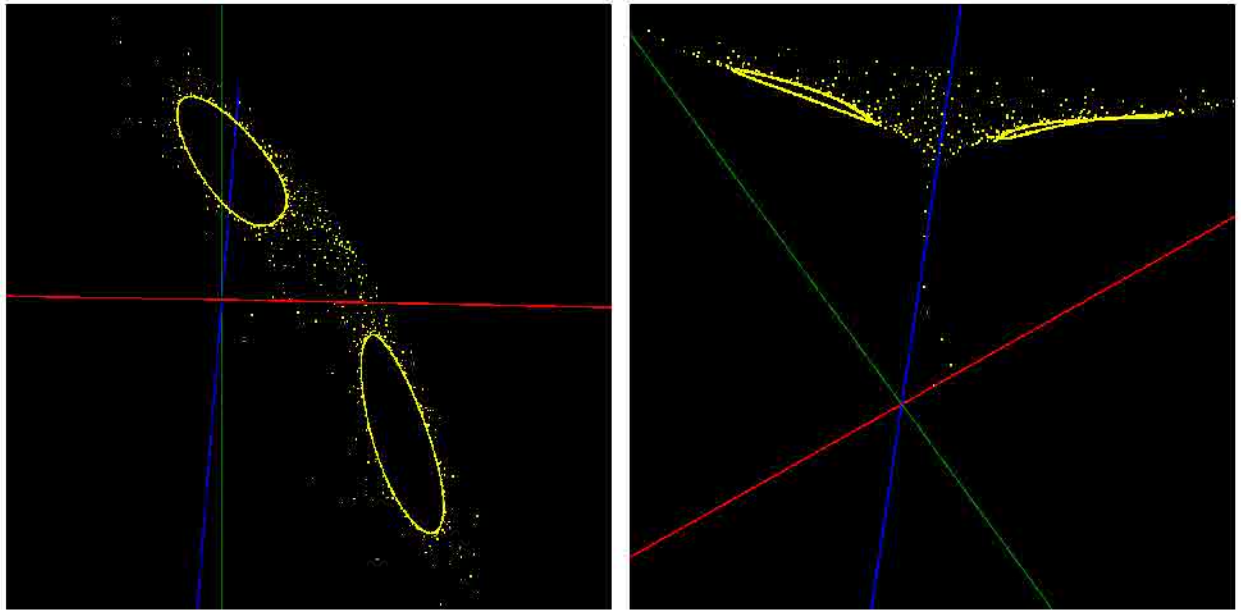


Рисунок 1.25. Дивний стан 1 динамічної системи

2. Дивний стан, це виникає при параметрах:

$k_{xy} = 0,45;$	$p = 0,4;$	$x = 0,01;$
$k_{yx} = 0,5;$	$q = 0,2;$	$y = 0,01;$
$k_{yz} = 0,3;$	$r = 0,55;$	$z = 0,01;$
$k_{zy} = 0,4;$	$s = 0,1;$	$w = 0,01;$
$k_{zw} = 0,0001;$	$x_{in} = 0,55;$	$N_{points} = 5000.$
$k_{wz} = 0,1;$	$k_{out} = 0,45;$	

Відображення 3-вимірного січення Poincare наведена на рис.1.26. Fractal властивість $D_{fr} = 3.51$, fractal властивість січення Poincare по вісі w $D_{fr} = 2,41$.

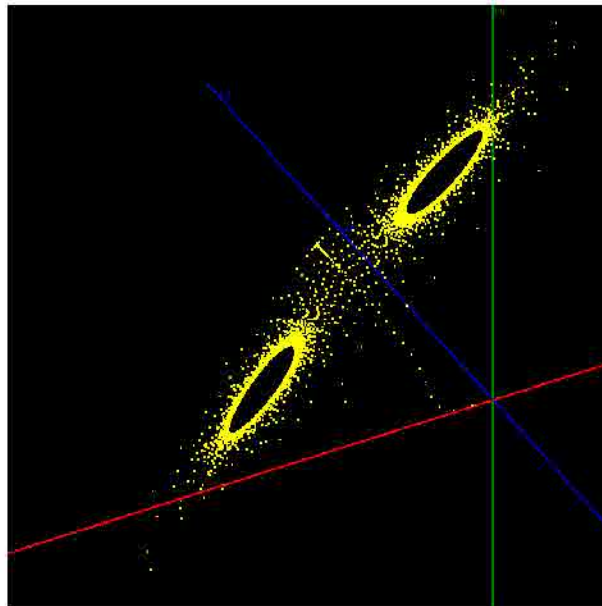


Рисунок 1.26. Дивний стан 2 динамічної системи

Залежно від величин перехідних змінних та розподільних коефіцієнтів

змінюються сценарії еволюції динамічної системи, описаної рівняннями (1.5).

Відображення доречного типу динамічної системи свідчить про наявність детермінованого хаосу в динамічній системі (рис.1.18-1.21). Динамічні системи на рис. 1.22-1.24 дають змогу говорити про відсутність хаосу і про періодичність (квазіперіодичність). Стан динамічної системи, показаний на рис.1.25-1.26, веде щодо утворення дивного режиму, це представляє собою притягуючу безліч нестійких траєкторій в фазовому світі – вони описують стохастичні автоколивання, підтримувані у динамічній системі поза рахунок зовнішнього джерела. В системах разом з тим зустрічаються та інші сценарії розвитку хаосу, наприклад перехід щодо хаосу через тор, при якому за першої біфуркації із частки народжується двочастотної тор, у загальному випадку із ірраціональним співвідношенням частот. Але при варіюванні керуючого параметра виникають синхронізації частот різного порядку, цьому відповідають динамічні системи в виді послідовності декількох часток, число яких дорівнює ступеню синхронізації. Зі збільшенням величини вхідного потоку тор руйнується та народжується хаотичний стан динамічної системи.

2 РОЗДІЛ ОХОРОНИ ПРАЦІ І ТЕХНІКИ БЕЗПЕКИ

Сучасного розвиток суспільства має характерну особливість. Це зростання сфер діяльності людини, у яких використовуються інформаційні технології.

Широке розповсюдження отримали персональні комп'ютери. Однак їх використання загострило проблеми збереження власного і суспільного здоров'я, вимагає удосконалення існуючих і розробки нових підходів щодо організації робочих місць, проведення профілактичних заходів задля запобігання розвитку негативних наслідків впливу ПК на здоров'я користувачів.

Забезпечення безпечних та здорових умов праці у значній мірі залежить від правильної оцінки небезпечних і шкідливих виробничих факторів. Однакові по складності зміни у організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне та розумове навантаження, нервово-емоційна напруга, але разом з тим різне сполучення цих причин.

У дипломному розділі дипломного проекту розглядається питання охорони праці програміста на стадії вирішення ним питань розробки алгоритмічного і програмного забезпечення задля моделювання і візуалізації динамічної системи.

2.1 Аналіз небезпечних і шкідливих чинників, це впливають на працівника

Оператори ПК та програмісти зіштовхуються із впливом таких фізично небезпечних та шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, недостатня освітленість робочої зони, електричний струм і інші. Тому на робочому місці програміста повинні бути створені умови задля високопродуктивної праці.

Щодо засобів захисту відносять: вентиляцію, штучне освітлення, звукоізоляцію. В системі мір, це забезпечують сприятливі умови праці, велике місце приділяється естетичним факторам: оформлення виробничого інтер'єра, устаткування, застосування функціональної музики і інше, котрі впливають на організм людини.

2.2 Розробка заходів із охорони праці

2.2.1 Виробничі приміщення

Розміщення робочих місць із ВДТ ЕОМ та ПЕОМ в підвальних приміщеннях і на цокольних поверхах заборонено.

Площа на одне робоче місце задля операторів повинна складати не менше 6,0 м², але об'єм – не менше 20,0 м³. Стіни повинні бути пофарбовані матовою краскою.

2.2.2 Мікроклімат робочої зони працівників, вентиляція

В виробничих приміщеннях температура, відносна вологість та швидкість руху повітря на робочих місцях повинні відповідати санітарним нормам мікроклімату виробничих приміщень – ДсанПіН 3.3.2-007-98. Висока температура повітря у сполученні із високою вологістю повітря дуже впливають на працездатність оператора. Поза таких умов різко збільшується час сенсомоторних реакцій, порушується координація рухів, збільшується число помилок. Висока температура негативно позначається й на ряді психологічних процедур людини. Зменшується обсяг запам'ятовуємої інформації, різко знижується здатність щодо асоціацій, погіршується протікання асоціативних та рахункових операцій, знижується увага. Тому оптимальні параметри мікроклімату в виробничому приміщенні повинні становити:

- температура повітря – 22-25⁰С;
- відносна вологість – 40-60%;
- швидкість руху повітряних мас – 0,1-0,2 м/сек.

