

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету)

Кафедра електронної техніки

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

Олександр

ВОВНА

(підпис)

(ім'я, прізвище)

«___» _____ 2022 р.

Випускна кваліфікаційна робота

магістра

(освітній ступінь)

на тему «Підсистема вирішувачів рівнянь проблемно орієнтованого паралельного моделюючого середовища»

Виконав: студент

2

курсу, групи

КІм-21

(шифр групи)

спеціальності

123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

Микита ЛИТВИНОВ

(ім'я та прізвище)

(підпис)

Керівник

Проф. каф. ЕТ, д.т.н., проф. Володимир СВЯТНИЙ

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензенти

Зав. кафедри ЕІН, к.т.н., доц. Олександр КОЛЛАРОВ

(посада, науковий ступінь, вчене звання, ім'я та прізвище)

(підпис)

Проф. каф. ПМІ, д. т. н., проф Євген БАШКОВ

Нормконтроль:

_____ доц. каф. ЕТ, к.т.н., доц. Віталій ШАМАЄВ

(підпис)

*Засвідчую, що у цій
дипломній роботі
немає запозичень з
праць інших авторів без
відповідних посилань.*

Студент

(підпис)

Луцьк – 2022 р.

ДВНЗ «Донецький національний технічний університет»

Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра електронної техніки
Освітній ступінь магістр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Вовна

« _____ » _____ 2022 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Литвинову Микиті Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Підсистема вирішувачів рівнянь проблемно орієнтованого паралельного моделюючого середовища»

керівник роботи Святний Володимир Андрійович, д. т. н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від 28.09 2022 року № 446

2. Строк подання студентом роботи 8 грудня 2022 року

3. Вихідні дані до роботи: результати переддипломної практики та самостійної роботи

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1. Огляд архітектур MIMD-систем, математична модель та Simulation-модель МДОРП, структура та проблемна орієнтація РПМС .
2. Функціональність і структура підсистеми вирішувачів рівнянь.
3. Підходи до розпаралелювання, алгоритми і програмна імплементація.
4. Методика і результати роботи

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сергієнко О.І.		

6. Дата видачі завдання 27.09.2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	MIMD-системи, модель і Simulation-модель МДОРП, структура РПМС	28.09.22-01.10.22	
2	Структура підсистеми, підходи до розпаралелювання	02.10.22-10.10.22	
3	Розробка алгоритмів вирішувачів	11.10.22-31.10.22	
4	Програмна реалізація	01.11.22-17.11.22	
5	Відлагодження та аналіз результатів	18.11.22-25.11.22	
6	Оформлення пояснювальної записки	26.11.22-07.12.22	
7	Нормоконтроль пояснювальної записки	08.12.22-13.12.22	
8	Рецензування роботи	14.12.22-16.12.22	
9	Захист роботи	23.12.22	
10	MIMD-системи, модель і Simulation-модель МДОРП, структура РПМС	28.09.22-01.10.22	

Студент

(підпис)

Литвинов М.О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Святний В.А.

(прізвище та ініціали)

АНОТАЦІЯ

Литвинов М. О. Підсистема вирішувачів рівнянь проблемно орієнтованого паралельного моделюючого середовища / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 123 «Комп'ютерна інженерія» – ДВНЗ ДонНТУ, Покровськ/Луцьк, 2022.

Об'єктом дослідження є підсистема вирішувачів рівнянь проблемно орієнтованого розподіленого паралельного моделюючого середовища (ПОРПМС).

Предметом дослідження виступають алгоритмічні й апаратно-програмні засоби, які забезпечують функціонування підсистеми вирішувачів рівнянь середовища.

Наукова новизна: отримав подальший розвиток принцип декомпозиції РПМС шляхом побудови підсистеми вирішувачів рівнянь у частинних похідних, що враховують проблемну орієнтацію середовища.

Практична цінність: розроблена підсистема РПМС впроваджується в практику роботи ParSimTech-центру ДонНТУ та в навчальний процес за спеціальністю «Комп'ютерна інженерія».

Для розробки підсистеми РПМС було використано мову програмування C, технологію програмування Windows Forms та інструментальне середовище Visual Studio 2019, MPI- та OpenMP-стандарти паралельного програмування.

Ключові слова:

ПРОБЛЕМНО ОРІЄНТОВАНЕ ПАРАЛЕЛЬНЕ МОДЕЛЮЮЧЕ СЕРЕДОВИЩЕ, МЕРЕЖЕВИЙ ДИНАМІЧНИЙ ОБ'ЄКТ З РОЗПОДІЛЕНИМИ ПАРАМЕТРАМИ, МОДЕЛЬ, SIMULATION-МОДЕЛЬ, СИМУЛЯТОР, ТОПОЛОГІЧНИЙ АНАЛІЗАТОР, ГЕНЕРАТОР РІВНЯНЬ, ВИРІШУВАЧ РІВНЯНЬ, ПАРАЛЕЛЬНЕ ПРОГРАМУВАННЯ.

ANNOTATION

Litvinov M. O. Gleichungslösersubsystem der problemorientierten parallelen Simulationsumgebung / Final qualification work for the degree of "master" in the specialty 123 "Computer Engineering" - SHEI DonNTU, Pokrovsk/Lutsk, 2022.

Objekt der Untersuchungen ist das Gleichungslösersubsystem der problemorientierten verteilten parallelen Simulationsumgebung (POVPSU).

Die Gegenstände der Untersuchungen und Entwicklungen sind die algorithmischen und HW/SW-Mittel, welche die Funktionierung des Subsystems von Gleichungslöser verwirklichen.

Wissenschaftliche Modernität: die erhaltene Ergebnisse beitragen zur weiteren Entwicklung der Dekomposition von VPSU durch Aufbau des Gleichungslösersubsystems für partielle Differentialgleichungen mit der Berücksichtigung der VPSU-Problemorientierung.

Der Nutzwert: vorgeschlagenes Subsystem der VPSU wird in Praxis des ParSimTech-Zentrums der DonNTU und in der Lehre des Fachgebiets „Computer Engineering“ eingesetzt.

Keywords: PROBLEMORIENTIERTE PARALLELE
SIMULATIONSUMGEBUNG, DYNAMISCHES NETZOBJEKT MIT
VERTEILTEN PARAMETERN, MODELL, SIMULATIONSMODELL,
SIMULATOR, TOPOLOGIEANALYSATOR, GLEICHUNGSGENERATOR,
PARALLELE PROGRAMMIERUNG, GLEICHUNGSLÖSER .

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ РОБІТ, РЕЛЕВАНТНИХ ТЕМІ, ЗАДАЧІ РОЗРОБОК І ДОСЛІДЖЕНЬ	11
1.1 Технологічна характеристика і математична модель мережевого динамічного об'єкту з розподіленими параметрами, на який орієнтовано РПМС.....	11
1.1 Структура проблемно орієнтованого РПМС.....	18
1.2 Принцип декомпозиції РПМС на підсистеми.	20
1.3 Задачі розробок і досліджень	23
2 ФУНКЦІОНАЛЬНІСТЬ І СТРУКТУРА ПІДСИСТЕМИ ВИРІШУВАЧІВ РІВНЯНЬ	24
2.1 Визначення підсистеми, її основні функції та структура	24
2.2 Формальні підходи до побудови дискретних Simulation-моделей МДОРП, обчислювальні методи	27
2.3 Рівні розпаралелювання Simulation-моделей.....	37
2.4 Висновки.....	40
3 Розробка послідовних вирішувачів рівнянь МДОРП	41
3.1 Вирішувачі j-гілок на основі послідовної мови моделювання	41
3.1.1 Тестовий МДОРП: топологія, параметри.	41
3.1.2 Апроксимація j-гілки за просторовою координатою	42
3.1.3 Рівняння Simulation-моделі j-гілки	42
3.1.4 Блок-схема БО-вирішувача рівнянь j-гілки, процеси $Q_j(v, t)$	44
3.2 Вирішувачі рівнянь МДОРП на основі послідовної мови моделювання ..	45
3.2.1. Специфікації апроксимованих гілок та системи рівнянь Simulation-моделі МДОРП.....	45
3.2.2 Simulation-модель МДОРП і розробка блок-схеми БО-вирішувача рівнянь МДОРП	52
3.3 Вирішувачі j-гілок на основі послідовної C-мови програмування	57
3.3.1 Дискретна Simulation-модель j-гілки.....	57
3.3.2 Блок-схема і текст програми C-вирішувача рівнянь j-гілки, процеси $Q_j(v, t)$	58

3.4 Вирішувачі рівнянь МДОРП на основі послідовної С-мови програмування	60
3.4.1 Дискретна Simulation-модель як основа С-вирішувача МДОРП.....	60
3.4.2 Блок-схема і текст програми С-вирішувача рівнянь МДОРП, процеси $Q_j(v, t)$	64
3.5 Висновки	67
4 АЛГОРИТМИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ ПАРАЛЕЛЬНИХ ВИРІШУВАЧІВ.....	68
4.1 Модель і Simulation-модель мережевого динамічного об'єкту з зосередженими параметрами.....	68
4.2 Можливі підходи до розпаралелювання програми ВР.....	69
4.3 Дискретна Simulation-модель МДОЗП і послідовний вирішувач.....	72
4.4 Розпаралелювання за принципом «Х- та Y-процеси».....	72
4.5 Розпаралелювання за принципом «XY-процеси».....	76
4.6 Підготовка до імплементації паралельного вирішувача.	78
4.7 Апріорний аналіз ефективності паралельних вирішувачів на рівні "Гілка МДОРП - MIMD-процес"	80
4.7.1 Розподіл завантаження між віртуальними процесами і оцінка нерівномірності	80
4.7.2 Порівняння об'ємів обчислювальних і допоміжних операцій, віртуальне прискорення обчислень паралельного вирішувача.	81
4.7.3 Можливість реалізації за принципом «процес – процесор»	84
4.7 Висновки.....	85
ЗАГАЛЬНІ ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ	86
ПЕРЕЛІК ПОСИЛАНЬ	88
ДОДАТОК А – Заява на затвердження теми і керівника роботи	91
ДОДАТОК Б – Охорона праці.....	92
ДОДАТОК В1 - БО-вирішувачі для гілок $Q_1 \dots Q_8$ і процеси $Q_j(t)$	101
ДОДАТОК В2 - С-програма вирішувача для j-гілок).	107
ДОДАТОК В3 - Динамічні процеси j-гілок, отримані на С-вирішувачі.....	109
ДОДАТОК В4 - Текст С-програми вирішувача рівнянь МДОРП.....	113
ДОДАТОК Г - Перелік зауважень нормоконтролера	119

ВСТУП

Актуальність теми. Розробка проблемно орієнтованих розподілених паралельних моделюючих середовищ (ПОРПМС) належить до проблематики технологій паралельного моделювання складних динамічних систем (Parallel Simulation Technology, Parallele Simulationstechnik – ParSimTech-проблеми). Серед об'єктів моделювання, що потребують застосування паралельних обчислювальних систем, актуальними є мережеві динамічні об'єкти з розподіленими параметрами (МДОРП), що широко розповсюджені в різних предметних областях (галузях промисловості) – в гірництві, гідравліці, аеромеханіці, газодинаміці, електротехніці, хімічних та біохімічних технологіях, машинобудуванні, електроніці, в інтердисциплінарних проєктах з захисту довкілля. Вони належать до складних динамічних систем (СДС), що характеризуються великою кількістю одночасно функціонуючих компонент, нелінійністю залежностей між параметрами, різними за фізичним змістом і одночасно працюючими динамічними процесами, ієрархічністю розташування засобів регулювання та керування, залежними від навантажень характеристиками джерел енергії (активні елементи – генератори, вентилятори, насоси, компресори, акумулятори і ін.). Динаміка процесів в МДОРП є предметом активної модельної підтримки інноваційних проєктів як вирішального фактору правильності проєктних рішень на всіх етапах розробок і, як наслідок, конкурентоспроможності об'єктів нової техніки та розвинутих технологій (Model Drive Engineering, Modellgetriebene Entwicklung – MDE/MGE-Problem). При вирішенні цих задач стосовно МДОРП реальної промислової розмірності виникає певний поріг складності математичних моделей, який може бути подоланим шляхом застосування паралельних обчислювальних систем (ПОС), що є основою симуляторів як апаратно-програмних реалізацій Simulation-моделей МДОРП. Наукова школа ДонНТУ з ParSimTech-проблем співпрацює з надпотужним обчислювальним центром та інститутами Штутгартського університету (HLRS –

Höchstleistungsrechenzentrum Stuttgart, IPVS – Institut für parallele und verteilte Systeme, IAS – Institut für Automatisierungs- und Softwaresysteme). Запропоновано проблемну орієнтацію розподіленого паралельного моделюючого середовища (РПМС) на вирішення задач моделювання МДОРП. Центральною функціональною складовою середовища має бути підсистема вирішувачів рівнянь у частинних похідних. Розробці цієї підсистеми з врахуванням актуальних архітектурних і програмних особливостей MIMD-систем присвячено дану кваліфікаційну роботу. Її виконано в рамках проєкту з побудови Українського науково-навчального ParSimTech-центру за сприяння стипендіальної DAAD-програми ім. Леонарда Ейлера.

Мета роботи – розробити підсистему вирішувачів рівнянь Simulation-моделей, що зумовить орієнтацію паралельного моделюючого середовища на вирішення задач активної модельної підтримки МДОРП-проєктів і сприятиме підвищенню рівня дружності РПМС до користувачів.

Для досягнення цієї мети в роботі вирішуються наступні задачі розробок і досліджень:

- 1) Дано технологічну характеристику, розглянуто математичну модель і Simulation-модель мережевого динамічного об'єкту з розподіленими параметрами (МДОРП), на який орієнтовано РПМС.
- 2) Визначено функціональність підсистеми вирішувачів рівнянь РПМС.
- 3) Розроблено структуру підсистеми.
- 4) Запропоновано генератори дискретних Simulation-моделей і послідовні вирішувачі рівнянь гілок МДОРП.
- 5) Розроблено алгоритм функціонування паралельного вирішувача на рівні «Гілка МДОРП – MIMD-процес».
- 6) Виконано експерименти і запропоновано використати результати розробок в ParSimTech-центрі ДонНТУ.

Об'єктом дослідження є підсистема вирішувачів рівнянь проблемно орієнтованого розподіленого паралельного моделюючого середовища (ПОРПМС).

Предметом дослідження виступають алгоритмічні й апаратно-програмні засоби, які забезпечують функціонування підсистеми вирішувачів рівнянь середовища.

Наукова новизна: отримав подальший розвиток принцип декомпозиції РПМС шляхом побудови підсистеми паралельних вирішувачів рівнянь у частинних похідних, що враховують проблемну орієнтацію середовища.

Практична цінність: розроблена підсистема ПОРПМС впроваджується в практику роботи ParSimTech-центру ДонНТУ та в навчальний процес за спеціальністю «Комп'ютерна інженерія».

1 АНАЛІЗ РОБІТ, РЕЛЕВАНТНИХ ТЕМІ, ЗАДАЧІ РОЗРОБОК І ДОСЛІДЖЕНЬ

1.1 Технологічна характеристика і математична модель мережевого динамічного об'єкту з розподіленими параметрами, на який орієнтовано РПМС

Для вирішення MDE-проблем кафедрою КІ запропоновано розподілене паралельне моделююче середовище (РПМС), в якому мають реалізовуватись паралельні MIMD-симулятори складних динамічних систем, до яких належить і мережеві динамічні об'єкти з розподіленими параметрами (МДОРП). Ця проблематика вирішується в співробітництві з надпродуктивним обчислювальним центром (HLRS) Штутгартського університету в рамках проєкту «ParSimTech: Розробка й імплементація українського дослідницько-навчального центру технологій паралельного моделювання в Донецькому національному технічному університеті (м. Покровськ)». РПМС орієнтовано на вирішення задач активної модельної підтримки інноваційних проєктів для системи вентиляції шахт, формальний опис якої дано в [1,2], а узагальнену структуру наведено на рис. 1.1.