Задля підтримки необхідних властивостей мікроклімату робоче приміщення оснащене системами опалення й кондиціювання.

В приміщеннях, тут здійснюються робота програміста вимоги щодо властивостей мікроклімату у цілому виконанні.

2.2.3 Освітлення робочого місця, шум, вібрація

Приміщення задля роботи із ВДТ повинні мати природне і штучне освітлення, тому щодо ДБН У.2.5-28-2006. В приміщеннях, призначених задля роботи із відео терміналами, доцільно, щоб віконця були орієнтовані на північ чи

					БКС 28. 28 000. 00 КРБ ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

північний захід. На вікнах повинні бути штора чи жалюзі, це регулюють рівень освітленості та захищають від прямого влучення сонячних променів на робоче місце. При кольоровому оформленні виробничих та допоміжних приміщень необхідно враховувати орієнтацію їхніх вікон стосовно частин світу та використовувати гармонійне сполучення кольорів. Задля стін та робочих поверхонь використовують мало насичені (основні) кольори, задля невеликих помешкань чи ділянок, це рідко потрапляють в поле зору працюючих, але разом з тим задля створення контрастності – усіх приміщеннях повинні бути білими.

Поверхні устаткування у приміщеннях повинні бути матовими чи напівматовими, задля виключення кольори середньої насиченості (допоміжні), задля маленьких по площі поверхонь – насичені (акценти) – як функціональне фарбування. Стелі в випадку відблисків світла у очі працюючого, але стіни бути пофарбованими фарбами пастельних тонів.

Задля штучного освітлення в приміщенні використовуються люмінесцентні лампи типа ЛБ, котрі у порівнянні із лампами розжарювання мають ряд істотних переваг: поза спектральним складом світла вони близькі щодо природного світла, мають підвищену світлову віддачу (в 2-5 разів вищу, ніж в ламп розжарювання); мають триваліший термін служби – щодо 10 тис годин.. Допускається застосування ламп розжарювання в світильниках місцевого освітлення.

В робочому приміщенні основними джерелами акустичних шумів являється шуми ПЕОМ. Це коливання елементів електромеханічних пристроїв під впливом змінних магнітних полів тощо. Шум може викликати стомлення слуху й ослаблення звукового сприйняття, але разом з тим значне стомлення всього організму.

Оптимальні показники рівня шумів в робочих приміщеннях визначаються поза ГОСТ 12.1.003-83. Припустимий рівень шуму при розумовій праці, це вимагає зосередженості – 50 дБ.

2.2.4 Електробезпека

Задля попередження поразок електричним струмом необхідно чітко й в повному обсязі виконувати правила провадження робіт та правил технічної

					<i>БКС 28. 28 000. 00 КРБ ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

експлуатації. Необхідно виключити спроможність доступу оператора щодо частин устаткування, це працює під небезпечною напругою, щодо неізольованим частинам, призначеним задля роботи при малій напрузі й не підключеним щодо захисного заземлення, але разом з тим підводити електроживлення щодо ПЕОМ від розетки поза поміччю спеціальної вилки із заземлюючим контактом.

2.2.5 Організація робочого місця користувача ПК

Конструкція робочого місця й взаємне розташування усіх його елементів (сидіння, органи керування, засобу відтворення інформації) відповідають антропометричним, фізіологічним та психологічним вимогам, але разом з тим характеру роботи. Конструкція робочих меблів дає спроможність забезпечувати спроможність індивідуального регулювання їх тому щодо потреб працівника задля підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою.

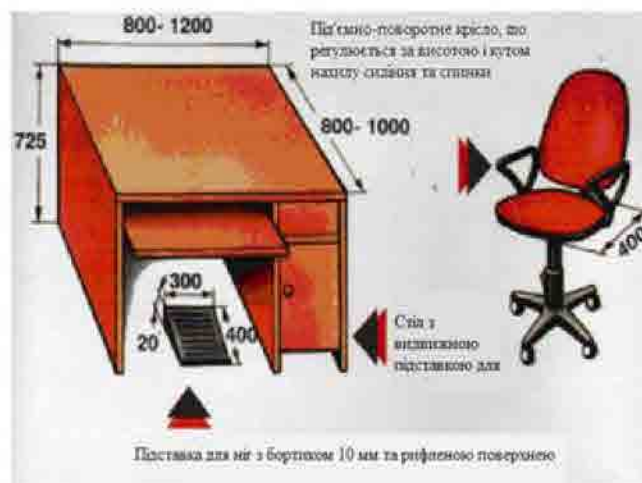


Рисунок 2.1. Організація робочого місця користувача ПК

розташований так, це його верхній край перебуває на рівні очей, на відстані близько 70 см, це укладається у припустимі рамки від 60 щодо 90 см. Частота мерехтіння екрана дорівнює 100 Гц, це дорівнює умові більше 70 Гц.

Обладнання та організація робочого місця із ВДТ мають забезпечувати відповідність конструкцій усіх елементів робочого місця і їх взаємного розташування, ергономічним вимогам, із урахуванням характеру та особливостей трудової діяльності (ДСанПіН 3.3.2.-007-98).

2.3 Пожежна безпека

Під пожежною безпекою розуміють систему державних та суспільних заходів, спрямованих на охорону від вогню людей та матеріальних цінностей

Заходи щодо пожежної безпеки підрозділяють на дві основні групи: попередження пожеж та ліквідація вже виниклих пожеж. Пожежна профілактика – це комплекс заходів, спрямованих на попередження пожежі, створення умов, сприяючих швидкій ліквідації пожежі.

Протипожежний захист приміщення забезпечується застосуванням автоматичної установки пожежної сигналізації, наявністю засобів пожежогасіння, застосуванням основних будівельних конструкцій будинку із регламентованими межами вогнестійкості, організацією своєчасної евакуації людей.

Задля ліквідації пожеж використовують первинні засоби пожежогасіння, котрі призначенні задля гасіння пожеж в початковій стадії їх розвитку. Вони являється в усіх виробничих приміщеннях, цехах.

Щодо первинних засобів пожежогасіння відносяться : вогнегасники, пожежний інвентар (покривала із негорючого теплоізоляційного полотна грубововняної тканини чи повсті, ящики із піском, бочки із водою, пожежні відра, совкові лопати) і пожежний інструмент (гаки, лом, сокири тощо).

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

В випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефону 101 пожежну команду, евакуювати людей із приміщення тому щодо плану евакуації та приступити щодо ліквідації пожеж.

					БКС 28. 28 000. 00 КРБ ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

ВИСНОВКИ

Виконане вивчення режимів роботи динамічної системи засобами комп'ютерного моделювання дозволило довести спроможність керування сценаріями розвитку динамічної системи, це може використовуватися у різних галузях науки та техніки. Такими динамічними системами можуть бути атмосфера, турбулентні потоки, деякі види аритмій серця, біологічні популяції, суспільство як система комунікацій та його підсистеми: економічні, політичні, психологічні (культурно-історичні і інтер-культуральні) і інші соціальні системи.

Реалізоване алгоритмічне і програмне забезпечення задля моделювання і візуалізації динамічної системи надає досліджувати еволюцію абстрактних систем, котрі складаються із компонент, це взаємодіють поза довільними законами. Поза поміччю засобів комп'ютерної графіки забезпечено спроможність візуалізувати динамічні системи і визначати перетини Poincare, обчислювати розмірності, це спрощує їх класифікацію. Реалізоване програмне забезпечення створене мовою програмування C++, має зручний візуальний інтерфейс і оптимізовану внутрішню архітектуру, це надає використовувати його на будь-якому ПК із системними вимогами, це пред'являються щодо ОС Windows 10.