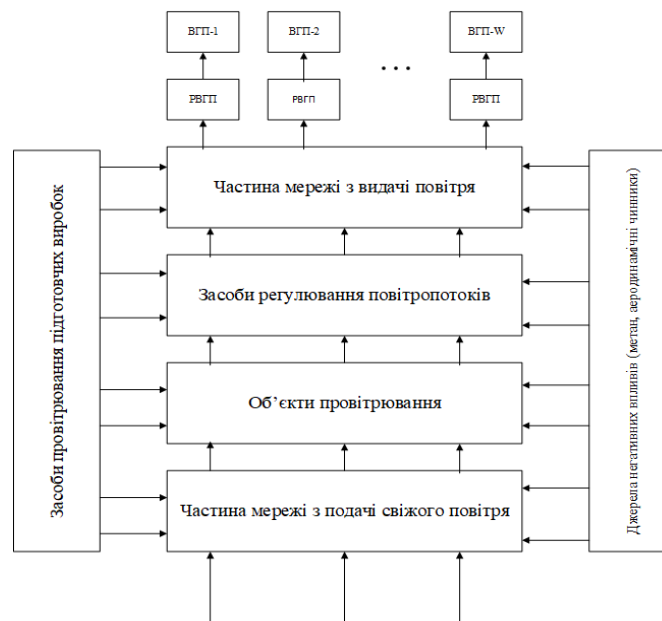


Рисунок 1.1 – Узагальнена структура системи вентиляції шахти як МДОРП.

Технологічно система вентиляції є важливим ресурсом гарантування безпечної праці гірників, подає на робочі місця свіже повітря і виводить домішки метану, який є головною загрозою підземних робіт. Основними об'єктами провітрювання є лави і підготовчі виробки. Вентилятори ВГП-1, ВГП-2, ..., ВГП- w головного провітрювання генерують різниці тиску, які необхідні для забезпечення потрібних за умовами безпеки потоків повітря в названих об'єктах. Шляхи подачі свіжого і відведення насиченого газами повітря – це підземні виробки, що утворюють аеродинамічну мережу, в якій на рис.1.1 виділено частини мережі свіжих потоків, провітрювання об'єктів і відведення відпрацьованих потоків.

Основними джерелами газових домішок є лави і вироблені простори: свіжі потоки проходять вздовж робочих просторів лав, фільтрують через вироблені простори і на виходах з цих об'єктів мають певні концентрації метану, які обмежуються правилами безпеки підземних робіт. З виконанням цих обмежень пов'язана проблема автоматизації систем вентиляції, вирішення якої ґрунтується на формальному описі й моделюванні вентиляційних систем як мережевих динамічних об'єктів керування з розподіленими параметрами [1,2].

Топологічно МДОРП представляється як граф $G(m, n)$, що кодується таблицею 1.1.

Тут маємо: QJ – потік повітря в J -гілці; AKI та EKK – початковий та кінцевий вузли J -гілки; $(I, K) \in (1, 2, \dots, n)$, $J \in (1, 2, \dots, m)$; $PAR(PJ_1, PJ_2, \dots, PJ_s)$ – множина s параметрів PJ j -гілки; AEJ – активний елемент в J -гілці; $VECOMJ$ – коментар щодо технологічної функції J -гілки.

Таблиця 1.1 - Кодування графу

AKI	EKK	QJ	$PAR(PJ_1, PJ_2, \dots, PJ_s)$	AEJ	$VECOMJ$
-------	-------	------	--------------------------------	-------	----------

Модель аеродинамічних процесів в j -гілці МДОРП описується рівняннями [1]

$$\begin{cases} -\frac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t} + r_j(\xi_r, t) Q_j^2 \\ -\frac{\partial P_j}{\partial t} = \frac{\rho a^2}{F_j} \frac{\partial Q_j}{\partial \xi}, \end{cases} \quad (1.1)$$

в яких: P_j , Q_j – тиск і потік повітря вздовж координати ξ , яка обчислюється від початкового (АКІ) до кінцевого (ЕКК) вузла; r_j – питомий аеродинамічний опір; F_j – площа поперечного перетину повітроводу; ρ – щільність повітря; a – швидкість руху звуку в повітрі; $r_j(\xi_r, t)$ – регульований опір; ξ_r – просторова координата регульованого опору.

Граничними умовами для (1.1) є функції друку P_{AKI} , P_{EKK} в вузлах j -гілки. Будемо розрізняти три види гілок та вузлів з позиції граничних умов:

1) гілки, що інцидентні внутрішнім n_1 МДО-вузлам; в них значення тисків обчислюються в процесі ров'язання МДОРП-рівнянь відповідно до динамічних вузлових умов:

$$-\frac{\partial P_{wi}}{\partial t} = \frac{\rho a^2}{F_{wi}} \frac{\partial Q_{wi}}{\partial \xi} \quad (1.2)$$

Тут P_{wi} – тиск в WI-вузлі; Q_{wi} – загальний потік повітря через WI-Вузол; F_{wi} – площа поперечного перетину вузлового простору;

2) гілки, що інцидентні n_2 вузлам підключення активних елементів (АЕJ); наприклад, тут задаються тиски як характеристики вентиляторів

$$P_{wi} = P_{AEJ}(Q_j); \quad (1.3)$$

3) гілки, що інцидентні n_3 вузлам підключення до атмосфери:

$$P_{wi} = P_{ATM} = const \quad (1.4)$$

МДОРПІ має загалом

$$n = n_1 + n_2 + n_3 \quad (1.5)$$

вузлів і відповідно n граничних умов.

Початковими умовами є значення шуканих змінних вздовж координати ξ :

$$P_j(\xi, 0), Q_j(\xi, 0) \quad (j = 1, 2, \dots, m) \quad (1.6)$$

Апроксимуючи рівняння (1.1) за методом прямих з кроком $\Delta\xi$, отримаємо для k -елемента j -гілки систему рівнянь :

$$\begin{aligned} -\frac{P_{jk} - P_{j,k+1}}{\Delta\xi_{jk}} &= \frac{\rho}{F_{jk}} \frac{dQ_{jk}}{dt} + r_{jk} Q_{jk} |Q_{jk}| + r_{jk}(\xi_p, t) Q_{jk} |Q_{jk}|, \\ -\frac{Q_{jk} - Q_{j,k+1}}{\Delta\xi_{jk}} &= \frac{F_{jk}}{\rho a^2} \frac{dP_{j,k+1}}{dt}. \end{aligned} \quad (1.7)$$

Внутрішні граничні умови типу (1.2) апроксимуються рівнянням

$$-\frac{dP_{wi}}{dt} = \frac{\rho a^2}{F_{wi}} \frac{Q_{jk} - \sum_{jwi} (Q_{jwi1} - Q_{jwiMj})}{\Delta\xi_{jk}}, \quad (1.8)$$

в якому позначено: $P_{wi} = P_{jM_{j+1}}$ – тиск в кінцевому вузлі останнього елемента $Q_{jk} = Q_{jMj}$ потоку j -гілки, який притікає до wi -вузла; $jwiMj$ – номери гілок із множини $j=1, 2, \dots, m$, які інцидентні вузлу wi ; при цьому $jwi1$ є першим елементом j -гілки з початковим вузлом wi (витік), в той час як $jwiMj$ є елемент з wi як кінцевим вузлом (притік).

Кожна гілка розбивається в результаті апроксимації на M_j елементів $Q_{j1}, \dots, Q_{jM_j}$. При цьому проводиться нумерація тисків як $P_{j1}, P_{j2}, \dots, P_{jM_j+1}$. Важливо нагадати, що j -гілка має початковий вузол w_i з тиском $P_{w_i} = P_{j1}$ і кінцевий вузол w_i+b ($b = \text{const}$) з тиском $P_{w_i+b} = P_{jM_j+1}$.

Таким чином, кожна гілка МДОРП презентується в результаті апроксимації двома векторами Q_j, P_j ($j = 1 \dots m$):

$$Q_j = (Q_{j1}, Q_{j2}, \dots, Q_{jM_j})^T \quad (1.9)$$

є потоком повітря в j -гілці, а

$$P_j = (P_{j1}, P_{j2}, \dots, P_{jM_j+1})^T \quad (1.10)$$

це тиск в j -гілці. При цьому кількість елементів в гілках обчислюється в залежності від довжини l_j і єдиного для МДОРП кроку апроксимації $\Delta\xi$:

$$M_j = l_j / \Delta\xi \quad (1.11)$$

Дискретизована в просторі модель j -гілки містить M_j систем рівнянь (1.7) при $k=1,2,\dots,M_j$. Для числових розрахунків компонентів векторів (1.9) і (1.10) рівняння (1.7) приводяться до форми дискретизованої відносно просторової координати Simulation-моделі:

$$\begin{cases} \dot{Q}_{jk} = \alpha_j (P_{jk} - P_{j,k+1}) - \beta_j Q_{jk} |Q_{jk}| - \beta_{rj} Q_{jk} |Q_{jk}| \\ \dot{P}_{j,k+1} = g_j (Q_{jk} - Q_{j,k+1}) \end{cases} \quad (1.12)$$

Коефіцієнти $\alpha_j, \beta_j, \beta_{rj}, g_j$ залежать від аеродинамічних параметрів j -гілки.

При розробці просторово-дискретизованої Simulation-моделі всього МДОРП треба взяти до уваги m систем рівнянь типу (1.12) для всіх гілок, тобто $j=1,2,..., m$:

$$\begin{cases} \dot{Q}_{1k} = \alpha_1(P_{1k} - P_{1k+1}) - \beta_1 Q_{1k} |Q_{1k}| - \beta_{r1} Q_{1k} |Q_{1k}| \\ \dot{P}_{1k+1} = g_1(Q_{1k} - Q_{1k+1}) \end{cases} \quad k=1,2,...,M_1$$

$$\dots\dots\dots$$

$$\begin{cases} \dot{Q}_{mk} = \alpha_m(P_{mk} - P_{mk+1}) - \beta_m Q_{mk} |Q_{mk}| - \beta_{rm} Q_{mk} |Q_{mk}| \\ \dot{P}_{mk+1} = g_m(Q_{mk} - Q_{mk+1}) \end{cases} \quad k=1,2,...,M_m \quad (1.13)$$

Повну систему МДОРП-рівнянь в формі Simulation-моделі можна записати так:

$$\begin{cases} \dot{Q}_{11} = \alpha_1(P_{11} - P_{12}) - \beta_1 Q_{11} |Q_{11}| - \beta_{r1} Q_{11} |Q_{11}| \\ \dot{P}_{12} = g_1(Q_{11} - Q_{12}) \end{cases} \dots \begin{cases} \dot{Q}_{1M_1} = \alpha_1(P_{1M_1} - P_{1M_1+1}) - \beta_1 Q_{1M_1} |Q_{1M_1}| - \beta_{r1} Q_{1M_1} |Q_{1M_1}| \\ \dot{P}_{1M_1+1} = g_1(Q_{1M_1} - Q_{1M_1+1}) \end{cases}$$

$$j=1, \quad k=1, \dots, M_1 \quad (1.14)$$

.....

$$\begin{cases} \dot{Q}_{m1} = \alpha_m(P_{m1} - P_{m2}) - \beta_m Q_{m1} |Q_{m1}| - \beta_{rm} Q_{m1} |Q_{m1}| \\ \dot{P}_{m2} = g_m(Q_{m1} - Q_{m2}) \end{cases} \begin{cases} \dot{Q}_{mk} = \alpha_m(P_{mk} - P_{mk+1}) - \beta_m Q_{mk} |Q_{mk}| - \beta_{rm} Q_{mk} |Q_{mk}| \\ \dot{P}_{mk+1} = g_m(Q_{mk} - Q_{mk+1}) \end{cases}$$

$$j=m, \quad k=1, \dots, M_m$$

МДОРП має $n = n_1 + n_2 + n_3$ граничних умов (1.2)/(1.8), (1.3), (1.4). Відповідно рівнянню (1.8) сформулюємо n_1 граничних умов для внутрішніх вузлів ($w_i=1, 2, \dots, n_1$):

$$\frac{dP_{w1}}{dt} = \frac{\rho a^2}{F_{w1}} \frac{Q_{jk} - \sum_{jw1} (Q_{jw11} - Q_{jw1Mj})}{\Delta \xi_{jk}} \quad (1.15)$$

.....

$$\frac{dP_{wn1}}{dt} = \frac{\rho a^2}{F_{wn1}} \frac{Q_{jk} - \sum_{jwn1} (Q_{jwn11} - Q_{jwn1Mj})}{\Delta \xi_{jk}}$$

Характеристики вентиляторів створюють по (1.3) ще n_2 граничні умови:

$$\begin{aligned} Pwi &= P1AEJ(QJ) \\ &..... \\ Pwi &= Pn2AEJ(QJ). \end{aligned} \quad (1.16)$$

При цьому $wi=n_1+1, n_1+2, \dots, n_1+n_2$, якщо нумерація вузлів проводиться послідовно з внутрішніми вузлами. Активні елементи (вентилятори) нумеруються в (1.16) від 1 до n_2 додатково.

Для n_3 вузлів підключення до атмосфери діють граничні умови згідно з (1.4):

$$\begin{aligned} Pwi &= P(n1 + n2) + 1 = P1ATM = const, \\ &..... \\ Pwi &= Pn = Pn3ATM = const. \end{aligned} \quad (1.17)$$

При такому нумеруванні маємо в (1.17) $wi=(n_1+n_2)+1, \dots, (n_1+n_2)+(n_3-1)$, n .

Рівняння (1.14) створюють разом з граничними умовами (1.15), (1.16), (1.17) просторово-дискретизовану МДОРП-Simulation-модель. З огляду на розмірності промислових МДОРП ($m \geq 1000$, $n \geq 300$, $M_j=l_j/\Delta \xi = 5 \dots 50$) актуальною є розробка засобів генерування цих Simulation-моделей.

1.1 Структура проблемно орієнтованого РПМС

За функціональним призначенням ПОПМС – це сполучна ланка між користувачем (розробником моделей) і операційною системою деякої апаратної платформи (рис. 1.2).



Рисунок - 1.2 Роль середовища моделювання

Загалом ПОПМС повинне якомога повно задовольняти вимоги, що орієнтовані на дружність до кінцевого користувача, та бути незалежним від різних платформ і мов реалізації. На цій підставі в роботах кафедри КІ сформульовано наступне визначення:

Проблемно орієнтоване паралельне моделююче середовище (ПОПМС) – це системна організація на основі нових інформаційних технологій спільного функціонування паралельних апаратних ресурсів, загальносистемного та цілеспрямовано розробленого моделюючого програмного забезпечення з врахуванням особливостей задач певної предметної області, що підтримують всі етапи математичного моделювання складних динамічних систем, оптимізації та ідентифікації параметрів паралельних моделей і симуляторів.

Структуру ПОПМС наведено на рис. 1.3. До апаратних засобів ПМС відносяться комп'ютерні системи MIMD-архітектур і персональні комп'ютери, які об'єднані в паралельно функціонуючу групу, звану кластером, за MIMD-принципом. Доступ до апаратних ресурсів реалізовано за допомогою мережових технологій з робочих місць, які представлені потужними персональними комп'ютерами. При цьому периферійні пристрої виконують візуалізацію і документування на різних етапах моделювання. Веб-сервер необхідний для підтримки веб-базованого підходу до паралельного моделювання в ПОПМС.