Розроблена задля моделювання і візуалізації динамічної системи комп'ютерна модель надає вивчати характер її поведінки, структуру та властивості її відбитків в трьох- та чотирьох-вимірному світі. Основні параметри, це дозволяють класифікувати динамічні системи по типу – їх розрахована fractal властивість, але разом з тим розташування по осях у фазовому світі, вигляд відбитку січення та щільність часток у різних областях їх візуалізації. Визначено три типи динамічних систем: хаотична, квазіперіодична та дивна. Залежно від величин перехідних змінних та розподільних коефіцієнтів змінюються сценарії еволюції динамічної системи, описаної рекурентними рівняннями запропонованої моделі. Тип сценарію знаходиться у однозначній відповідності із типом динамічної системи і її перетином.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Данилов В.Я. Синергетичні методи аналізу: навч. посіб. / В.Я. Данилов, А.Ю. Зінченко. — К.: НТУУ «КПІ» ВПІ ВПК «Політехніка», 2011. — 340 с.
2. Кветний Р.Н., Комп'ютерне моделювання систем та процесів. Методи обчислень: навчальний посібник – Вінниця: ВНТУ, 2012. – 193 с.
3. Хусаїнов Д.Я. Введення в моделювання динамічних систем: Навч. посібник – КНУ ім. Т. Шевченка, 2010.
4. Poincare H. Les Methodes Nouvelles de la Mecanique Celeste. Nabu Press, 2010. 398 p.
5. Volkov V. Transition of combustion to explosion and decision support systems for explosion protection / Lecture Notes in Computational Intelligence and Decision Making, Advances in Intelligent Systems and Computing, volume 1246, Springer, Cham. 2021, pp. 437-447. doi: 10.1007/978-3-030-54215-3.
6. Herega A., Kryvchenko Yu. Three-dimensional Poincare cross-sections in the model of oscillatory interaction of different-scaled structures in solids / Proc. of 3rd Int. Conf. «Topological methods in dynamics and related topics. Shilnikov workshop», 2019. – P. 26-27.
7. Кривченко Ю.В. Комп'ютерна реалізація атракторних систем у багатовимірних фазових просторах / Збірник доповідей XII міжнародної науково-практичної конференції «Інформаційні технології та автоматизація». – 2019. – Т. 1. – С. 49-50.
8. Кривченко Ю.В. Моделювання поведінки динамічних систем в процесі механоактивації / Матеріали XIV міжнародної науково-практичної конференції «Інформаційні технології і автоматизація». – 2021. – С. 34-36
9. Трофименко О.Г. C++. Алгоритмізація та програмування: підручник / 2-ге вид. перероб. і доповн. – Одеса : Фенікс, 2019. – 477 с.
10. Stroustrup B. A Tour of C++ (Second Edition). – Addison-Wesley, 2018. – 240 p. – ISBN 978-0-13-499783-4.
11. Бібліотека MSDN [Електронний ресурс]. – Режим доступу: URL <http://msdn.microsoft.com/ru-ru/library/default.aspx>.

					БКС 28. 28 000. 00 КРБ ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

Фрагмент коду функції **Dimension** класу **Fractal** мовою **C++**

```

void Fractal::Dimension(double ps1, int fr_s1, Fractal *pf) {
    double N_c;
    string name, name_in, name2;
    FILE * file_dim, *f_name;
    N_c = (ps1*ps1*ps1) / (x / 2.0);
    int i, j, k, l;
    int fl_print = 0;
    if(fl_print == 1)
        MessageBoxW(NULL, L"Dimension! Preliminary computation.", L"Info!", MB_OK);
    char b112[150];
    char b150[150];
    sprintf(b150, "pf->xmax = %lf", pf->xmax);
    QMessageBox msgBox8;
    msgBox8.setWindowTitle(pf->xmax);
    msgBox8.setText(b150);
    if (fl_print == 1) {
        if (msgBox8.exec() == QMessageBox::Yes) {;}
        sprintf(b150, "pf->xmin = %lf", pf->xmin);
        msgBox8.setWindowTitle(pf->xmin);
        msgBox8.setText(b150);
        if (fl_print == 1) {
            if (msgBox8.exec() == QMessageBox::Yes) {;}
            sprintf(b150, "pf->ymin = %lf", pf->ymin);
            msgBox8.setWindowTitle(pf->ymin);
            msgBox8.setText(b150);
            if (fl_print == 1) {
                if (msgBox8.exec() == QMessageBox::Yes) {;}
                sprintf(b150, "pf->zmin = %lf", pf->zmin);
                msgBox8.setWindowTitle(pf->zmin);
                msgBox8.setText(b150);
                if (fl_print == 1) {
                    if (msgBox8.exec() == QMessageBox::Yes) {;}
                    sprintf(b150, "Nn = %lf", pf->Nn);
                    msgBox8.setWindowTitle(pf->Nn);
                    msgBox8.setText(b150);
                    if (fl_print == 1) {
                        if (msgBox8.exec() == QMessageBox::Yes) {;}
                        sprintf(b150, "pf->xp_n.size() = %d", (pf->xp_n).size());
                        msgBox8.setWindowTitle(pf->xp_n.size());
                        msgBox8.setText(b150);
                        if (fl_print == 1) {
                            if (msgBox8.exec() == QMessageBox::Yes) {;}
                            sprintf(b150, "pf->yp_n.size() = %d", (pf->yp_n).size());
                            msgBox8.setWindowTitle(pf->yp_n.size());
                            msgBox8.setText(b150);

```

```

if (fl_print == 1) {
if (msgBox8.exec() == QMessageBox::Yes) //&& (fl_print == 1)) {;}
sprintf(b150, "pf->zp_n.size() = %d ", (pf->zp_n).size());
msgBox8.setWindowTitle("pf->zp_n.size()");
msgBox8.setText(b150);
if (fl_print == 1) {
if ((msgBox8.exec() == QMessageBox::Yes)) {;}
char b125[150];
for (j = 0; j < pf->xp_n.size(); j++)
for (k = 0; k < pf->yp_n.size(); k++)
for (l = 0; l < pf->zp_n.size(); l++) {
sprintf(b112, "%d, %d, %d", j, k, l);
name2 = static_cast<string>(b125);
pf->pointsinPallets_in[name] = 0;
pf->pointsinPallets[name] = 0; }
int m;
if ((fr_s1 > 190) && (fr_s1 <= 1500)) {
if (fopen_s(&file_dim, "c:\\db0\\dim.txt", "r+") == NULL) {}
else {MessageBoxW(NULL, L"The error while attempting to open the file dim.txt!",
L"Info!", MB_OK);}
if (fl_print == 1) MessageBoxW
(NULL, L"End of preliminary calculation of the fractal dimension!", L"Info!", MB_OK);
if (file_dim) fclose(file_dim); }
sprintf(b150, "pf->xmax = %lf ", pf->xmax);
msgBox8.setWindowTitle("pf->xmax");
msgBox8.setText(b150);
sprintf(b150, "pf->xn.size() = %d ", pf->xn.size());
msgBox8.setWindowTitle("pf->xn.size()");
msgBox8.setText(b150);
if (fl_print == 1) {
if (msgBox8.exec() == QMessageBox::Yes) {;}
if (fl_print == 1)
MessageBoxW(NULL, L"Dimension! The end of the first phase!", L"Info!", MB_OK);
int inf_dim_i;
inf_dim_i = 1;
Npmax = 0;
if (inf_dim_i == 0) {;}
for (j = 0; j < pf->xp_n2.size(); j++)
for (k = 0; k < pf->yp_n2.size(); k++)
for (l = 0; l < pf->zp_n2.size(); l++) {
sprintf(b112, "%d, %d, %d", j, k, l);
name = static_cast<string>(b112);
pf->pointsinPallets_in[name] = 0;
pf->pointsinPallets[name] = 0;
pf->pinPa_calc[name] = 0;
sprintf(b125, "%d ", j);
name2 = static_cast<string>(b125);
pf->Attr_val_calc[name2] = 0;
sprintf(b112, "%d, %d, %d", -j, k, l); //; %d", j, k, l, i);
name = static_cast<string>(b112);
pf->pointsinPallets_in[name] = 0;

```

```

pf->pointsinPallets[name] = 0;
pf->pinPa_calc[name] = 0;
sprintf(b112, "%d, %d, %d", j, -k, l); //; %d", j, k, l, i);
name = static_cast<string>(b112);
pf->pointsinPallets_in[name] = 0;
pf->pointsinPallets[name] = 0;
pf->pinPa_calc[name] = 0;
sprintf(b112, "%d, %d, %d", j, k, -l); //; %d", j, k, l, i);
name = static_cast<string>(b112);
pf->pointsinPallets_in[name] = 0;
pf->pointsinPallets[name] = 0;
pf->pinPa_calc[name] = 0;}
int kyn, lzn, jxn;
kyn = lzn = jxn = 0;
l = 0; j = 0; k = 0;
// Minus Values of coords
it_xp_ns = pf->xp_n.begin();
it_yp_ns = pf->yp_n2.begin();
it_zp_ns2 = pf->zp_n.begin();
it_xp_n = pf->xp_n.begin();
it_yp_n = pf->yp_n.begin();
it_zp_n = pf->zp_n.begin();
int tt;
for (i = 0, it_xn = pf->xn.begin(), it_yn = pf->yyn.begin(), it_zn = pf->zn.begin();
i < pf->xn.size(), it_xn != pf->xn.end(), it_yn != yyn.end(), it_zn != zn.end();
i++, it_xn++, it_yn++, it_zn++) {
if ((*it_xn > 0.0) && (*it_yn > 0.0) && (*it_zn > 0.0)) continue;
for (j = 0, it_xp_ns = pf->xp_n.begin(); j < (pf->xp_n.size()), it_xp_ns != pf->xp_n.end();
j++, it_xp_ns++) {
for (k = 0, it_yp_ns = pf->yp_n2.begin(); k < (pf->yp_n2.size()), it_yp_ns != yp_n2.end();
k++, it_yp_ns++) {
for (l = 0, it_zp_ns2 = pf->zp_n.begin(); l < (pf->zp_n.size()), it_zp_ns2 != zp_n.end();
l++, it_zp_ns2++) {
sprintf(b150, "i = %d, m = %d, j = %d, k = %d, l = %d, *it_xp_ns = %lf,
*it_yp_ns = %lf, *it_xp_n = %lf\n", i, m, j, k, l, *it_xp_ns, *it_yp_ns, *it_xp_n);
msgBox8.setWindowTitle("Data");
msgBox8.setText(b150);
if ((*it_xn > ps1*(j - 1)) && (*it_xn < ps1*(j)) && (*it_yn > -ps1*k) && (*it_yn
< -ps1*(k - 1)) && (*it_zn > ps1*(l - 1)) && (*it_zn < ps1*l)) { // || (k == 0) || (l == 0)) &&
((( *it_yn < 0.0) )) && (*it_xn > 0.0) && (*it_zn > 0.0) && ((*it_xn < *(it_xp_ns)) || (*it_zn
< *(it_zp_ns2)) || (*it_yn > *it_yp_ns))) { // && (*it_xn < ps1) && (*it_zn < ps1))) { // (*it_xn
< (*it_xp_n)) && (*it_xn > (*it_xp_n - 1))) && (*it_yn < (*it_yp_n)) && (*it_yn > (*it_yp_n
- 1))) && (*it_zn > (*it_zp_ns2)) && (*it_zn < (*it_zp_ns2 - 1))) {
if (((l > 0) && (j == 0) && (k > 0)) || ((k > 0) && (j > 0) && (l == 0)) || ((j > 0) && (k == 0)
&& (l > 0)) || ((j == 0) && (k == 0) && (l > 0)) || ((j == 0) && (k > 0) && (l == 0)) || ((j > 0)
&& (k == 0) && (l == 0)) || ((j == 0) && (k == 0) && (l == 0))) {
sprintf(b112, "%d, %d, %d", j, -k, l); //; %d", j, k, l, i);
name = static_cast<string>(b112);
if (pf->pinPa_calc[name] != 1) {
pf->pointsinPallets[name] = i;
pf->pointsinPallets_in[name] = 1;

```

```

pf->pinPa_calc[name] = 1;}
else {
for (int kk = 2; kk < 5000; kk++) {
sprintf(b112, "%d, %d, %d;%d", j, -k, l, kk);
name2 = static_cast<string>(b112);
if ((pf->pointsinPallets_in[name2] != 1) && (pf->pinPa_calc[name2] != 1) &&
(((pf->xn[pf->pointsinPallets[name]] - *it_xn) > 0.000001) ||
((pf->yyn[pf->pointsinPallets[name]] - *it_yn) > 0.000001) ||
((pf->zn[pf->pointsinPallets[name]] - *it_zn) > 0.000001))) { //i - *it_xn > 0.01) &&
((pf->yyn[i] - *it_yn) > 0.01)) {
pf->pointsinPallets[name2] = i;
pf->pointsinPallets_in[name2] = 1;
pf->pinPa_calc[name2] = 1;
cout << "pointsinPallets first = " << name2 << "; " << "pointsinPallets second = "
<< pointsinPallets[name2] << "; " << endl;
cout << "pointsinPallets_in first = " << name2 << "; " << "pointsinPallets_in second = "
<< pointsinPallets_in[name2] << "; " << endl;
break;}
else continue;}}
sprintf(b150, "pf->pointsinPallets_in[%s] = %d ", b112, pf->pointsinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointsinPallets_in[name_in]");
msgBox8.setText(b150);
if (fl_print == 1) {
if (msgBox8.exec() == QMessageBox::Yes) {}}}
if ((j >= 1) && (k >= 1) && (l >= 1) && (((*it_yn < 0.0) && (*it_yn > *it_yp_ns))) &&
(*it_yn <= (*it_yp_ns - 1))) && (*it_xn > *(it_xp_ns - 1)) && (*it_xn <= *(it_xp_ns)) && (*it_zn
> *(it_zp_ns2 - 1)) && (*it_zn <= *(it_zp_ns2))) { //(*it_xn < *(it_xp_n)) &&
(*it_xn > *(it_xp_n - 1)) && (*it_yn < *(it_yp_n)) && (*it_yn > *(it_yp_n - 1)) && (*it_zn >
*(it_zp_ns)) && (*it_zn < *(it_zp_ns - 1))))) {
sprintf(b112, "%d, %d, %d", j, -k, l); //; %d", j, k, l, i);
name = static_cast<string>(b112);
if (pf->pinPa_calc[name] != 1) {
pf->pointsinPallets[name] = i;
pf->pointsinPallets_in[name] = 1;
pf->pinPa_calc[name] = 1;}
else{
for (int kk = 2; kk < 5000; kk++) {
sprintf(b112, "%d, %d, %d;%d", j, -k, l, kk);
name2 = static_cast<string>(b112);
if ((pf->pointsinPallets_in[name2] != 1) && (pf->pinPa_calc[name2] != 1) && (((
pf->xn[pf->pointsinPallets[name]] - *it_xn) > 0.000001) || ((pf->yyn[pf->pointsinPallets[name]] -
*it_yn) > 0.000001) || ((pf->zn[pf->pointsinPallets[name]] - *it_zn) > 0.000001))) { //i -
*it_xn > 0.01) && ((pf->yyn[i] - *it_yn) > 0.01)) {
pf->pointsinPallets[name2] = i;
pf->pointsinPallets_in[name2] = 1;
pf->pinPa_calc[name2] = 1;
cout << "pointsinPallets first = " << name2 << "; " << "pointsinPallets second = "
<< pointsinPallets[name2] << "; " << endl;
cout << "pointsinPallets_in first = " << name2 << "; " << "pointsinPallets_in second = "
<< pointsinPallets_in[name2] << "; " << endl;
break;}

```

```

else continue; }}
sprintf(b150, "pf->pointsinPallets_in[%s] = %d ", b112, pf->pointsinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointsinPallets_in[name_in]");
msgBox8.setText(b150);
if (fl_print == 1) {
if (msgBox8.exec() == QMessageBox::Yes) {;}
cout << "pointsinPallets first = " << name << ", "
<< "pointsinPallets second = "
<< pointsinPallets[name] << ", " << endl;
cout << "pointsinPallets_in first = " << name << ", " << "pointsinPallets_in second = "
<< pointsinPallets_in[name] << ", " << endl;}}}}
for (i = 0, it_xn = pf->xn.begin(), it_yn = pf->yyn.begin(), it_zn = pf->zn.begin();
i < pf->xn.size(), it_xn != pf->xn.end(), it_yn != yyn.end(), it_zn != zn.end(); i++, it_xn++,
it_yn++, it_zn++)
for (j = 0, it_xp_n = pf->xp_n.begin(); j < pf->xp_n.size(), it_xp_n != pf->xp_n.end(); j++,
it_xp_n++)
for (k = 0, it_yp_n = yp_n.begin(); k < pf->yp_n.size(), it_yp_n != pf->yp_n.end(); k++,
it_yp_n++)
for (l = 0, it_zp_n = zp_n.begin(); l < pf->zp_n.size(), it_zp_n != pf->zp_n.end(); l++,
it_zp_n++) {
sprintf(b125, "%d", i);
name2 = static_cast<string>(b125);
if (pf->Attr_val_calc[name2] == 1) {continue;}
if (((j == 0)) && (((*it_xn > 0.0) && (*it_xn < *it_xp_n)))) {
sprintf(b112, "%d, %d, %d", j, k, l); //; %d", j, k, l, i);
name = static_cast<string>(b112);
sprintf(b150, "pf->pointsinPallets_in[%s] = %d ", b112, pf->pointsinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointsinPallets_in[name_in]");
msgBox8.setText(b150);
if ((pf->pointsinPallets[name] < fr_s1) && (pf->pointsinPallets[name] > 0)) {
name_in = name;
pf->pointsinPallets_in[name_in]++;
if ((inf_dim_i == 1) && (pf->pointsinPallets_in[name_in] > Npmax)) {
Npmax = pf->pointsinPallets_in[name_in];}
pf->Attr_val_calc[name2]++;
sprintf(b150, "pf->pointsinPallets_in[%s] = %d ", b112, pf->pointsinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointsinPallets_in[name_in]");
msgBox8.setText(b150);
cout << "name_in = (plusplus)" << name_in << endl;
char conv[100]; // string conv;
sprintf(conv, "%d", pf->pointsinPallets_in[name_in]);
name = name + static_cast<string>(";") + static_cast<string>(conv);
pf->pointsinPallets[name] = i;
pf->pointsinPallets_in[name] = 1;}
if (pf->pointsinPallets[name] == 0) {
name_in = name;
pf->pointsinPallets_in[name_in] = 1;
pf->Attr_val_calc[name2]++;
sprintf(b150, "pf->pointsinPallets_in[%s] = %d ", b112, pf->pointsinPallets_in[name_in]);
msgBox8.setWindowTitle("pf->pointsinPallets_in[name_in]");}