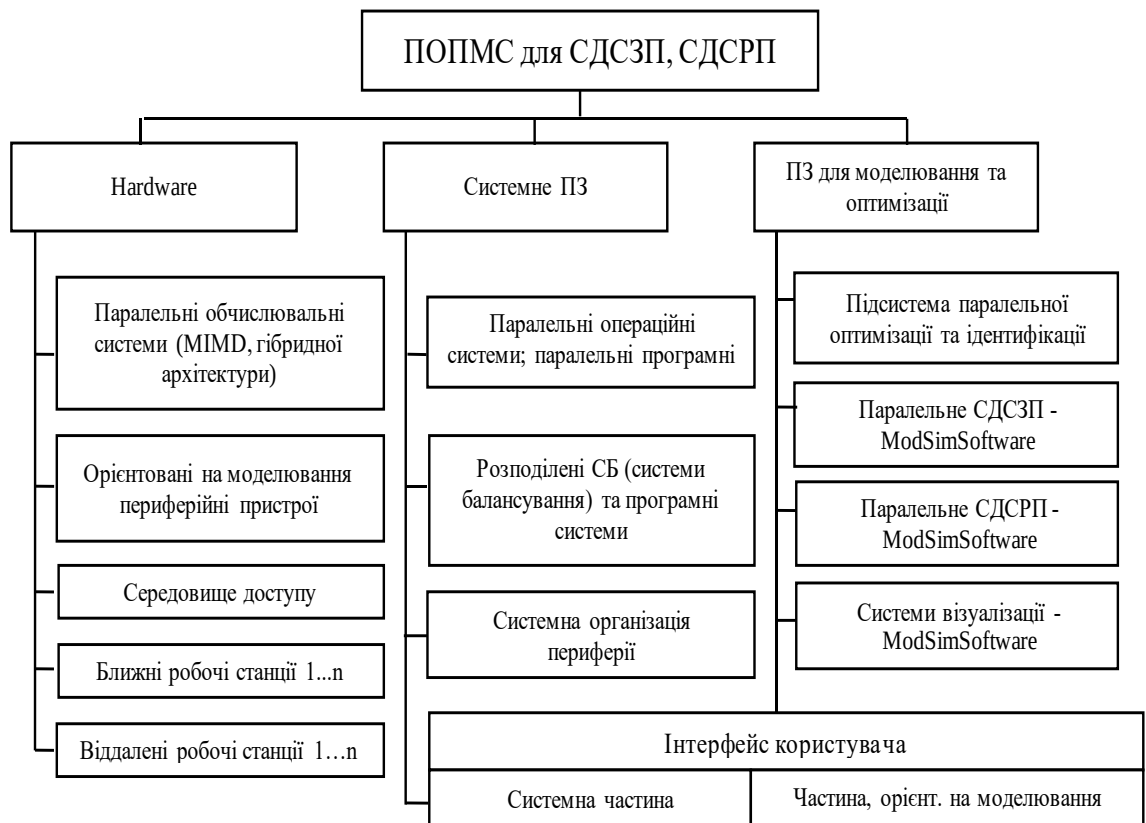


Рисунок 1.3 – Системна організація ПОПМС

Структуру апаратної реалізації ПОПМС наведено на рис. 1.4 [8].



Рисунок 1.4 – Апаратне забезпечення ПОПМС

В основі даної структури є організація віддаленого веб-базованого підходу до моделювання. Відзначимо, що на відміну від стандартного підходу, тут моделююча програма запускається на виконання не на веб-сервері, а на супер-EOM MIMD-архітектури або кластері. У даній структурі веб-сервер з'єднаний безпосередньо з сервером бази даних, що дозволяє користувачеві зберігати параметри моделей і самі моделі в базі даних. При цьому маєтись на увазі, що клієнт має локальний доступ до периферійних пристроїв для візуалізації та документування результатів моделювання.

1.2 Принцип декомпозиції РПМС на підсистеми.

Згідно з принципом декомпозиції розподілених паралельних моделюючих середовищ в [5, 8] запропоновано наступну повнофункціональну множину підсистем:

1) Підсистема діалогу з користувачем, яка відповідає за демонстрацію можливостей РПМС, навчання користувача, планування та супроводження експериментальних досліджень. Також вона забезпечує зв'язок між різними компонентами системи.

2) Підсистема топологічного аналізу, яка необхідна для словесного та графічного представлення різноманітних топологій, їх кодування у форми, на основі яких далі складаються рівняння, що описують певний об'єкт моделювання.

3) Підсистема генерування рівнянь – вона безпосередньо працює з підсистемою топологічного аналізу, займається перетворенням початкових рівнянь у матричні форми, генерує пристосовані для певних обчислювальних методів дискретні ДСРП-, ДСЗП-Simulation-моделі.

4) Підсистема віртуальних паралельних Simulation-моделей призначена для інтерактивного їх представлення в залежності від обраного рівня розпаралелювання. Дана підсистема також проводить апріорний аналіз для кожного з цих рівнів, дозволяє виявити найприйнятніший варіант розпаралелювання ще до початку самого процесу розробки симуляторів і паралельного моделювання.

5) Підсистема вирішувачів рівнянь, яка повинна відповідати за вирішення систем алгебраїчних, звичайних диференціальних та диференціальних в частинних похідних рівнянь за допомогою найоптимальнішого для певної задачі обчислювального методу.

6) Підсистема обміну даними – важлива підсистема, що відповідає за обмін даними, яких може бути дуже велика кількість, між різними складовими РПМС.

7) Підсистема балансування завантаження, яка повинна робити оцінку об'ємів завантаження ресурсів та, в залежності від цього, корегувати роботу РПМС. Вона також має робити статичне балансування завантаження процесорів на всіх можливих рівнях розпаралелювання.

8) Підсистема візуалізації – збирає, обробляє та наочно представляє топології об'єктів моделювання, їх моделі, Simulation-моделі, симулятори, плани проведення модельних експериментів та результати моделювання.

9) Підсистема баз даних, яка є інтегрованою зі всіма підсистемами та відповідає за збереження та архівацію даних РПМС.

10) Підсистема ІТ-підтримки – забезпечує віддалений доступ до надпотужних обчислювальних систем.

На рис.1.5 наведено компоненти моделюючого програмного забезпечення ПОРПМС. Відзначимо, що тут модельно-орієнтована частина підсистеми діалогу є компонентою моделюючого програмного забезпечення. Підсистема діалогу істотно полегшує працю експертів предметних областей, забезпечуючи діалогову комп'ютерну підтримку всіх етапів побудови і застосування паралельних моделей і симуляторів. Інтеграція ТА, ГР, ВР з координацією з боку підсистеми діалогу утворює паралельний симулятор ПОРПМС.

Розробкам підсистеми вирішувачів рівнянь як апаратно-програмної компоненти ПОРПМС не приділено достатньої уваги, тому вони залишаються актуальною складовою ParSimTech-проблематики, а певна частина цих задач має вирішуватись у даній роботі.

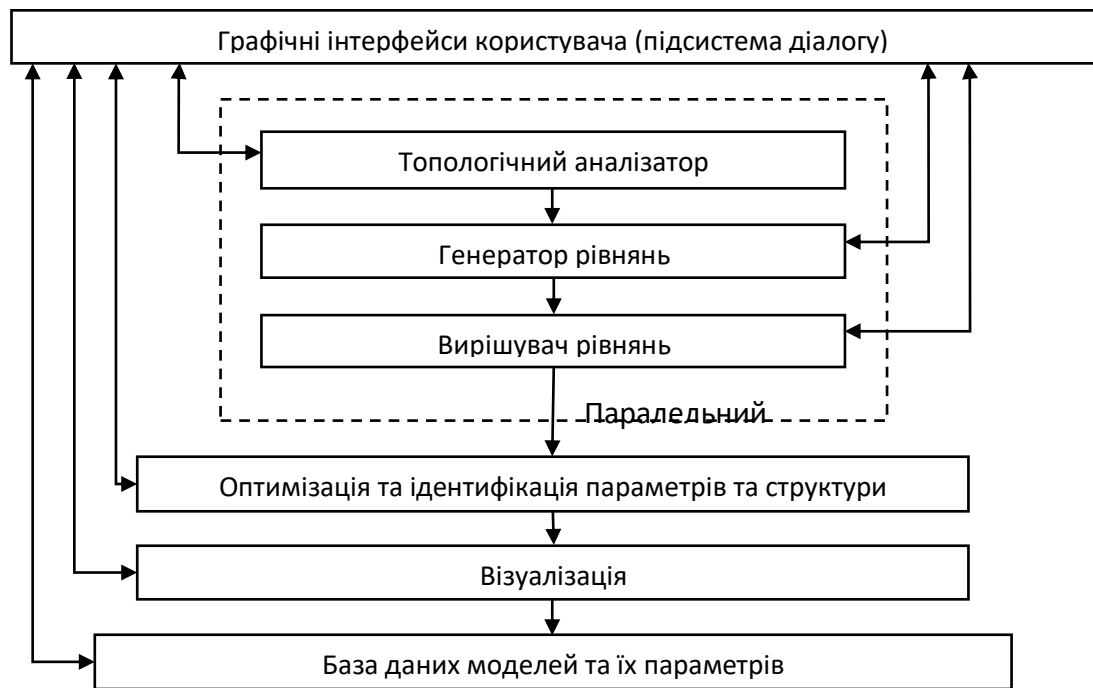


Рисунок 1.5 – Компоненти моделюючого програмного забезпечення

1.3 Задачі розробок і досліджень

Проведений вище короткий аналіз робіт, що близькі до теми даної магістерської роботи, дозволяє сформулювати наступні актуальні задачі розробок і досліджень:

- 1) Дати технологічну характеристику, розглянути математичну модель і Simulation-модель мережевого динамічного об'єкту з розподіленими параметрами (МДОРП), на який орієнтовано РПМС.
- 2) Визначити функціональність підсистеми вирішувачів рівнянь ПОРПМС.
- 3) Розробити структуру підсистеми.
- 4) Запропонувати генератори дискретних Simulation-моделей і послідовні вирішувачів рівнянь гілок МДОРП.
- 5) Розробити алгоритм функціонування паралельного вирішувача на рівні «Гілка МДОРП – MIMD-процес».
- 6) Виконати експерименти і запропонувати використання результатів розробок в ParSimTech-центрі ДонНТУ.

2 ФУНКЦІОНАЛЬНІСТЬ І СТРУКТУРА ПІДСИСТЕМИ ВИРІШУВАЧІВ РІВНЯНЬ

2.1 Визначення підсистеми, її основні функції та структура

Розглянемо вирішувач рівнянь (ВР), який являє собою програму, що реалізовує алгоритм чисельного рішення матрично- векторної системи рівнянь та розрахункових формул дискретних Simulation-моделей (ДСМ), сформованих генератором рівнянь. Грунтуючись на вимогах до середовища моделювання, алгоритми ВР будуються таким чином, щоб користувачі мали можливість обрати паралельну реалізацію відомих їм обчислювальних методів. Дані методи функціонують в сучасних мовах моделювання, тому їх застосування в ПОПМС відповідає принципам наступності й сумісності паралельних середовищ з послідовними засобами моделювання МДОЗП, МДОРП. Актуальною є побудова паралельних вирішувачів систем ДСМ-рівнянь на основі нових блокових обчислювальних методів (БОМ), що мають певні якості притаманного їм паралелізму [8]. Структурно місце вирішувача рівнянь в моделюючому ПЗ ПОПМС показано на рис. 2.1.

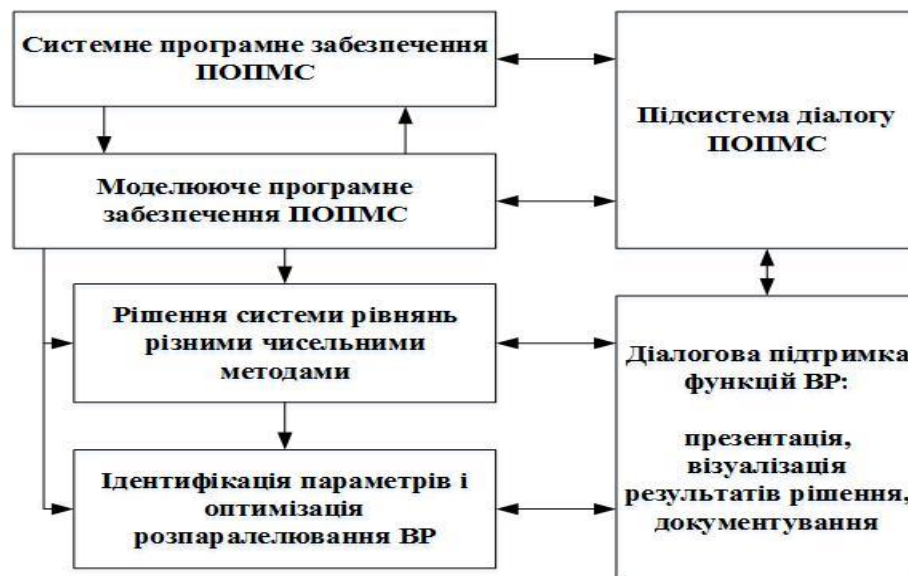


Рисунок 2.1 – Вирішувачі рівнянь у складі моделюючого програмного забезпечення

У лівій верхній частині рисунка 2.1 знаходяться два блоки – системне та моделююче програмне забезпечення РПМС. До системного ПЗ відносяться паралельні операційні системи (системи паралельного програмування), розподілені операційні системи та системи програмування, системна організація периферії. Моделююче ПЗ складається з паралельного програмного забезпечення для ДСЗП та паралельного програмного забезпечення для ДСРП. Блоки програмного забезпечення пов’язані з підсистемою діалогу, причому комунікація може відбуватися як з програмним, так і з моделюючим ПЗ.

Основна задача моделюючого програмного забезпечення полягає в тому, щоб певна система рівнянь, за допомогою якої описується МДОРП, була вирішена. При цьому користувачу повинно бути запропоновано вирішити дану систему якимось певним обчислювальним методом. В ідеальному варіанті програма повинна сама запропонувати найкращий шлях вирішення задачі.

У моделюючому програмному забезпеченні повинні бути реалізовані різноманітні методи розпаралелювання для різних рівнів. Інформація про те, який рівень розпаралелювання більш прийнятний до конкретної задачі, повинна надходити також від підсистеми віртуальних паралельних Simulation-моделей, яка займається апріорним аналізом.

Необхідно зазначити, що хоча основною задачею постає рішення системи рівнянь паралельно, в моделюючому ПЗ повинні існувати і послідовні варіанти алгоритмів, бо користувачем можуть бути поставлені на виконання такі задачі, які більш доцільно було б вирішити послідовно. Послідовні вирішувачі потрібні передусім для порівняння результатів паралельного і послідовного моделювання.

Функції вирішувача рівнянь повинні бути підтриманими з боку підсистеми діалогу шляхом презентації та візуалізації результатів вирішення. Також користувач повинен мати змогу впливати на роботу вирішувача, а саме

обираючи певний обчислювальний метод, задаючи бажану точність рішення, рівень розпаралелювання МДОРП та інше.

Функції взаємодії даної підсистеми з іншими «релевантними» підсистемами зумовлюються структурами РПМС (рис.1.5) та вирішувача рівнянь (рис.2.1). Звідти можна побачити, що ПВР тісно взаємодіє лише з наступними підсистемами:

- підсистема генерування рівнянь (ПГР);
- підсистема візуалізації (ПВ);
- підсистема діалогу (ПД).

Виходячи з цього, визначимо функції взаємодії з цими трьома підсистемами.

Головна функція ПГР – генерування рівнянь та подальша їх передача до ПВР, де вже безпосередньо буде виконуватися їх вирішення певним обчислювальним методом. Дана підсистема тісно співпрацює з підсистемою топологічного аналізу, вона займається представленням початкових рівнянь (моделі) у векторно-матричній формі, формальним перетворенням ДСРП- або ДСЗП-моделі у Simulation-модель. Таким чином, у якості даних, якими будуть обмінюватися ПВР та ПГР, будуть вектори та матриці, що кодують рівняння, якими описується об'єкт моделювання. Передачу даних можна реалізувати через файл.

Підсистема візуалізації – наступна підсистема, з якою «спілкується» ПВР. Після того, як відпрацює вирішувач рівнянь, необхідно представити результати його роботи в зрозумілому для фахівця предметної області вигляді. Цим і займається підсистема візуалізації. У якості даних, які отримує дана підсистема, виступає файл, в якому наведено результати моделювання МДОРП та МДОЗП. У найпростішому вигляді це файл з координатами x (час) та y (значення певного параметру, наприклад, значення тиску або потоку повітря). Тут необхідно зробити зауваження, що розробити підсистему можна таким чином, щоб всі дані були представлені в одному файлі, а можна і так,

щоб кожний параметр знаходився в своєму файлі. Другий варіант є більш зрозумілим та прозорим, але він призводить до більшого навантаження на файлову систему. Також необхідно врахувати той факт, що вихідних даних може бути дуже багато, тобто вихідний файл буде дуже великого розміру, що негативно вплине на швидкість роботи усього ПОРПМС. Тому необхідно розробити алгоритми в ПВР таким чином, щоб розмір вихідного файлу був оптимальним, не занадто великим та таким, щоб отриманих даних було достатньо для зрозумілої візуалізації, тобто такої, яка повно представляє картину, отриману після проведення експерименту.

Остання «релевантна» підсистема – це підсистема діалогу, яка співпрацює зі всіма підсистемами ПОРПМС. Усі комутації, що відбуваються між підсистемами, повинні проходити через неї. Через підсистему діалогу користувач може отримати доступ до конкретної підсистеми. Якщо розглядати ПВР, то користувач, наприклад, безпосередньо може звернутися до неї та запустити необхідну програму. Також через дану підсистему можна змінювати параметри моделювання (час моделювання, точність та інші).