```

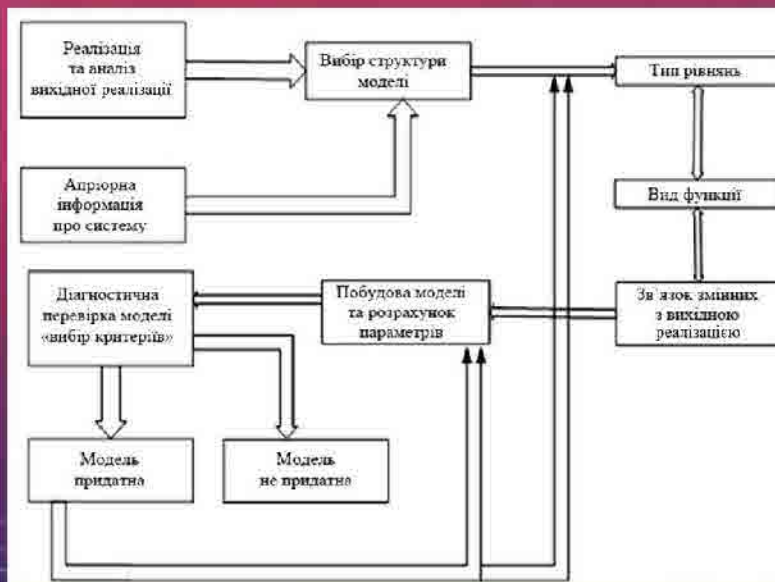
Слайди мультимедійної презентації

Дослідження режимів роботи
динамічної системи засобами
комп'ютерного моделювання

Токарчук Гліб, ОТФК ОНТУ



Послідовність етапів
побудови
математичної моделі



Функціональна
схема
реконструкції
систем за
скалярною
реалізацією

Система ітераційних рівнянь для опису динамічної системи

$$\Phi(x, y, z) = \begin{cases} x_{n+1} = x_n - k_{xy} p x_n^2 + k_{yx} q y_n^2 + x_m \\ y_{n+1} = y_n + k_{yx} p x_n^2 - (k_{xy} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ z_{n+1} = z_n + k_{yz} q y_n^2 - (k_{xy} + k_{oz}) r z_n^2 \end{cases} \quad (1)$$

$$\Phi(x, y, z, w) = \begin{cases} x_{n+1} = x_n - k_{xy} p x_n^2 + k_{yx} q y_n^2 + x_m \\ y_{n+1} = y_n + k_{yx} p x_n^2 - (k_{xy} + k_{yz}) q y_n^2 + k_{zy} r z_n^2 \\ z_{n+1} = z_n + k_{yz} q y_n^2 - (k_{xy} + k_{zw}) r z_n^2 + k_{xz} s w_n^2 \\ w_{n+1} = w_n + k_{zw} r z_n^2 - (k_{xy} + k_{oz}) s w_n^2 \end{cases} \quad (2)$$

де x, y, z, w – динамічні змінні, що описують, відповідно, стан кожного з рівнів динамічної системи; k_{ij} – перехідні коефіцієнти, що характеризують статичну і динамічну взаємодію рівнів відповідно; p, q, r, s – розподільні коефіцієнти, x_m – потік, що входить до системи, причому, $\{k_{ij}\} \in \{p, q, r, s\} \in (0, 1)$, $\{x, y, z\} \in R$, $x_m = \text{const} \in R^+$.

Інтерфейс застосунку для моделювання динамічної системи

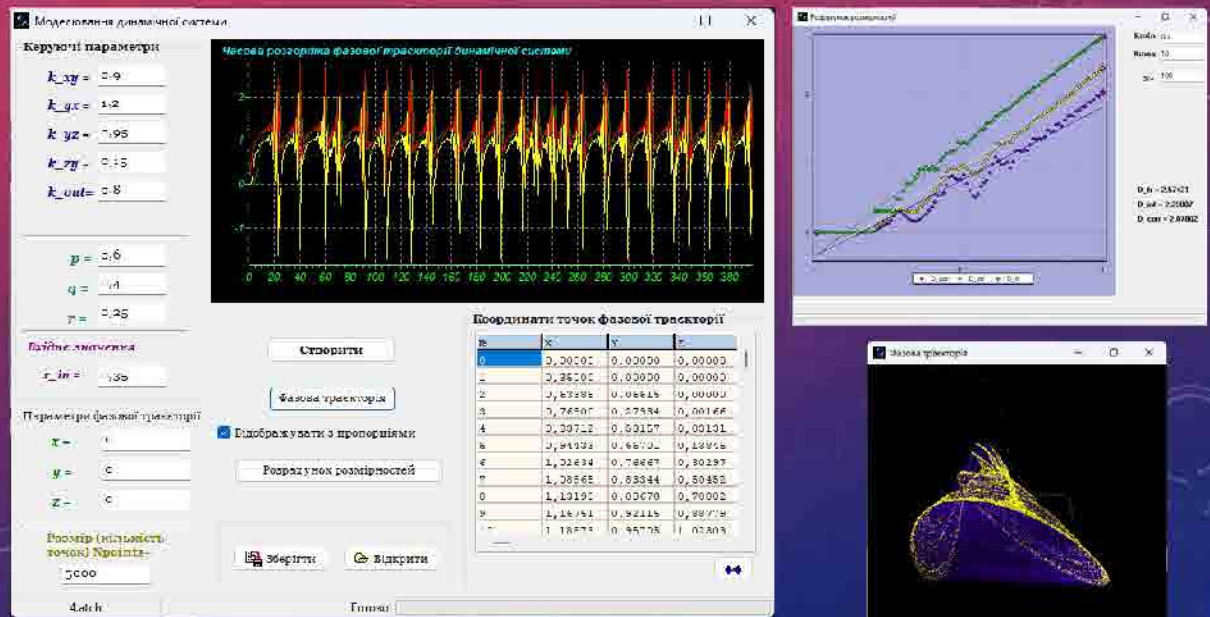
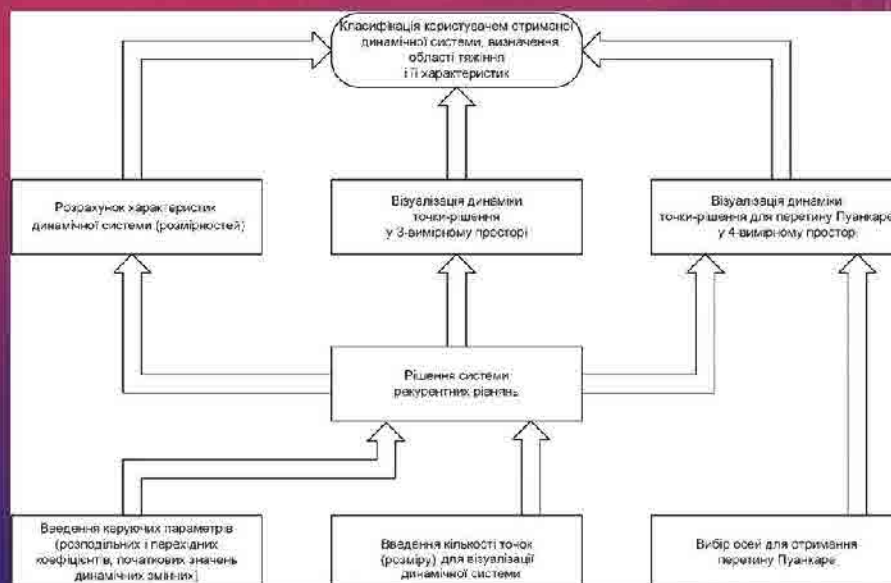
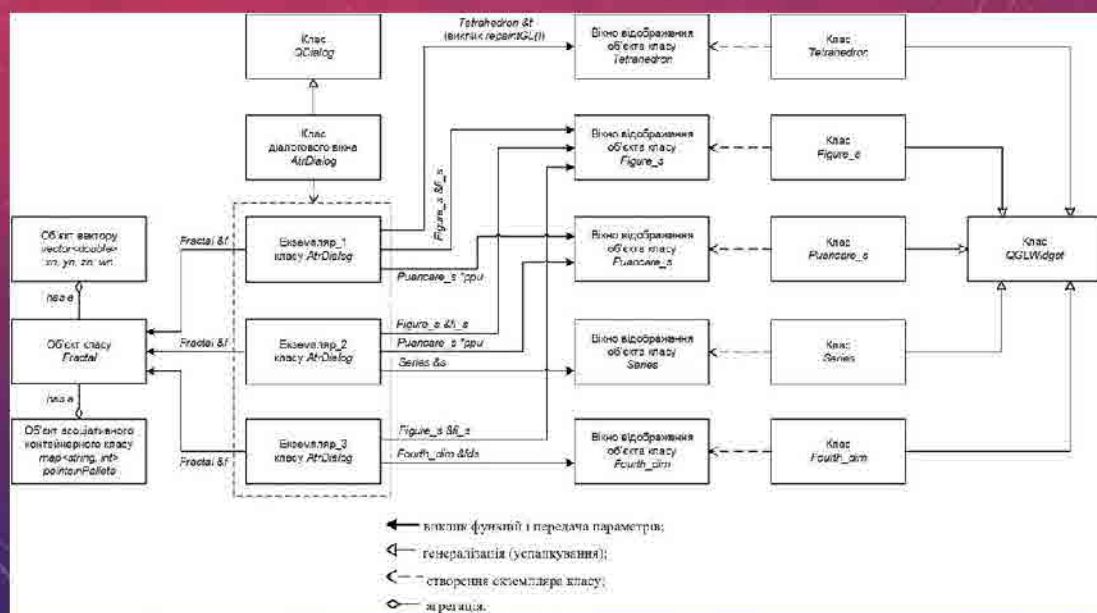