2.2 Формальні підходи до побудови дискретних Simulation-моделей МДОРП, обчислювальні методи

Рівняння (1.14) створюють разом з граничними умовами (1.15), (1.16), (1.17) дискретизовану за просторовою координатою МДОРП-Simulation-модель. З огляду на розмірності промислових МДОРП ($m \geq 1000$, $n \geq 300$, $M_j = l_j / \Delta \xi = 5 \dots 50$) актуальною є розробка формальних засобів генерування цих Simulation-моделей та дискретних Simulation-моделей (ДСМ), що зумовлюються обраними обчислювальними методами. Саме рівняння ДСМ є головними вхідними об'єктами підсистеми вирішувачів. Розглянемо формальні підходи до побудови ДСМ МДОРП.

Граничні умови (ГУ) для рівнянь МДОРП у частинних похідних

$$\begin{cases} -\frac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t} + r_j(\xi_r, t) Q_j^2 \\ -\frac{\partial P_j}{\partial t} = \frac{\rho a^2}{F_j} \frac{\partial Q_j}{\partial \xi}, \end{cases} \quad (2.1)$$

мають три наступних види [1, 2] :

ГУ-1. Гілки, інцидентні внутрішнім n_1 вузлам; значення вузлового тиску розраховуються відповідно до динамічних умов вузла під час вирішення системи рівнянь:

$$-\frac{\partial P_{wi}}{\partial t} = \frac{\rho a^2}{F_{wi}} \frac{\partial Q_{wi}}{\partial x} \quad (2.2)$$

тут P_{wi} – тиск у вузлі wi ; Q_{wi} – сумарний потік повітря через вузол wi ; F_{wi} – площа поперечного перерізу вузлового простору.

Внутрішні граничні умови типу (2.2) апроксимуємо звичайними диференціальними рівняннями:

$$-\frac{dP_{wi}}{dt} = \frac{\rho a^2}{F_{wi}} \frac{Q_{jk} - \sum_{jwi} (Q_{jwi1} - Q_{jwiM_j})}{\Delta \xi_{jk}} \quad (2.3)$$

Тут: $P_{wi} = P_{jM_{j+1}}$ – тиск у кінцевому вузлі останнього елемента $Q_{jk} = Q_{jM_j}$ струму j -гілки, що притікає в wi -вузол; $jwiM_j$ – номери гілок з множини $j=1, 2, \dots, m$, інцидентні вузлу wi ; де $jwi1$ є першим елементом j -гілки з wi як початковим вузлом (відтік), тоді як $jwiM_j$ є останнім елементом з wi як кінцевим вузлом (вхід).

Загальний потік повітря Q_{wi} через вузол wi ($i=1, 2, \dots, n_1$) обчислюється в (2.3) як алгебраїчна сума потоків повітря, що втікає та витікає. Вектор

$$\bar{Q}_w = (Q_{w1}, Q_{w2}, \dots, Q_{wn1})^T \quad (2.4)$$

визначається вузловим векторно-матричним рівнянням

$$\bar{Q}_w = A * \bar{Q} \quad (2.5)$$

де A – матриця інцидентності графа МДОРП. Векторне матричне представлення граничних умов (2.2) для МДОРП таке:

$$-\frac{\partial \bar{P}_w}{\partial t} = \text{Dia} \left(\frac{\rho a^2}{F_{wi}} \right) * \frac{\partial \bar{Q}_w}{\partial x} \quad (2.6)$$

підставимо (2.5) у (2.6):

$$-\frac{\partial \bar{P}_w}{\partial t} = \text{Dia} \left(\frac{\rho a^2}{F_{wi}} \right) * \frac{\partial (A * \bar{Q})}{\partial x} \quad (2.7)$$

Рівняння (2.7) описує динамічні стани вузлів і служить тестом для перевірки стаціонарних станів: при $-\frac{\partial \bar{P}_w}{\partial t} = 0$ ми отримуємо алгебраїчну нульову суму потоків повітря у вузлах, що відповідає рівнянню (2.5).

Загалом ми позначаємо:

$$Q_{wi} = \sum_{w_i} Q_{zu} - \sum_{w_i} Q_{aus} \quad (2.8)$$

і отримаємо наближене рівняння (2.2) з урахуванням заданих напрямків струму:

$$\frac{dP_{wi}}{dt} = \frac{\rho a^2}{F_{wi}} \frac{(\sum_{w_i} Q_{zu} - \sum_{w_i} Q_{aus})}{\Delta x}$$

або

$$\frac{dP_{wi}}{dt} = \gamma_{wi} (\sum_{w_i} Q_{zu} - \sum_{w_i} Q_{aus}) \quad (2.9)$$

з коефіцієнтами $\gamma_{wi} = \frac{\rho a^2}{F_{wi} \Delta x}$, де $i = 1, 2, \dots, n_1$.

ГУ-2. Гілки, що інцидентні n_2 вузлам підключення активних елементів (АЕJ); наприклад, тут задаються тиски як характеристики вентиляторів

$$P_{wi} = P_{AEJ}(QJ); \quad (2.10)$$

ГУ-3. Гілки, що інцидентні n_3 вузлам підключення до атмосфери:

$$P_{wi} = P_{ATM} = const. \quad (2.11)$$

МДОРП має загалом $n = n_1 + n_2 + n_3$ вузлів і відповідно n граничних умов. В МДОРП-рівняннях індекси повинні мати такі значення: $1 \leq j \leq m$; $1 \leq k \leq M_j$; $1 \leq w_i \leq n_1$; $1 \leq w_{vi} \leq n_2$; $1 \leq w_{Ai} \leq n_3$. Загалом МДОРП має вектори граничних умов:

$$\begin{aligned} \overline{P_w} &= (P_{w1}, P_{w2}, \dots, P_{wi}, \dots, P_{wn_1})^T \\ \overline{P_{wv}} &= (P_{wv1}, P_{wv2}, \dots, P_{wvn_2})^T \\ \overline{P_{wA}} &= (P_{wA1}, \dots, P_{wAn_3})^T \end{aligned} \quad (2.12)$$

Розглянемо зображення векторів:

$$\begin{aligned} \overline{Q} &= (Q_1, Q_2, \dots, Q_k, \dots, Q_m)^T, \\ \overline{P} &= (P_1, P_2, \dots, P_k, \dots, P_m)^T, \end{aligned}$$

елементи яких перетворюються на рядки типу $Q_{j1} \ Q_{j2} \ \dots \ Q_{jk} \ \dots \ Q_{jM_j}$ і $P_{j1} \ P_{j2} \ \dots \ P_{jk} \ \dots \ P_{jM_j}$. Тут $M_j = l_j / \Delta_X$ визначає розміри рядків. Через різну довжину l_j гілок M_j різні. Апроксимовані вектори $\overline{Q}$ і $\overline{P}$ пропонується визначати як матричні структури з m рядків і $M_{j\max}$ стовпців:

Тепер ми можемо представити вектор потоків через вузли за допомогою елементів (2.8) таким чином

$$\bar{Q}_w = \begin{pmatrix} Q_{w1} \\ Q_{w2} \\ \vdots \\ Q_{wn1} \end{pmatrix} = A_{um} * \begin{pmatrix} Q_{11} \\ Q_{1M1} \\ Q_{21} \\ Q_{2M2} \\ \vdots \\ Q_{m1} \\ Q_{mMm} \end{pmatrix} \quad (2.15)$$

Для подальших міркувань введемо нове позначення для вектора Q з елементами Q_{j1} і Q_{jMj} :

$$\bar{Q}_{um} = (Q_{11}Q_{1M1}, Q_{21}Q_{2M2}, \dots, Q_{m1}Q_{mMm})^T$$

Тоді використаємо (2.15) у такому вигляді:

$$\bar{Q}_w = A_{um} * \bar{Q}_{um} \quad (2.16)$$

Вектор $\bar{P}_w$ значень тиску у внутрішніх вузлах при врахуванні (2.16) визначається рівнянням

$$\frac{d\bar{P}_w}{dt} = Dia(\gamma_w) * A_{um} * \bar{Q}_{um}$$

або

$$\frac{d\bar{P}_w}{dt} = Dia(\gamma_w) * \bar{Q}_w \quad (2.17)$$

Елементи $\overline{P_{wA}}$ і $\overline{P_{wV}}$ відомі з таблиці кодування топології, тому ми можемо сформулювати початкові та кінцеві значення тиску для j-гілок за допомогою операцій призначення та отримати наступні вектори:

$$\overline{P_{Jo}} = \begin{pmatrix} P_{10} = P_{wA} \\ P_{20} = P_{w1} \\ P_{30} = P_{w2} \\ P_{40} = P_{w2} \\ P_{50} = P_{w1} \\ \dots \dots \\ P_{m0} = P_{wn1} \end{pmatrix} \text{ – для початкових вузлів,} \quad (2.18)$$

$$\overline{P_{JMj}} = \begin{pmatrix} P_{1M1} = P_{w1} \\ P_{2M2} = P_{w2} \\ P_{3M3} = P_{w3} \\ P_{4M4} = P_{w3} \\ P_{5M5} = P_{w4} \\ \dots \dots \\ P_{mMm} = P_{wV} \end{pmatrix} \text{ – для кінцевих вузлів} \quad (2.19)$$

В принципі, рівняння (2.1) описують весь МДОРП для $1 \leq j \leq m$ і $1 \leq k \leq Mj$.
Для гілки $j=1$:

$$\left. \begin{aligned} \frac{dQ_{11}}{dt} &= \alpha_1(P_{10} - P_{11}) - \beta_1 Q_{11}|Q_{11}| \\ \frac{dP_{11}}{dt} &= \gamma_1(Q_{11} - Q_{12}) \end{aligned} \right\} \dots\dots\dots (2.20)$$

$$\begin{aligned} \frac{dQ_{1M1}}{dt} &= \alpha_1(P_{1M1-1} - P_{1M1}) - \beta_1 Q_{1M1}|Q_{1M1}| \\ \frac{dP_{1M1-1}}{dt} &= \gamma_1(Q_{1M1-1} - Q_{1M1}) \\ \frac{dP_{1M1}}{dt} &= \gamma_{w1}(Q_{w1}) \end{aligned}$$

Розглянемо «підвектор» Q_1 як

$$\bar{Q}_1 = (Q_{11} \ Q_{12} \ \dots \ Q_{1k}, \dots, Q_{1M1})^T$$

і підвектор $\bar{Z}_1$ як

$$\bar{Z}_1 = (Z_{11} \ \dots \ Z_{1k} \ \dots \ Z_{1M1})^T = (q_{11}|q_{11}| \ \dots \ q_{1k}|q_{1k}|, \dots, q_{1M1}|q_{1M1}|)^T$$

Введемо різницеві вектори для цієї гілки:

$$\begin{aligned}\overline{DP}_1 &= [(P_{10} - P_{11}) \ \dots \ (P_{1M1-1} - P_{1M1})]^T \\ \overline{DQ}_1 &= [(Q_{11} - Q_{12}) \ \dots \ (Q_{1M1-1} - Q_{1M1})]^T\end{aligned}$$

Ці вектори дозволяють нам записати рівняння (2.20) для гілки $j=1$ у вигляді:

$$\left. \begin{aligned}\frac{d\bar{Q}_1}{dt} &= \alpha_1 * \overline{DP}_1 - \beta_1 * \bar{Z}_1 \\ \frac{d\bar{P}_1}{dt} &= \gamma_1 * \overline{DQ}_1\end{aligned}\right\} \quad (2.21)$$

Граничні умови для гілки Q_1 – це тиски $P_{10} = P_{wA}$, $P_{1M1} = P_{w1}$, де діє рівняння

$$\frac{dP_{w1}}{dt} = \gamma_{w1}(Q_{w1}).$$

Тут $\alpha_1, \beta_1, \gamma_1$ константи, які множаться на вектори $\overline{DP}_1, \overline{DQ}_1, \bar{Z}_1$. Векторно-матричне представлення рівнянь у вигляді Simulation-моделі отримується з (2.21) при $1 \leq j \leq m$ та векторів граничних умов (2.12)

$$\begin{aligned}\overline{P}_w &= (P_{w1} \ P_{w2} \ \dots \ P_{wi} \ \dots \ P_{wn_1})^T \\ \overline{P}_{wv} &= (P_{wv1} \ P_{wv2} \ \dots \ P_{wvn_2})^T \\ \overline{P}_{wA} &= (P_{wA1} \ \dots \ P_{wAn_3})^T\end{aligned}$$

з урахуванням (2.18) $\overline{P}_{jo}$ та (2.19) $\overline{P}_{jMj}$:

$$\left. \begin{aligned} \frac{d\bar{Q}_j}{dt} &= \alpha_j * \overline{DP_j} - \beta_j * \bar{Z}_j \\ \frac{d\bar{P}_j}{dt} &= \gamma_j * \overline{DQ_j} \end{aligned} \right\} \quad (2.22)$$

На основі цього представлення Simulation-моделі необхідно розробити дискретні Simulation-моделі (DSM), які залежать від обраних чисельних методів. На основі методу Ейлера DSM матиме такий вигляд:

$$\bar{Q}_j(i) = \bar{Q}_j(i-1) + h * [\alpha_j * \overline{DP_j}(i-1) - \beta_j * \bar{Z}_j(i-1)] \quad (2.23)$$

$$\bar{P}_j(i) = \bar{P}_j(i-1) + h * [\gamma_j * \overline{DQ_j}(i-1)] \quad (2.24)$$

Метод Адамса-Бешфорта дає наступну DSM:

$$\bar{Q}_j(i) = \bar{Q}_j(i-1) + h * [\alpha_j * \overline{DP_j}(i-1) - \beta_j * \bar{Z}_j(i-1)] \text{ для } i=1$$

$$\begin{aligned} \bar{Q}_j(i) &= \bar{Q}_j(i-1) + 1.5h * [\alpha_j * \overline{DP_j}(i-1) - \beta_j * \bar{Z}_j(i-1)] - \\ &- 0.5h * [\alpha_j * \overline{DP_j}(i-2) - \beta_j * \bar{Z}_j(i-2)] \text{ для } i>1 \end{aligned} \quad (2.25)$$

$$\bar{P}_j(i) = \bar{P}_j(i-1) + h * [\gamma_j * \overline{DQ_j}(i-1)] \text{ для } i=1$$

$$\begin{aligned} \bar{P}_j(i) &= \bar{P}_j(i-1) + 1.5h * [\gamma_j * \overline{DQ_j}(i-1)] - \\ &- 0.5h * [\gamma_j * \overline{DQ_j}(i-2)] \text{ для } i>1 \end{aligned} \quad (2.26)$$

Розрахункові формули методу Рунге-Кутта виглядають так:

$$Q(i+1) = Q(i) + \frac{1}{6}K_1 + \frac{1}{3}(1 - \sqrt{\frac{1}{2}})K_2 + \frac{1}{3}(1 + \sqrt{\frac{1}{2}})K_3 + \frac{1}{6}K_4, \quad (2.27)$$

де коефіцієнти обчислюються за формулами

$$\begin{cases} K_1 = h * [F(i)] \\ K_2 = h * \left[F(i) + \frac{1}{2} K_1 \right] \\ K_3 = h * \left[F(i) - (0.5 - \sqrt{0.5}) K_1 + (1 - \sqrt{0.5}) K_2 \right] \\ K_4 = h * \left[F(i) - \sqrt{0.5} K_2 + (1 + \sqrt{0.5}) K_3 \right] \end{cases}$$

а $F(i)$ є правою частиною диференціального рівняння. Введемо позначення правих частин векторних матричних рівнянь (2.22):

$$\begin{aligned} FQ &= \alpha_j * \overline{DP_j} - \beta_j * \bar{Z}_j \\ FP &= \gamma_j * \overline{DQ_j} \end{aligned}$$

Відповідно до формули (2.27) вектори $\bar{Q}_j$ і $\bar{P}_j$ обчислюються так:

$$\begin{aligned} \bar{Q}_j(i) = \bar{Q}_j(i-1) + \frac{1}{6} K1 + \frac{1}{3} (1 - Kor0.5) K2 + \frac{1}{3} (1 + Kor0.5) K3 + \\ \frac{1}{6} K4 \end{aligned} \quad (2.28)$$

$$\begin{aligned} K1 &= h * [FQ(i-1)] \\ K2 &= h * [FQ(i-1) + 0.5 K1] \\ K3 &= h * [FQ(i-1) - (0.5 - Kor0.5) K1 + (1 - Kor0.5) K2] \\ K4 &= h * [FQ(i-1) - (Kor0.5) K2 + (1 + Kor0.5) K3] \\ \bar{P}_j(i) &= \bar{P}_j(i-1) + \frac{1}{6} K1 + \frac{1}{3} (1 - Kor0.5) K2 + \frac{1}{3} (1 + Kor0.5) K3 + \\ &\quad \frac{1}{6} K4 \end{aligned} \quad (2.29)$$

$$\begin{aligned} K1 &= h * [FP(i-1)] \\ K2 &= h * [FP(i-1) + 0.5 K1] \\ K3 &= h * [FP(i-1) - (0.5 - Kor0.5) K1 + (1 - Kor0.5) K2] \\ K4 &= h * [FP(i-1) - (Kor0.5) K2 + (1 + Kor0.5) K3] \end{aligned} \quad (2.30)$$

2.3 Рівні розпаралелювання Simulation-моделей

Для розробки паралельного вирішувача рівнянь дамо наступні визначення.

Визначення 1. Віртуальною паралельною моделлю (ВПМ) називатимемо абстракцію, яка складається з підготовлених генератором рівнянь систем рівнянь та розробленої відповідно до одного з підходів до розпаралелювання структурою MIMD-процесів для паралельного розв'язання цих систем рівнянь.

При MIMD-рівні (рівні процедур або функцій) розпаралелювання різні розділи однієї й тієї ж програми повинні виконуватися паралельно. Ці розділи називають "процесами" [5]. Таким чином, віртуальний MIMD-процес – це відносно автономна програма, що відповідає частині алгоритму розв'язання системи рівнянь та спілкується за допомогою інтерфейсу із сусідніми процесами. Алгоритм поділяється на незалежні частини так, щоб по можливості рідше виконувати операції обміну даними між процесами і зменшити витрати часу на ці допоміжні операції. Логічні зв'язки між MIMD-процесами у вигляді графа або матриці є віртуальною комунікаційною мережею (комунікаційним графом або матрицею).

Визначення 2. Віртуальною паралельною обчислювальною архітектурою (ВПА) будемо називати необмежену неоднорідну множину повнофункціональних процесорів з локальною пам'яттю, що функціонує за MIMD-принципом і виконує рішення задачі моделювання з програмно регульованим обміном даними за допомогою віртуальної комунікаційної мережі.

Визначення 3. Девіртуалізація віртуальної паралельної моделі є таке перетворення моделі, яке веде до реалізації моделі на даній цільовій паралельній архітектурі і виконує такі основні функції:

- представлення наявних ресурсів цільової обчислювальної архітектури;

- встановлення відповідностей «віртуальні процеси – реальні процесори» та «віртуальні інтерфейси – комунікаційний граф (матриця) – цільова обчислювальна архітектура – комунікаційна мережа» з урахуванням таких критеріїв, як балансування завантаження та мінімальний обмін даними;
- формування специфікації процесорів, які відповідають MIMD-процесам.

Визначення 4. Цільова паралельна обчислювальна архітектура (ЦПА) – наявна у розпорядженні розробника певна множина повнофункціональних процесорів з локальною пам'яттю, що функціонує за MIMD-принципом і виконує рішення задачі моделювання з програмно регульованим обміном даними за допомогою комунікаційної мережі.

З огляду на структуру МДОРП, безперервний характер динамічних процесів та можливі об'єми обчислювальних операцій при реалізації DSM визначимо наступні рівні розпаралелювання вирішувачів рівнянь.

Рівень паралельності РП-1: 1 рівняння – 1 процес.

Віртуальна паралельна модель (ВПМ) для однієї гілки мережного об'єкта, розділеної на M піделементів, представлена на рис. 2.2.

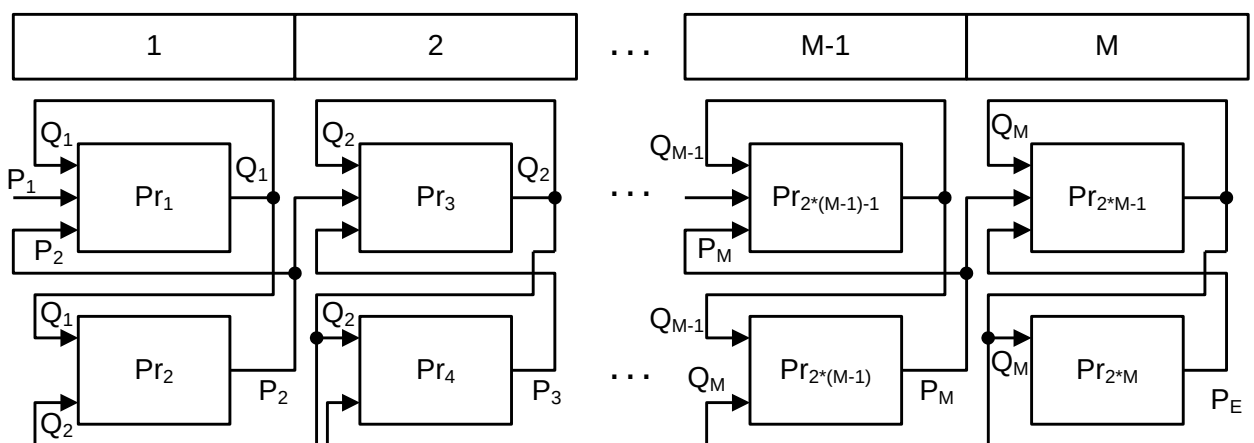


Рисунок 2.2 – Паралельне обчислення змінних $Q(k, t)$, $P(k, t)$ на рівні РП-1

Процеси віртуальної паралельної моделі для вирішувача рівнянь МДОРП відбивають матричну структуру потоків і тисків повітря. Алгоритм передбачає, що кожен процес містить один елемент матриці. Так як рівняння вирішуються в паралельному режимі, то час виконання першого кроку алгоритму обчислення дорівнює часу розв'язання рівняння для dQ_{jk}/dt .

Для мережного об'єкта дана ВПМ вимагає $2(M_1 + M_2 + \dots + M_m)$ процесів для витрат повітря та тиску гілок, де m – кількість гілок у мережному об'єкті, M_j – кількість елементів у гілках.

Рівень паралельності РП-2: 1 елемент гілки – 1 процес.

Система рівнянь Simulation-моделі для k -ого елемента ($1 \leq k \leq M_j$) для j -гілки ($1 \leq j \leq m$) вирішується у відповідному k -му MIMD-процесі. Топологія віртуальної паралельної моделі відображає граф мережного об'єкта і складається з процесів обчислення Q і P у j -гілці, тобто загалом із $m \cdot M_j$ (при $M_1 = M_2 = \dots = M_m$) або $\sum_{j=1}^m M_j$ (при $M_1 \neq M_2 \neq \dots \neq M_m$) процесів. Таким чином, кожен процес вирішує систему з двох рівнянь. Віртуальну паралельну модель однієї гілки показано на рис.2.3.

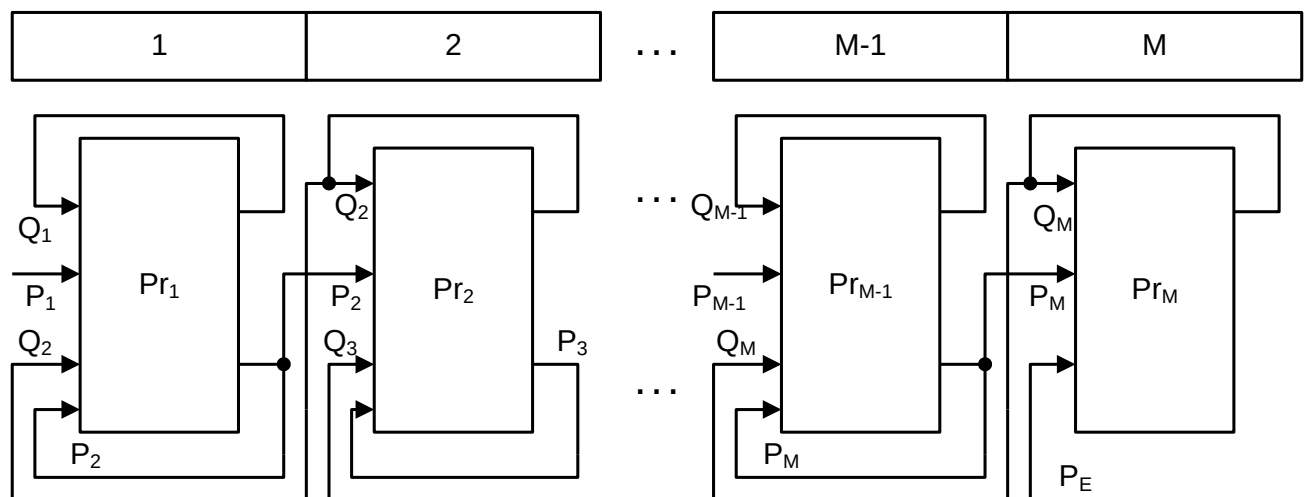


Рисунок 2.3 – Паралельне обчислення змінних $Q(k, t)$, $P(k, t)$ на рівні РП-2.

Рівень паралельності РП-3: 1 гілка – 1 процес.

Рівняння Simulation-моделі для кожної j -гілки ($1 \leq j \leq m$) вирішуються у відповідному їй j -му MIMD-процесі. Топологія віртуальної паралельної моделі відбиває граф мережного об'єкта і складається з m процесів. Кожен процес містить 2 вектора: вектор тисків – P_j і вектор витрат повітря – Q_j . А кожен вектор – M_j елементів.

Рівень паралельності РП-4: 1фрагмент графу МДОРП – 1 процес.

При такому підході до розпаралелювання граф мережного об'єкта розбивається на G_s підграфів, що мають найменшу кількість зв'язків один з одним. Відповідні системи рівнянь вирішуються на G_s MIMD-процесах. У цьому випадку кількість процесів залежить від кількості підграфів, на які розбивається МДОРП.

2.4 Висновки

- 1) Дано визначення підсистеми вирішувачів рівнянь, сформульовано функціональність, розроблено структуру підсистеми.
- 2) Розглянуто формальні підходи до побудови дискретних Simulation-моделей МДОРП, на основі яких мають розроблюватись вирішувачі рівнянь та симулятори.
- 3) Розробки вирішувачів зорієнтовано на можливі рівні розпаралелювання Simulation-моделей МДОРП.

3 Розробка послідовних вирішувачів рівнянь МДОРП

3.1 Вирішувачі j-гілок на основі послідовної мови моделювання

3.1.1 Тестовий МДОРП: топологія, параметри.

На рис. 3.1 показано граф тестового мережевого об'єкту з розподіленими параметрами, а в таблиці 3.1 – параметри гілок та тестові величини потоків.

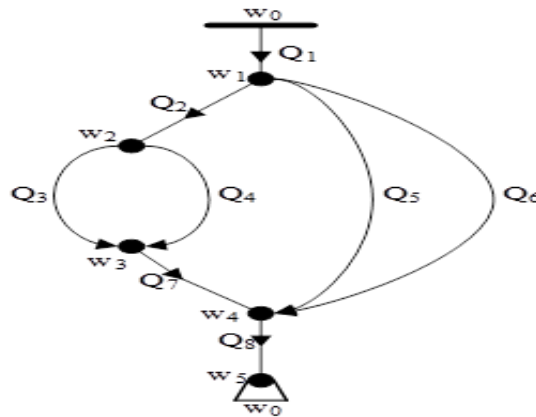


Рисунок 3.1 – Граф тестового МДОРП

Таблиця 3.1 - Параметри гілок тестового МДОРП

Par/Qj	L_j	F_j	r_j	$\Delta x, m$	M_j	α_j	β_j	γ_j	Test P_A	Test P_V	Test $Q_{\text{стац}}$
Q1	800	6.5	0.000484	40	20	0.13	0.00252	523.6	0.0		71.6
Q2	150	6.5	0.000484	40	4	0.13	0.00252	523.6			28.9
Q7	150	6.5	0.000484	40	4	0.13	0.00252	523.6			28.9
Q8	650	6.5	0.000484	40	16	0.13	0.00252	523.6		-4100.	71.6
Q3	1925	4.67	0.00105	40	48	0,093	0.004	728.6			14.4
Q4	1410	5.98	0.0019	40	35	0.12	0.009	569.1			14.9
Q5	1720	4.5	0.00112	40	43	0.09	0.004	756.2			19.2
Q6	934	4.97	0.00137	40	23	0.1	0.0055	684.7			23.7

У топології ми маємо вісім гілок та чотири вузла, у таблиці із тестовими параметрами описано всі необхідні дані (довжина гілки, кількість секцій, діаметр труби, тестові показники тиску на вході та виході системи, тощо) для кожної гілки.

3.1.2 Апроксимація j -гілки за просторовою координатою

Користуючись методом прямих [1], апроксимуємо j -гілку множиною M_j елементів, що мають довжину Δx , та пронумеруємо кінцеві пункти цих елементів як змінну $v = 1, 2, \dots, M_j$. З врахуванням напрямку руху повітряного потоку початок гілки позначимо пунктом 0. Потоки повітря в межах виділених елементів позначимо як $Q_{j1}, Q_{j2}, \dots, Q_{jv}, \dots, Q_{jM_j}$, а величини тисків в пунктах – як $P_{j1}, P_{j2}, \dots, P_{jv}, \dots, P_{jM_j}$. В графі МДОРП (рис. 3.1) початковий пункт 0 співпадає з початковим вузлом мережі AK_{wi} , а кінцевий M_j – з кінцевим вузлом мережі EK_{wi} (рис. 3.2).

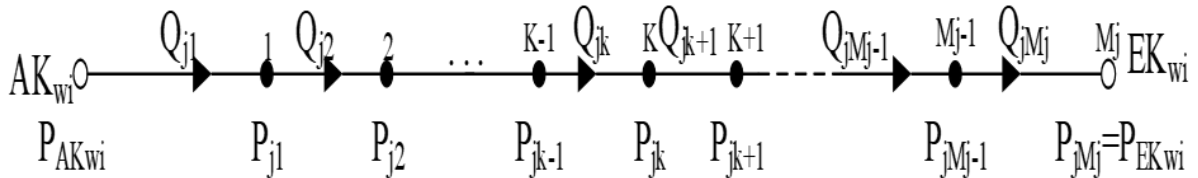


Рисунок 3.2 – Апроксимована j -гілка

3.1.3 Рівняння Simulation-моделі j -гілки

Динаміка руху повітря в j -гілці МДОРП описується системою рівнянь [1]

$$\begin{cases} -\frac{\partial P_j}{\partial x} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t}, \\ -\frac{\partial P_j}{\partial t} = \frac{\rho a^2}{F_j} \frac{\partial Q_j}{\partial x} \end{cases} \quad (3.1)$$

Тут: P_j , Q_j – тиск і потік повітря вздовж координати x , яка змінюється від початкового вузла AK_{wi} до кінцевого вузла EK_{wi} в межах $0 \leq x \leq l_j$ довжини гілки l_j ; r_j – питомий аеродинамічний опір; F_j – площа поперечного перерізу гілки (труба, підземна виробка та ін.); ρ – щільність повітря; a – швидкість звуку в повітрі. Граничними умовами для (3.1) є тиски $P_{AK_{wi}}$, $P_{EK_{wi}}$ в початковому й кінцевому вузлах j -гілки. Система (3.1) є моделлю гілки МДОРП.

З врахуванням напрямку потоку (рис.3.2) рівняння системи (3.1) дискретизуються в пунктах v наступним чином:

$$\left. \begin{aligned} \frac{dQ_{jv}}{dt} &= \alpha_j (P_{jv-1} - P_{jv}) - \beta_j Q_{jv} |Q_{jv}| \\ \frac{dP_{jv}}{dt} &= \gamma_j (Q_{jv} - Q_{jv+1}) \end{aligned} \right\} \quad (3.2)$$

Коефіцієнти обчислюються за формулами:

$$\alpha_j = \frac{F_j}{\rho \Delta x}, \quad \beta_j = \frac{F_j r_j}{\rho}, \quad \gamma_j = \frac{\rho a^2}{F_j \Delta x}$$

Систему (3.2) будемо називати Simulation-моделлю j -гілки МДОРП. Запишемо систему рівнянь Simulation-моделі (3.2) для $1 \leq v \leq M_j$, звернувши увагу на рівняння для $v=M_j-1$ та кінцевого пункту $v=M_j$, в якому

$$P_{M_j} = -R^* Q^2 = \text{const}$$

Для $v=1, j=3$

$$\left. \begin{aligned} \frac{dQ_{3,1}}{dt} &= \alpha_3 (P_{3,0} - P_{3,1}) - \beta_3 Q_{3,1} |Q_{3,1}| \\ \frac{dP_{3,1}}{dt} &= \gamma_3 (Q_{3,1} - Q_{3,2}) \end{aligned} \right\}$$

.....