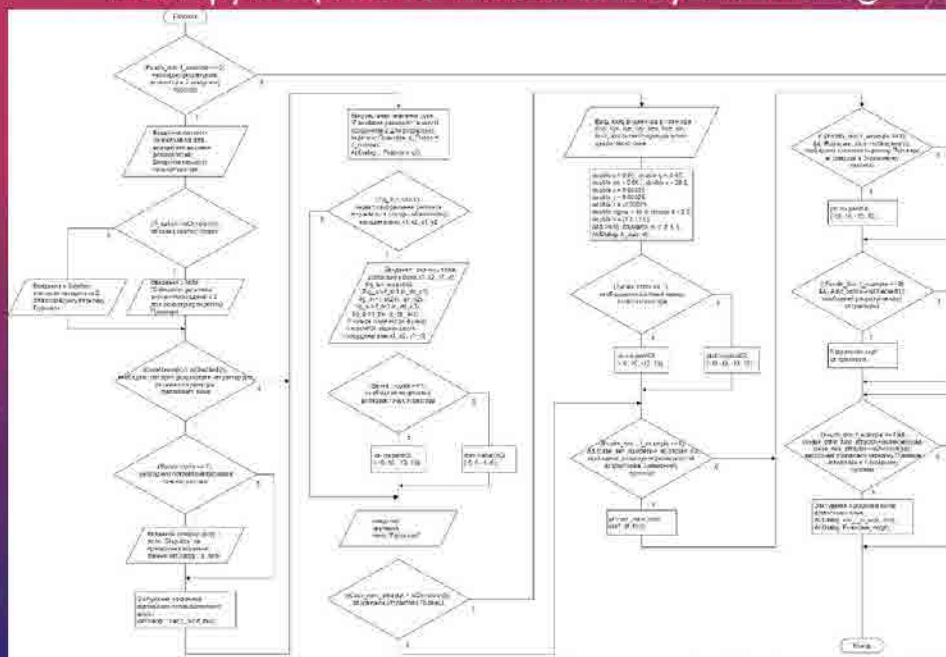
Схема обчислювального процесу у програмному комплексі



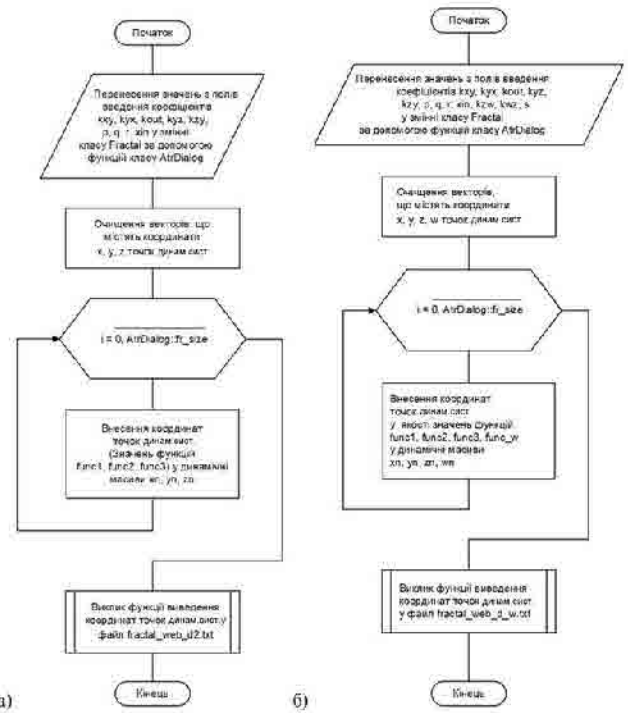
Діаграма взаємодії об'єктів АТД програмного застосунку



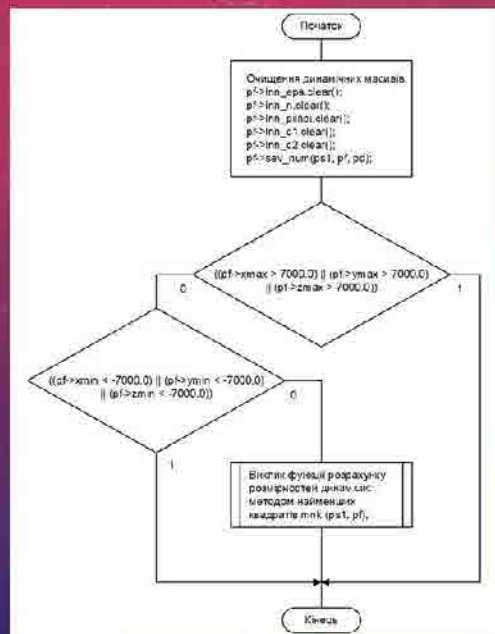
БСА функції CalcClicked класу AtrDialog



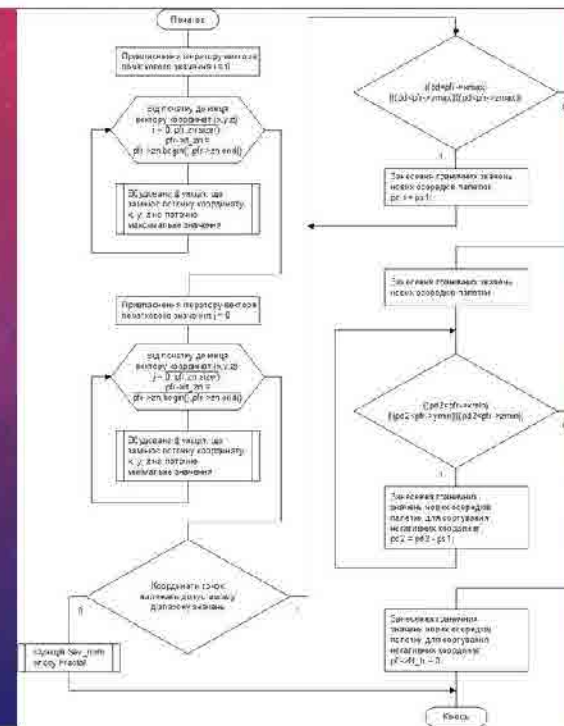
БСА функцій web_d2 і web_d_w класу Fractal