(3.3)

Для $v = M_3 - 1$

$$\left. \begin{aligned} \frac{dQ_{3,47}}{dt} &= \alpha_3(P_{3,46} - P_{3,47}) - \beta_3 Q_{3,47} |Q_{3,47}| \\ \frac{dP_{3,47}}{dt} &= \gamma_3(Q_{3,47} - Q_{3,48}) \end{aligned} \right\}$$

Для $v = M_3$ підставити: $P_{3M3} = -R_3 \cdot Q^2 = \text{const}$

$$\left. \begin{aligned} \frac{dQ_{3,48}}{dt} &= \alpha_3(P_{3,47} - P_{3,48}) - \beta_3 Q_{3,48} |Q_{3,48}| \\ P_{3,48} &= \text{const} = -R_3 \cdot 100 = -202 \end{aligned} \right\}$$

3.1.4 Блок-схема БО-вирішувача рівнянь j-гілки, процеси $Q_j(v, t)$

На рис. 3.3 наведено блок-схему вирішувача системи рівнянь (3.3) j-гілки, побудовану засобами послідовної мови моделювання Simulink.

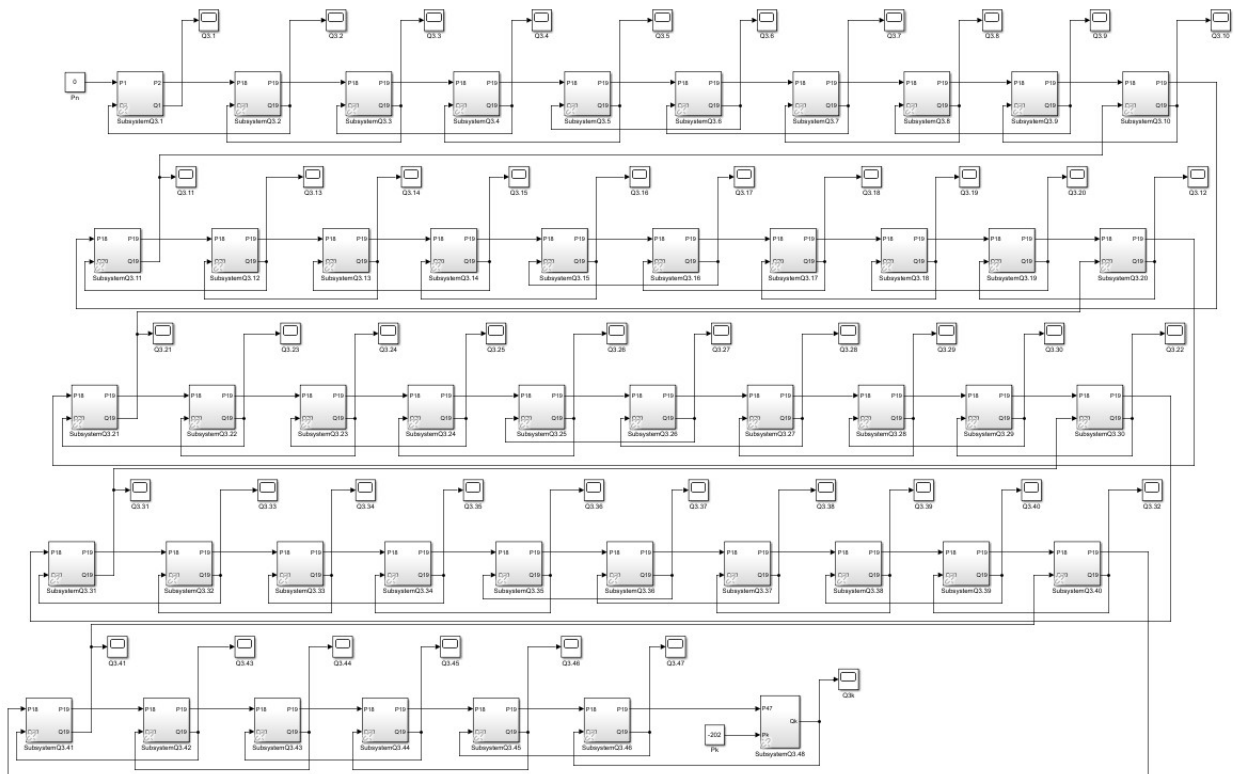


Рисунок 3.3 – Simulink-блок-схема БО-вирішувача рівнянь j-гілки

Підготуємо тестовий тиск для $Q=10 \text{ м}^3/\text{сек}$ як $P_{ЕК} = -R \cdot Q^2$. Так, для гілки Q_3 маємо $R_3=r_3 \cdot l_3 = 0.00105 \cdot 1925 = 2.02$, $P_{ЕК} = -2.02 \cdot 100 = -202 \text{ Ньютон/м}^2$.
Граничні умови: $P_{АК} = 0.0$, $P_{ЕК} = -R \cdot Q^2 = \text{const}$; початкові умови: $Q(x,0)=0.0$ для $0 \leq x \leq l$, $P(x,0)=0.0$ для $0 \leq x \leq l-\Delta x$, а в кінці даємо «стрибок» $P(l,0) = -R \cdot Q^2$.

Для подальшої розробки БО-симулятора всього МДОРП виконано моделювання кожної з гілок, тобто отримано $m=8$ працездатних БО-вирішувачів, які готові до об'єднання в симулятор/вирішувач за топологією МДОРП. Блок-схеми БО-вирішувачів і графіки перехідних процесів змін потоків повітря гілок $Q_1, \dots, Q_8$ наведено в додатку В1. БО-вирішувачі дають стабільні процеси. Після коливань, викликаних розповсюдженням хвилі тиску, потоки виходять на тестову величину $10 \text{ м}^3/\text{сек}$. Чітко фіксуються інтервали запізнення початку процесів в різних пунктах гілки: ці запізнення зумовлені різними моментами часу досягнення пунктів хвилею тиску $P(v, t)$.

3.2 Вирішувачі рівнянь МДОРП на основі послідовної мови моделювання

3.2.1. Специфікації апроксимованих гілок та системи рівнянь Simulation-моделі МДОРП

Розглянемо апроксимовані за просторовою координатою гілки МДОРП як об'єкти спільного моделювання та їх взаємозв'язки через вузли мережі. Вузлові рівняння мають вигляд:

$$\begin{aligned}\frac{dP_{w1}}{dt} &= \gamma_{w1}(Q_{1M1} - Q_{21} - Q_{51} - Q_{61}) \\ \frac{dP_{w2}}{dt} &= \gamma_{w2}(Q_{2M2} - Q_{31} - Q_{41}) \\ \frac{dP_{w3}}{dt} &= \gamma_{w3}(Q_{3M3} + Q_{4M4} - Q_{71}) \\ \frac{dP_{w4}}{dt} &= \gamma_{w4}(Q_{7M7} - Q_{81} - Q_{5M5} - Q_{6M6})\end{aligned}\tag{3.4}$$

Прийmemo наступні значення вузлових коефіцієнтів:

$$\gamma_{w1} = \gamma_1; \gamma_{w2} = \gamma_2; \gamma_{w3} = \gamma_3; \gamma_{w4} = \gamma_4;$$

Гілка Q1. Зовнішня гранична умова – це тиск атмосфери, тому відхилення від абсолютного тиску тут буде $P_{10} = 0$. Потоки повітря в елементах гілки нумеруються як $Q_{11}, Q_{12}, \dots, Q_{1v-1}, Q_{1v}, Q_{1v+1}, \dots, Q_{1M1-1}, Q_{1,M1}$, при $\Delta x = 40M$ маємо $M_1 = 20$ (табл.3.1). Змінні тисків позначаються як $P_{11}, P_{12}, \dots, P_{1v-1}, P_{1v}, \dots, P_{1M1-2}, P_{1M1-1}, P_{1M1}$, при цьому тиском в кінці Q_1 -гілки є тиск у вузлі w1, тобто $P_{1M1} = P_{w1}$ – це внутрішня гранична умова для гілки Q_1 . Невідомі змінні $Q_{11}, Q_{12}, \dots, Q_{1v-1}, Q_{1v}, Q_{1v+1}, \dots, Q_{1M1-1}, Q_{1,M1}$ та $P_{11}, P_{12}, \dots, P_{1v-1}, P_{1v}, \dots, P_{1M1-2}, P_{1M1-1}, P_{1M1}$ можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі гілки Q1, в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_1$ (Тут маємо v замість k). Система рівнянь для останнього елемента гілки Q1 має містити вузлове рівняння, що визначає тиск P_{w1} :

$$\left. \begin{aligned} \frac{dQ_{1,20}}{dt} &= \alpha_1(P_{1,19} - P_{w1}) - \beta_1 Q_{1,20} |Q_{1,20}| \\ \frac{dP_{w1}}{dt} &= \gamma_{w1}(Q_{120} - Q_{21} - Q_{51} - Q_{61}), \end{aligned} \right\} \quad (3.5)$$

Отримані змінні P_{w1} використовуються в якості граничних умов для гілок Q2, Q5, Q6 (рис. 3.1).

Гілка Q2. Внутрішніми граничними умовами є вузлові тиски P_{w1}, P_{w2} . Повітряні потоки в елементах гілки пронумеруємо як $Q_{21}, Q_{22}, \dots, Q_{2v-1}, Q_{2v}, Q_{2v+1}, \dots, Q_{2M2-1}, Q_{2,M2}$, кількість елементів при $\Delta x = 40M$ становить $M_2 = 4$ (таблиця). Величини тисків запишемо у вигляді змінних $P_{w1} = P_{20}, P_{21}, P_{22}, \dots, P_{2v-2}, P_{2v-1}, P_{2v}, \dots, P_{2M2-2}, P_{2M2-1}, P_{2M2} = P_{w2}$, при цьому тиск в кінці Q2-гілки співпадає з тиском у вузлі w2, тобто $P_{2M2} = P_{w2}$ є внутрішньою граничною умовою для Q2-гілки. Невідомі змінні $Q_{21}, Q_{22}, \dots, Q_{2v-1}, Q_{2v}, Q_{2v+1}, \dots, Q_{2M2-1}, Q_{2,M2}$ та $P_{21}, P_{22}, \dots, P_{2v-2}, P_{2v-1}, P_{2v}, \dots, P_{2M2-2}, P_{2M2-1}$

1, P2M2 можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі, в яких індекс v (k) приймає значення $1 \leq v \leq M2$. ГУ-тиски P_{w1} вже визначено (див. вище гілку Q1), а P_{w2} -величини мають обчислюватись як розв'язки рівнянь Simulation-моделі гілки Q2. Система рівнянь для першого елемента гілки Q2 має містити ГУ-тиск $P_{w1}=P_{20}$, отриманий з (3.5), а саме:

$$\left. \begin{aligned} \frac{dQ_{2,1}}{dt} &= \alpha_2(P_{w1} - P_{2,1}) - \beta_2 Q_{2,1} |Q_{2,1}| \\ \frac{dP_{2,1}}{dt} &= \gamma_2(Q_{2,1} - Q_{2,2}) \end{aligned} \right\} \quad (3.6)$$

Система рівнянь для останнього елемента гілки Q2 має містити вузлове рівняння, що визначає тиск P_{w2} :

$$\left. \begin{aligned} \frac{dQ_{2,4}}{dt} &= \alpha_2(P_{2,3} - P_{w2}) - \beta_2 Q_{2,4} |Q_{2,4}| \\ \frac{dP_{w2}}{dt} &= \gamma_{w2}(Q_{2,4} - Q_{3,1} - Q_{4,1}), \end{aligned} \right\} \quad (3.7)$$

Отримані P_{w2} -величини використовуються в якості граничних умов для гілок Q3, Q4, що інцидентні вузлу $w2$ (граф МДОРП, рис.3.1).

Гілка Q3. Внутрішніми граничними умовами є вузлові тиски P_{w2} і P_{w3} . Повітряні потоки в елементах гілки пронумеруємо як $Q_{31}, Q_{32}, \dots, Q_{3v-1}, Q_{3v}, Q_{3v+1}, \dots, Q_{3M3-1}, Q_{3M3}$, кількість елементів при $\Delta x=40M$ становить $M_3 = 48$ (таблиця). Величини тисків запишемо у вигляді змінних $P_{w2}=P_{30}, P_{31}, P_{32}, \dots, P_{3v-2}, P_{3v-1}, P_{3v}, \dots, P_{3M3-2}, P_{3M3-1}, P_{3M3}=P_{w3}$, при цьому тиск в кінці Q₃-гілки співпадає з тиском у вузлі $w3$, тобто $P_{3M3} = P_{w3}$ є внутрішньою граничною умовою для Q₃-гілки. Невідомі змінні $Q_{31}, Q_{32}, \dots, Q_{3v-1}, Q_{3v}, Q_{3v+1}, \dots, Q_{3M3-1}, Q_{3M3}$ та $P_{31}, P_{32}, \dots, P_{3v-2}, P_{3v-1}, P_{3v}, \dots, P_{3M3-2}, P_{3M3-1}, P_{3M3}$ можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі, в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_3$. ГУ-тиски P_{w2} вже визначені (див. вище гілку Q2), а змінна $P_{3M3}=P_{w3}$

має обчислюватись як розв'язок рівнянь Simulation-моделі гілки Q3. При цьому система рівнянь для першого елементу гілки Q3 має містити ГУ-тиск $P_{w2}=P_{30}$ з системи (3.7):

$$\left. \begin{aligned} \frac{dQ_{3,1}}{dt} &= \alpha_3(P_{w2} - P_{3,1}) - \beta_3 Q_{3,1} |Q_{3,1}| \\ \frac{dP_{3,1}}{dt} &= \gamma_3(Q_{3,1} - Q_{3,2}) \end{aligned} \right\} \quad (3.8)$$

Система рівнянь для останнього елементу гілки Q3 має містити вузлове рівняння, що визначає тиск P_{w3} :

$$\begin{aligned} \frac{dQ_{3,48}}{dt} &= \alpha_3(P_{3,47} - P_{w3}) - \beta_3 Q_{3,48} |Q_{3,48}| \\ \frac{dP_{w3}}{dt} &= \gamma_{w3}(Q_{3,48} + Q_{4,35} - Q_{7,1}) \end{aligned} \quad (3.9)$$

Отримані P_{w3} - величини використовуються також в якості граничних умов для гілок Q3, Q4, Q7, що інцидентні вузлу $w3$ (граф МДОРП, рис.3.1).