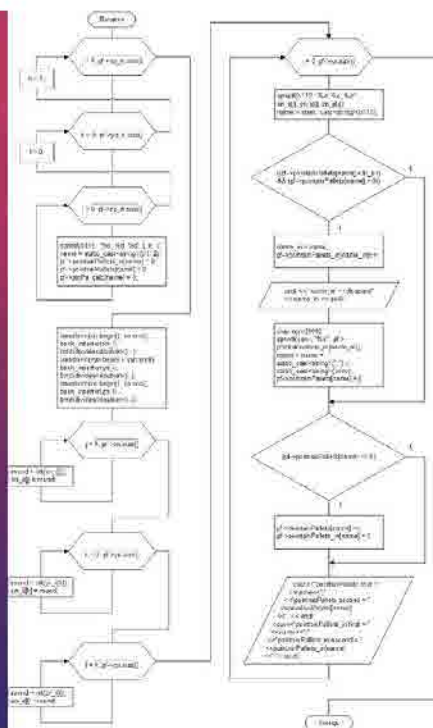
БСА функції sev_num_exes класу Fractal

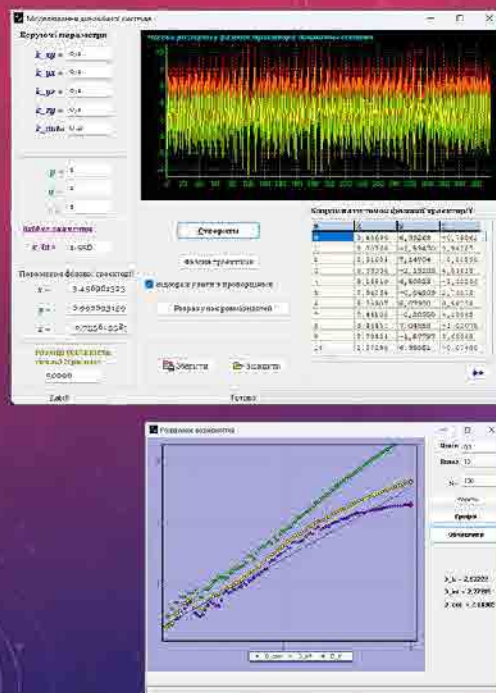


БСА функції Pallett класу Fractal

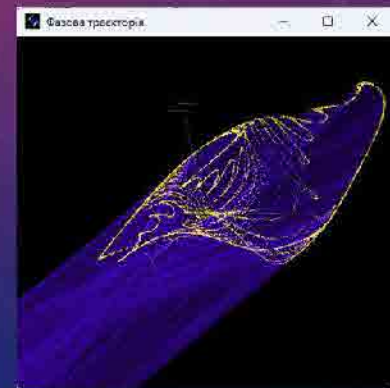


БСА функції Dimension класу Fractal

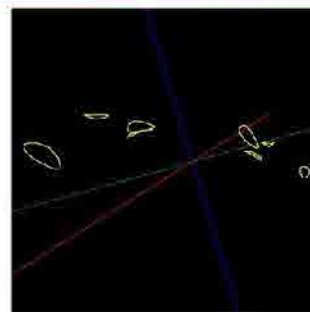
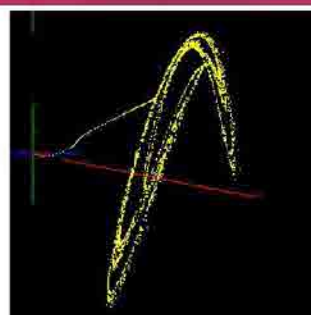




Інтерфейс застосунку для
чотиривимірної моделі
в процесі формування
перетину Пуанкаре



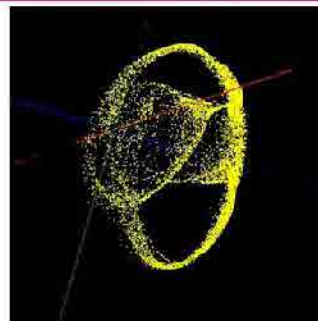
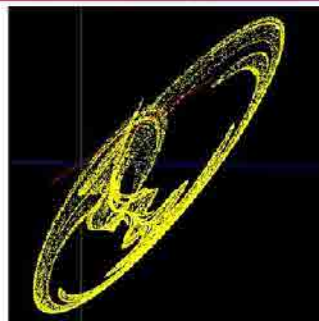
Візуалізація динамічної системи
у хаотичному режимі



$k_{xy} = 0.5;$	$p = 0.001;$	$x = 0;$	$k_{xy} = 0.1;$	$p = 1;$	$x = 3;$
$k_{yx} = 0.4;$	$q = 0.09;$	$y = 0;$	$k_{yx} = 0.1;$	$q = 1;$	$y = 7.9;$
$k_{yz} = 0.3;$	$r = 0.04;$	$z = 0;$	$k_{yz} = 0.1;$	$r = 1;$	$z = -2.13;$
$k_{zy} = 0.3;$	$s = 0.1;$	$w = 0;$	$k_{zy} = 0.1;$	$s = 0.1;$	$w = 0.01;$
$k_{wx} = 0.000001;$	$x_0 = 6;$	$N_{points} = 5000;$	$k_{wx} = 0.00001;$	$x_0 = 1.536;$	$N_{points} = 5000;$
$k_{wz} = 0.1;$	$k_{wz} = 0.4;$		$k_{wz} = 0.1;$	$k_{wz} = 0.2;$	

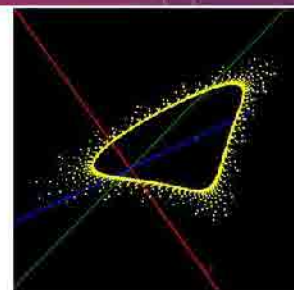
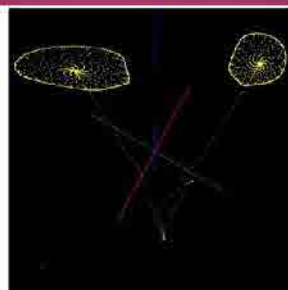
$D_F = 3.29, \quad D_{F_0} = 2.18, \quad D_F = 3.2, \quad D_{F_0} = 2.08.$

Візуалізація динамічної системи у хаотичному режимі



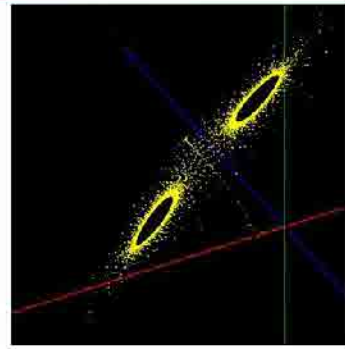
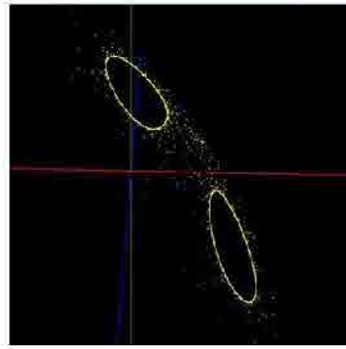
$k_{xy} = 0,65;$	$p = 1;$	$x = 0,927;$	$k_{xy} = 0,5;$	$p = 0,01;$	$x = 17,93;$
$k_{yx} = 0,25;$	$q = 1;$	$y = 0,49;$	$k_{yx} = 0,4;$	$q = 1;$	$y = 1,46;$
$k_{yz} = 0,65;$	$r = 1;$	$z = -0,13;$	$k_{yz} = 0,3;$	$r = 1;$	$z = -0,74;$
$k_{zy} = 0,25;$	$s = 0,1;$	$w = 0,01;$	$k_{zy} = 0,3;$	$s = 1;$	$w = 0,1;$
$k_{zw} = 0,000001;$	$x_{in} = 0,435;$	$N_{points} = 5000;$	$k_{zw} = 0,001;$	$x_{in} = 0,475;$	$N_{points} = 5000;$
$k_{wz} = 0,1;$	$k_{out} = 0,4;$		$k_{wz} = 1;$	$k_{out} = 0,4;$	
$D_F = 3,58, \quad D_{\beta_p} = 2,56.$			$D_F = 3,66, \quad D_{\beta_p} = 2,64.$		