Гілка Q4. Ця гілка паралельна гілці Q3, тому має внутрішні граничні умови P_{w2} і P_{w3} . Повітряні потоки в елементах гілки пронумеруємо як $Q_{41}, \dots, Q_{4v-1}, Q_{4v}, Q_{4v+1}, \dots, Q_{4M4-1}, Q_{4M4}$, кількість елементів при $\Delta x=40M$ становить $M_4 = 35$ (табл.3.1). Величини тисків запишемо у вигляді змінних $P_{w2}=P_{40}, P_{41}, P_{42}, \dots, P_{4v-2}, P_{4v-1}, P_{4v}, \dots, P_{4M4-2}, P_{4M4-1}, P_{4M4}=P_{w3}$, при цьому тиск в кінці Q_4 -гілки співпадає з тиском у вузлі $w3$, тобто $P_{4M4} = P_{w3}$ є внутрішньою граничною умовою для Q_4 -гілки. Невідомі змінні $Q_{41}, Q_{42}, \dots, Q_{4v-1}, Q_{4v}, Q_{4v+1}, \dots, Q_{4M4-1}, Q_{4M4}$ та $P_{41}, P_{42}, \dots, P_{4v-2}, P_{4v-1}, P_{4v}, \dots, P_{4M4-2}, P_{4M4-1}$ можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі, в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_4$. ГУ-тиски P_{w2} та $P_{w3}=P_{4M4}$ вже визначено (див. вище гілки Q2 і Q3). В рівнянні для $Q_{4,1}$ треба використовувати тиск $P_{w2}=P_{40}$ з системи (3.7):

$$\left. \begin{aligned} \frac{dQ_{4,1}}{dt} &= \alpha_4(P_{w2} - P_{4,1}) - \beta_4 Q_{4,1} |Q_{4,1}| \\ \frac{dP_{4,1}}{dt} &= \gamma_4(Q_{4,1} - Q_{4,2}) \end{aligned} \right\} \quad (3.10)$$

В рівнянні для Q_{4M4} треба використовувати тиск $P_{4M4}=P_{w3}$, тобто:

$$\left. \begin{aligned} \frac{dQ_{4,35}}{dt} &= \alpha_4(P_{4,34} - P_{w3}) - \beta_4 Q_{4,35} |Q_{4,35}| \\ P_{w3} &- \text{з системи (3.9)} \end{aligned} \right\} \quad (3.11)$$

Гілка Q5. Внутрішніми граничними умовами є вузлові тиски P_{w1} , P_{w4} . Повітряні потоки в елементах гілки пронумеруємо як Q_{51} , $Q_{52}, \dots, Q_{5v-1}$, Q_{5v} , Q_{5v+1} , $\dots, Q_{5M5-1}$, Q_{5M5} , кількість елементів при $\Delta x=40M$ становить $M_5 = 43$ (табл. 3.1). Величини тисків запишемо у вигляді змінних $P_{w1}=P_{50}$, P_{51} , $P_{52}, \dots, P_{5v-2}$, P_{5v-1} , $P_{5v}, \dots, P_{5M5-2}$, P_{5M5-1} , $P_{5M5}=P_{w4}$, при цьому тиск в кінці Q_5 -гілки співпадає з тиском у вузлі $w4$, тобто $P_{5M5} = P_{w4}$ є внутрішньою граничною умовою для Q_5 -гілки. Невідомі змінні Q_{51} , $Q_{52}, \dots, Q_{5v-1}$, Q_{5v} , Q_{5v+1} , $\dots, Q_{5M5-1}$, Q_{5M5} та P_{51} , $P_{52}, \dots, P_{5v-2}$, P_{5v-1} , $P_{5v}, \dots, P_{5M5-2}$, P_{5M5-1} можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі гілки Q5, в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_5$. ГУ-тиск P_{w1} вже визначено (див. вище гілку Q1), а $P_{5M5}=P_{w4}$ -величини мають обчислюватись як розв'язки рівнянь Simulation-моделі гілки Q5. В рівнянні для $Q_{5,1}$ треба використовувати тиск $P_{w1}=P_{50}$ із системи (3.5):

$$\left. \begin{aligned} \frac{dQ_{5,1}}{dt} &= \alpha_5(P_{w1} - P_{5,1}) - \beta_5 Q_{5,1} |Q_{5,1}| \\ \frac{dP_{5,1}}{dt} &= \gamma_5(Q_{5,1} - Q_{5,2}) \end{aligned} \right\} \quad (3.12)$$

Система рівнянь для останнього елемента гілки Q5 ($Q_{5,43}$) має містити вузлове рівняння, що визначає тиск P_{w4} :

$$\begin{aligned}\frac{dQ_{5,43}}{dt} &= \alpha_5(P_{5,42} - P_{w4}) - \beta_5 Q_{5,43} |Q_{5,43}| \\ \frac{dP_{w4}}{dt} &= \gamma_{w4}(Q_{7,4} + Q_{5,43} + Q_{6,23} - Q_{8,1})\end{aligned}\quad (3.13)$$

Отримані P_{w4} - величини використовуються також в якості граничних умов для гілок Q5, Q6, Q7 та Q8, що інцидентні вузлу w4 (граф МДОРП, рис.3.1).

Гілка Q6. Ця гілка паралельна гілці Q5, тому має внутрішні граничні умови P_{w1} і P_{w4} . Повітряні потоки в елементах гілки пронумеруємо як $Q_{61}, \dots, Q_{6v-1}, Q_{6v}, Q_{6v+1}, \dots, Q_{6M6-1}, Q_{6M6}$, кількість елементів при $\Delta x=40M$ становить $M_6 = 23$ (табл. 3.1). Величини тисків запишемо у вигляді змінних $P_{w1}=P_{60}, P_{61}, P_{62}, \dots, P_{6v-2}, P_{6v-1}, P_{6v}, \dots, P_{6M6-2}, P_{6M6-1}, P_{6M6}=P_{w4}$, при цьому тиск в кінці Q_6 -гілки співпадає з тиском у вузлі w4, тобто $P_{6M6} = P_{w4}$ є внутрішньою граничною умовою для Q_6 -гілки. Невідомі змінні $Q_{61}, \dots, Q_{6v-1}, Q_{6v}, Q_{6v+1}, \dots, Q_{6M6-1}, Q_{6M6}$ та $P_{61}, P_{62}, \dots, P_{6v-2}, P_{6v-1}, P_{6v}, \dots, P_{6M6-2}, P_{6M6-1}$ можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі гілки Q6, в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_6$. ГУ-тиски P_{w1} та $P_{w4} = P_{6M6}$ вже визначено (див. вище гілки Q1 і Q5).

В рівнянні для $Q_{6,1}$ треба використовувати тиск $P_{w1}=P_{60}$ із системи (3.5):

$$\left. \begin{aligned}\frac{dQ_{6,1}}{dt} &= \alpha_6(P_{w1} - P_{6,1}) - \beta_6 Q_{6,1} |Q_{6,1}| \\ \frac{dP_{6,1}}{dt} &= \gamma_6(Q_{6,1} - Q_{6,2})\end{aligned}\right\} \quad (3.14)$$

В рівнянні для Q_{6M6} треба використовувати тиск $P_{6M6} = P_{w4}$, тобто:

$$\left. \begin{aligned}\frac{dQ_{6,23}}{dt} &= \alpha_6(P_{6,22} - P_{w4}) - \beta_6 Q_{6,23} |Q_{6,23}| \\ P_{w4} &- \text{з системи (3.13)}\end{aligned}\right\} \quad (3.15)$$

Гілка Q7. Внутрішніми граничними умовами є вузлові тиски P_{w3} , P_{w4} . Повітряні потоки в елементах гілки пронумеруємо як $Q_{71}, \dots, Q_{7v-1}, Q_{7v}, Q_{7v+1}, \dots, Q_{7M7-1}, Q_{7M7}$, кількість елементів при $\Delta x=40M$ становить $M_7 = 4$ (табл. 3.1). Величини тисків запишемо у вигляді змінних $P_{w3}=P_{70}, P_{71}, P_{72}, \dots, P_{7v-2}, P_{7v-1}, P_{7v}, \dots, P_{7M7-2}, P_{7M7-1}, P_{7M7}=P_{w4}$, при цьому тиск в кінці Q7-гілки співпадає з тиском у вузлі w4, тобто $P_{7M7} = P_{w4}$ є внутрішньою граничною умовою для Q7-гілки. Невідомі змінні $Q_{71}, \dots, Q_{7v-1}, Q_{7v}, Q_{7v+1}, \dots, Q_{7M7-1}, Q_{7M7}$ та $P_{71}, P_{72}, \dots, P_{7v-2}, P_{7v-1}, P_{7v}, \dots, P_{7M7-1}$ можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі гілки Q7, в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_7$. ГУ-тиски P_{w3} та $P_{w4} = P_{7M7}$ вже визначено (див. вище гілки Q3 і Q5). В рівнянні для $Q_{7,1}$ треба використовувати тиск $P_{w3}=P_{70}$ із системи (3.9):

$$\left. \begin{aligned} \frac{dQ_{7,1}}{dt} &= \alpha_7(P_{w3} - P_{7,1}) - \beta_7 Q_{7,1} |Q_{7,1}| \\ \frac{dP_{7,1}}{dt} &= \gamma_7(Q_{7,1} - Q_{7,2}) \end{aligned} \right\} \quad (3.16)$$

В рівнянні для Q_{7M7} треба використовувати тиск $P_{7M7} = P_{w4}$.

$$\left. \begin{aligned} \frac{dQ_{7,4}}{dt} &= \alpha_7(P_{7,3} - P_{w4}) - \beta_7 Q_{7,4} |Q_{7,4}| \\ P_{w4} &- \text{з системи (3.13)} \end{aligned} \right\} \quad (3.17)$$

Гілка Q8. Внутрішньою граничною умовою є вузловий тиск P_{w4} , зовнішньою – тиск вентилятора P_v (граф МДОРП, рис.3.1). Повітряні потоки в елементах гілки пронумеруємо як $Q_{81}, Q_{82}, \dots, Q_{8v-1}, Q_{8v}, Q_{8v+1}, \dots, Q_{8M8-1}, Q_{8M8}$, кількість елементів при $\Delta x=40M$ становить $M_8 = 16$ (табл. 3.1). Величини тисків запишемо у вигляді змінних $P_{80} = P_{w4}, P_{81}, P_{82}, \dots, P_{8v-2}, P_{8v-1}, P_{8v}, \dots, P_{8M8-2}, P_{8M8-1}, P_{8M8}$, при цьому тиск в кінці Q8-гілки співпадає з тиском вентилятора,

тобто $P_{8M8} = P_V$ є зовнішньою граничною умовою гілки Q_8 . Невідомі змінні $Q_{81}, Q_{82}, \dots, Q_{8v-1}, Q_{8v}, Q_{8v+1}, \dots, Q_{8M8-1}, Q_{8,M8}$ та $P_{81}, P_{82}, \dots, P_{8v-2}, P_{8v-1}, P_{8v}, \dots, P_{8M8-2}, P_{8M8-1}$ можуть бути обчислені як результати розв'язання рівнянь Simulation-моделі гілки Q_8 , в яких індекс $v(k)$ приймає значення $1 \leq v \leq M_8$. ГУ-тиски $P_{8M8} = P_{w4}$ вже визначено (див. вище гілку Q_5). ГУ- P_V задається як характеристика вентилятора $P_V = f(Q_8)$. Для модельних експериментів приймемо $P_V = \text{const}$ (табл. 3.1). В рівнянні для $Q_{8,1}$ треба використовувати тиск $P_{w4} = P_{80}$ із системи (3.13):

$$\left. \begin{aligned} \frac{dQ_{8,1}}{dt} &= \alpha_8(P_{w4} - P_{8,1}) - \beta_8 Q_{8,1} |Q_{8,1}| \\ \frac{dP_{8,1}}{dt} &= \gamma_8(Q_{8,1} - Q_{8,2}) \end{aligned} \right\} \quad (3.18)$$

В рівнянні для Q_{8M8} треба використовувати тиск $P_{8M8} = P_V$.

$$\left. \begin{aligned} \frac{dQ_{8,16}}{dt} &= \alpha_8(P_{8,15} - P_V) - \beta_8 Q_{8,16} |Q_{8,16}| \\ P_V &= -4100, \text{ const} \end{aligned} \right\} \quad (3.19)$$

3.2.2 Simulation-модель МДОРП і розробка блок-схеми БО-вирішувача рівнянь МДОРП

Simulation-модель МДОРП складається з m систем рівнянь (3.2) з відповідними граничними умовами. Для тестового МДОРП (рис 3.1) маємо рівняння вузлових тисків (3.4), рівняння гілок типу (3.3) та рівняння (3.5)...(3.19), в яких враховано граничні умови.

Блок-схему БО-вирішувача рівнянь Simulation-моделі МДОРП показано на рис.3.4.

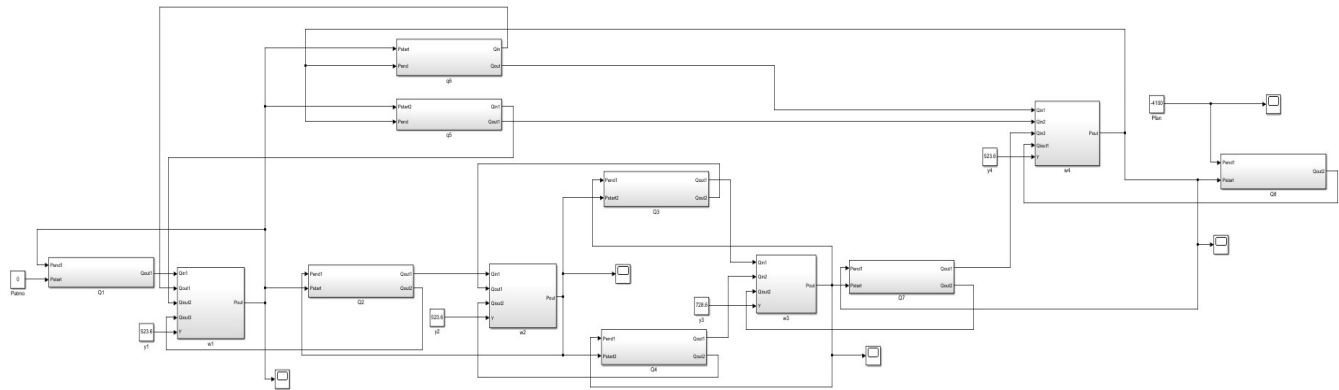


Рисунок 3.4 – БО-вирішувач рівнянь МДОРП.

Перехідні процеси $Q_j(v, t)$ на рисунках 3.5...3.12 ($Q_1...Q_8$)