Візуалізація динамічних систем у квазіперіодичному режимі



$k_{xy} = 0,5;$	$p = 0,065;$	$x = 21,67;$	$k_{xy} = 0,5;$	$p = 0,1;$	$x = 22;$	$k_{xy} = 0,5;$	$p = 0,05;$	$x = 30;$
$k_{yx} = 0,4;$	$q = 0,03;$	$y = 15,53;$	$k_{yx} = 0,4;$	$q = 0,1;$	$y = 10;$	$k_{yx} = 0,4;$	$q = 0,065;$	$y = 10;$
$k_{yz} = 0,3;$	$r = 0,03;$	$z = 15,64;$	$k_{yz} = 0,3;$	$r = 0,001;$	$z = -91;$	$k_{yz} = 0,3;$	$r = 0,0075;$	$z = 10;$
$k_{zy} = 0,3;$	$s = 0,1;$	$w = 0,01;$	$k_{zy} = 0,3;$	$s = 0,1;$	$w = 0,01;$	$k_{zy} = 0,3;$	$s = 0,1;$	$w = 0,1;$
$k_{zw} = 0,000001;$	$x_{in} = 6,05;$	$N_{points} = 5000;$	$k_{zw} = 0,000001;$	$x_{in} = 2,6;$	$N_{points} = 5000;$	$k_{zw} = 0,000001;$	$x_{in} = 5,9;$	$N_{points} = 5000;$
$k_{wz} = 0,1;$	$k_{out} = 0,4;$		$k_{wz} = 0,1;$	$k_{out} = 0,4;$		$k_{wz} = 0,1;$	$k_{out} = 0,4;$	
$D_F = 3,14, \quad D_{\beta_p} = 2,08.$			$D_F = 3,21, \quad D_{\beta_p} = 2,07.$			$D_F = 3,18, \quad D_{\beta_p} = 2,16.$		

Візуалізація динамічної системи у дивному режимі



$k_{xy} = 0,8;$ $p = 0,95;$ $x = 0,025;$
 $k_{yx} = 1,1;$ $q = 0,25;$ $y = 0,025;$
 $k_{yz} = 1,0;$ $r = 0,2;$ $z = 0,025;$
 $k_{zy} = 0,2;$ $s = 0,1;$ $w = 0,025;$
 $k_{zw} = 0,0001;$ $x_{in} = 0,33;$ $N_{points} = 5000.$
 $k_{wz} = 0,1;$ $k_{out} = 0,48;$

$D_{\hat{F}} = 3,24$

$D_{\hat{F}_p} = 2,13$

$k_{xy} = 0,45;$ $p = 0,4;$ $x = 0,01;$
 $k_{yx} = 0,5;$ $q = 0,2;$ $y = 0,01;$
 $k_{yz} = 0,3;$ $r = 0,55;$ $z = 0,01;$
 $k_{zy} = 0,4;$ $s = 0,1;$ $w = 0,01;$
 $k_{zw} = 0,0001;$ $x_{in} = 0,55;$ $N_{points} = 5000.$
 $k_{wz} = 0,1;$ $k_{out} = 0,45;$

$D_{\hat{F}} = 3,51$

$D_{\hat{F}_p} = 2,41.$

ВИСНОВКИ

Основні параметри, що дозволяють класифікувати динамічні системи по типу – їх розрахована фрактальна розмірність, а також розташування по осях в фазовому просторі, вигляд відбитку перетину і щільність точок в різних областях їх візуалізації. Визначено три типи динамічних систем: хаотична, квазіперіодична і дивна. Залежно від величин перехідних змінних і розподільних коефіцієнтів змінюються сценарії еволюції динамічної системи, описаної рекурентними рівняннями запропонованої моделі. Тип сценарію знаходиться в однозначній відповідності з типом динамічної системи та її перетином, розмірностями.

ВІДГУК

керівника про кваліфікаційну роботу бакалавра

Токарчука Гліба Сергійовича

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Тема кваліфікаційної роботи Дослідження режимів роботи динамічної системи засобами комп'ютерного моделювання

ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

а) Обсяг і якість виконання роботи (графічного матеріалу і розрахунково-пояснювальної записки) Випускна робота виконана відповідно технічному завданню. Пояснювальна записка до випускної роботи містить 79 сторінок. У пояснювальній записці описано етапи моделювання та візуалізації динамічних систем та процес розробки програмного застосунку у ICP Visual Studio з використанням фреймворку Qt. Графічна частина складається з 20 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра, розробку виконано у повному обсязі.

б) Самостійність роботи Протягом виконання випускної бакалаврської роботи Токарчук Гліб поступово та послідовно виконував всі етапи, проявив ініціативу у створенні загальної концепції та реалізації випускної роботи. Всі роботи він виконував самостійно, з оглядом на рекомендації керівника.

в) Теоретична підготовка здобувача освіти

Токарчук Гліб під час роботи над випускною бакалаврською роботою вивчив і опрацював достатню кількість літературних джерел за даною тематикою.

Вважаю, що теоретична підготовка здобувача освіти достатня і він готовий до захисту роботи.

г) Вміння розв'язувати виробничі і конструкторські питання на базі останніх досліджень науки і техніки, передових методів виробництва

Під час виконання роботи Токарчук Гліб мав змогу самостійно приймати окремі рішення з виконання програмної частини роботи та показав вміння організовано працювати над поставленою задачею, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як Microsoft Visual Studio, Qt, Microsoft PowerPoint, Microsoft Visio та ін.

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Відмінно

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові Кривченко Юрій Вікторович

Місце роботи і посада керівника роботи ВСП "Одеський технічний фаховий коледж ОНТУ", викладач кафедри комп'ютерної інженерії, голова циклової комісії комп'ютерних технологій та програмної інженерії

Підпис

[Підпис]
«13» *березня* 2024 р.

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра здобувача освіти
відділення комп'ютерних систем

Токарчука Гліба Сергійовича

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Освітня програма «Комп'ютерна інженерія»

Керівник дипломного проекту (роботи) Кривченко Юрій Вікторович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Дослідження режимів роботи динамічної системи засобами комп'ютерного моделювання

Обсяг розрахунково-пояснювальної записки 79 сторінок

Обсяг графічної (презентаційної) частини 20 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню
Представлена на рецензію кваліфікаційна робота бакалавра повністю відповідає меті проектування та технічному завданню. Тематика кваліфікаційної роботи є актуальною для своєї галузі та присвячена проблемі моделювання та візуалізації динамічних систем, складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту (роботи)
Кваліфікаційна робота складається зі вступу, двох розділів, висновків, переліку використаних джерел. У основному розділі розглянуті питання створення математичної моделі, вибору технологій та інструментів, розробки схеми ОП, структур даних, БСА, інтерфейсу результатів моделювання. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора ЕОТ.

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 охайно 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) _____

Реалізовано математичну модель динамічної системи, яка статистично достовірно відображує поведінку об'єктів, заданих системою рівнянь, у просторі;

Розроблене програмне забезпечення має широкі функціональні можливості та дозволяє виконувати пошук різних динамічних систем перебором параметрів

д) основні недоліки дипломного проекту (роботи) _____

1. Для більш простого завдання початкових параметрів динамічної системи варто було б передбачити деякі їх межі у інтерфейсі застосунку;

2. Варто було б передбачити кольорове маркування окремих рівнів для візуалізації динамічної системи.

Оцінка розрахункової частини _____ Відмінно

Оцінка графічної частини _____ Відмінно

Загальна оцінка _____ Відмінно

Прізвище, ім'я, по батькові рецензента _____ Васіліу Євген Вікторович

Місце роботи і посада рецензента _____ Державний університет інтелектуальних технологій і зв'язку, д.т.н., проф. кафедри КБ та ТЗІ

ПІДПИС ПОСВІАЧУЮ
НАЧАЛЬНИК ВІДДІЛУ
КАДРІВ ДУІТЗ



Підпис: _____

« 17 » 06 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016231721

Дата перевірки:
06.05.2024 18:10:02 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
06.05.2024 18:15:04 EEST

ID користувача:
100011688

Назва документа: **2БКС-28_Гліб_Токарчук**

Кількість сторінок: **58** Кількість слів: **10357** Кількість символів: **74577** Розмір файлу: **7.55 MB** ID файлу: **1016011847**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

11.7%

Схожість

Найбільша схожість: **3.54%** з Інтернет-джерелом (<https://card-file.ontu.edu.ua/server/api/core/bitstreams/9fc96430-656...>)

11.7% Джерела з Інтернету

300

Сторінка 60

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

31

Підозріле форматування

26
сторінок

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

Ми, що нижче підписалися,

Токарчук Гліб Сергійович,
здобувач освіти гр. 2БКС-28, та

Кривченко Юрій Вікторович,
керівник випускної кваліфікаційної роботи,

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

«Дослідження режимів роботи динамічної системи засобами комп'ютерного моделювання» (автор роботи – Токарчук Г.С., керівник роботи – Кривченко Ю.В.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 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



/ Токарчук Г.С. /

Керівник



/ Кривченко Ю.В. /

«13» червня 2024 р.