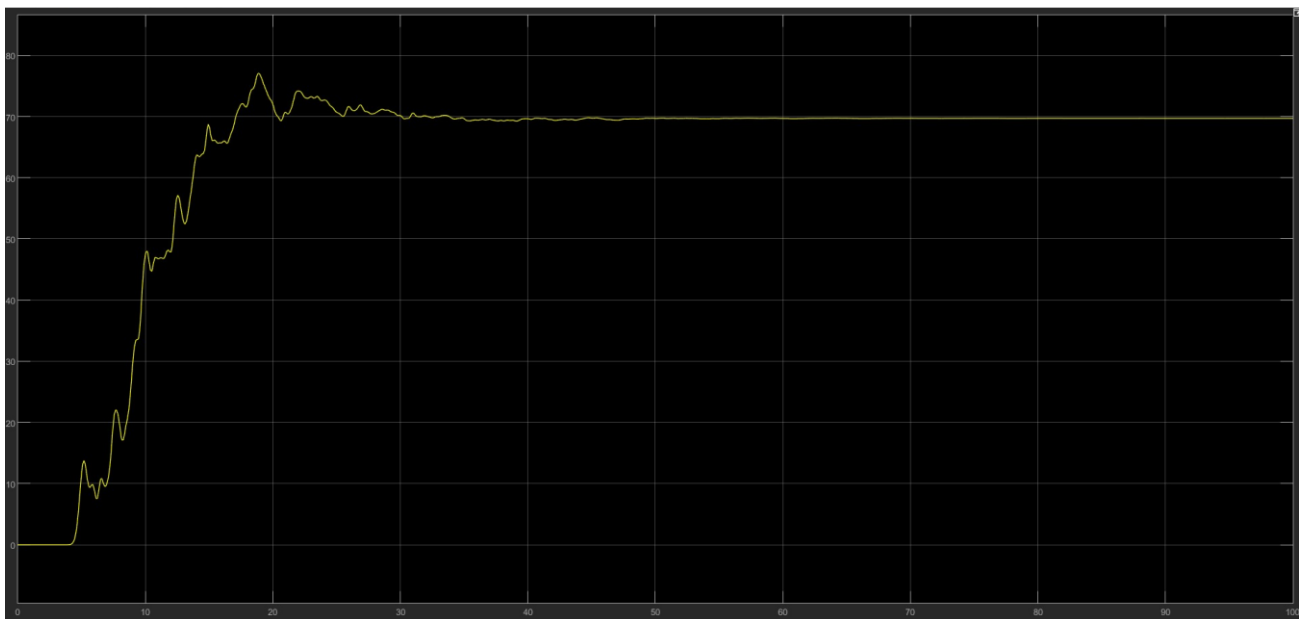


Рисунок 3.5 - Перехідні процеси $Q_1(v, t)$

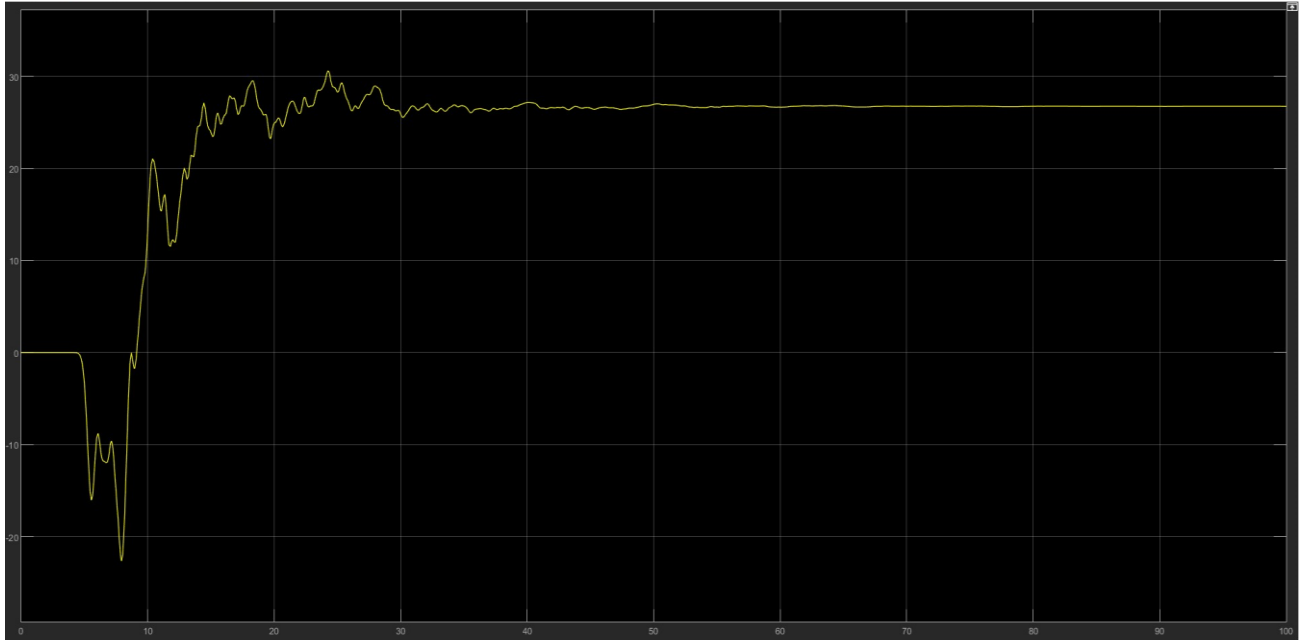


Рисунок 3.6 - Перехідні процеси $Q_2(v, t)$

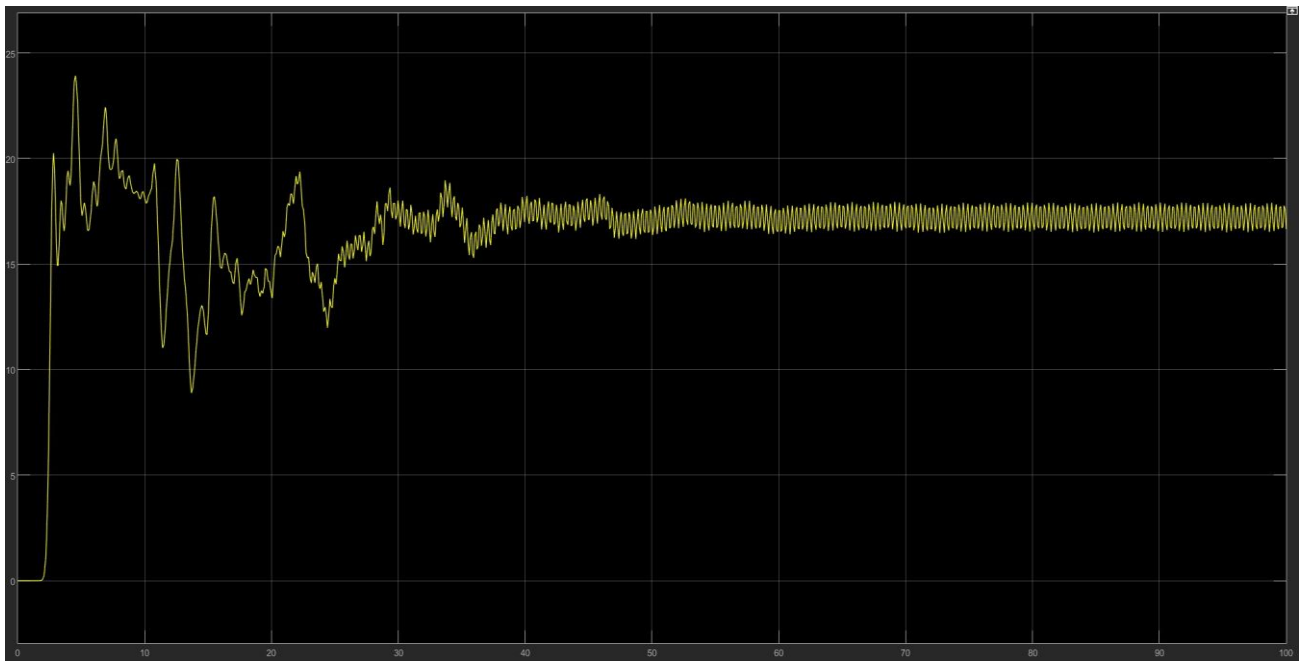


Рисунок 3.7 - Перехідні процеси $Q_3(v, t)$

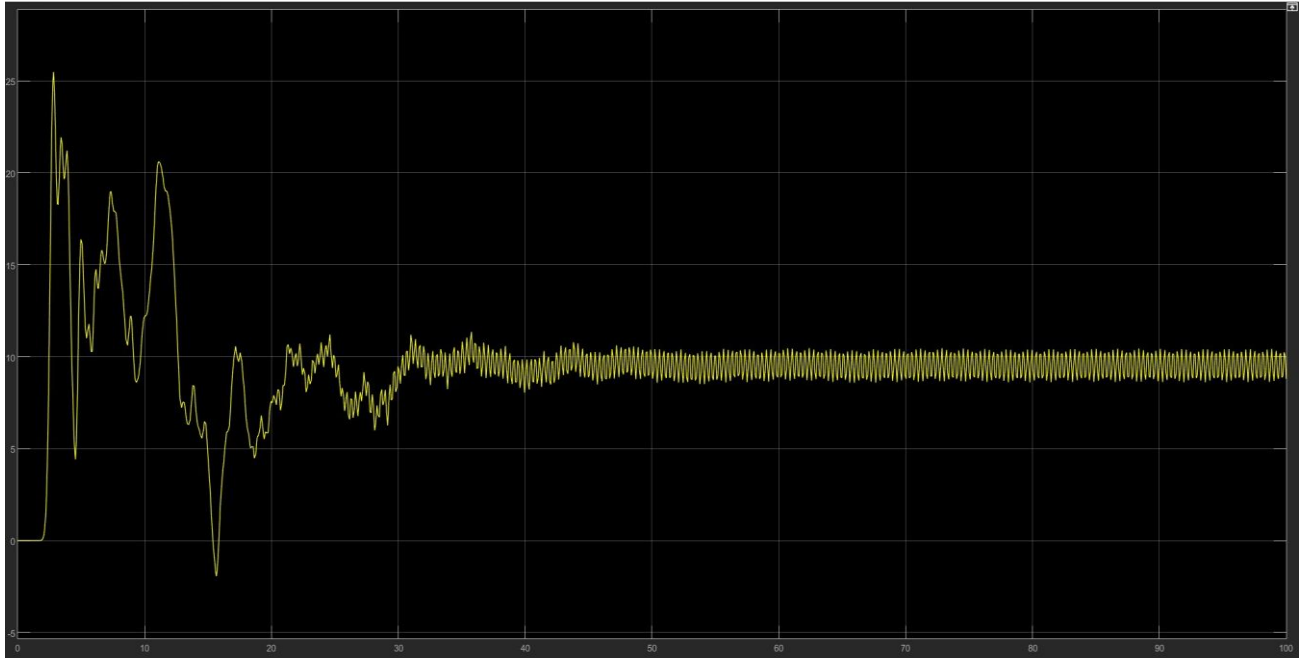


Рисунок 3.8 - Перехідні процеси $Q4(v, t)$

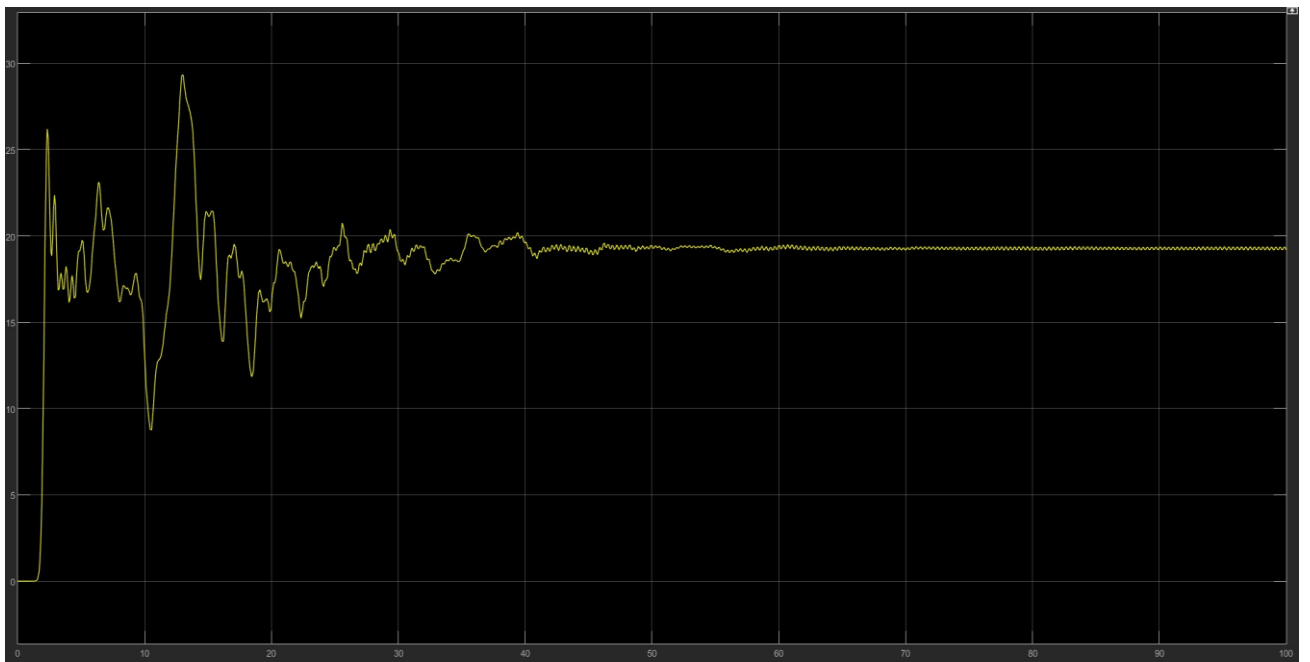


Рисунок 3.9 - Перехідні процеси $Q5(v, t)$

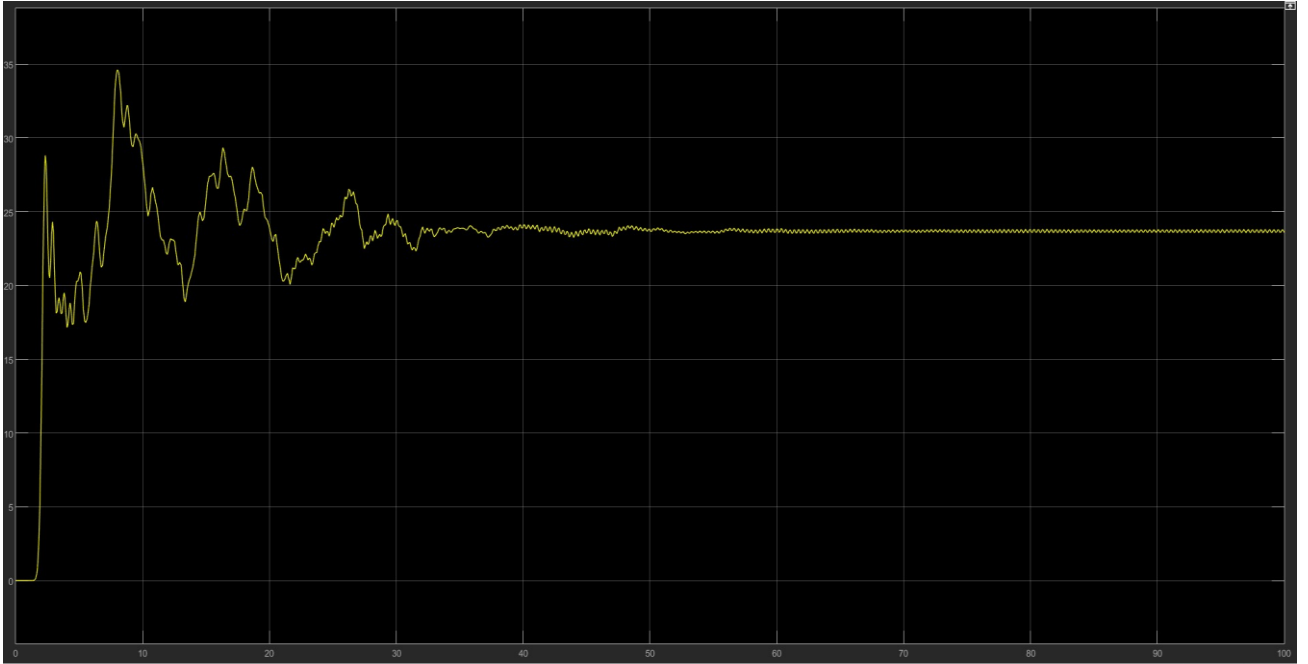


Рисунок 3.10 - Перехідні процеси $Q_6(v, t)$

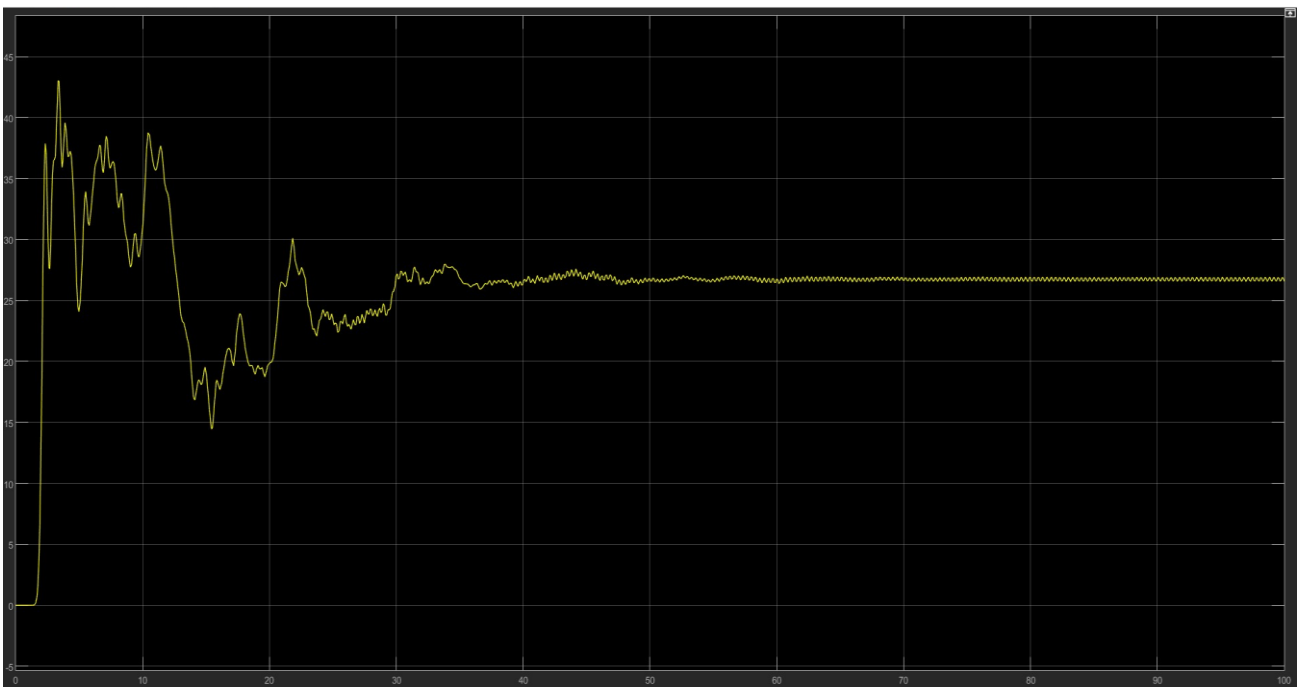


Рисунок 3.11 - Перехідні процеси $Q_7(v, t)$

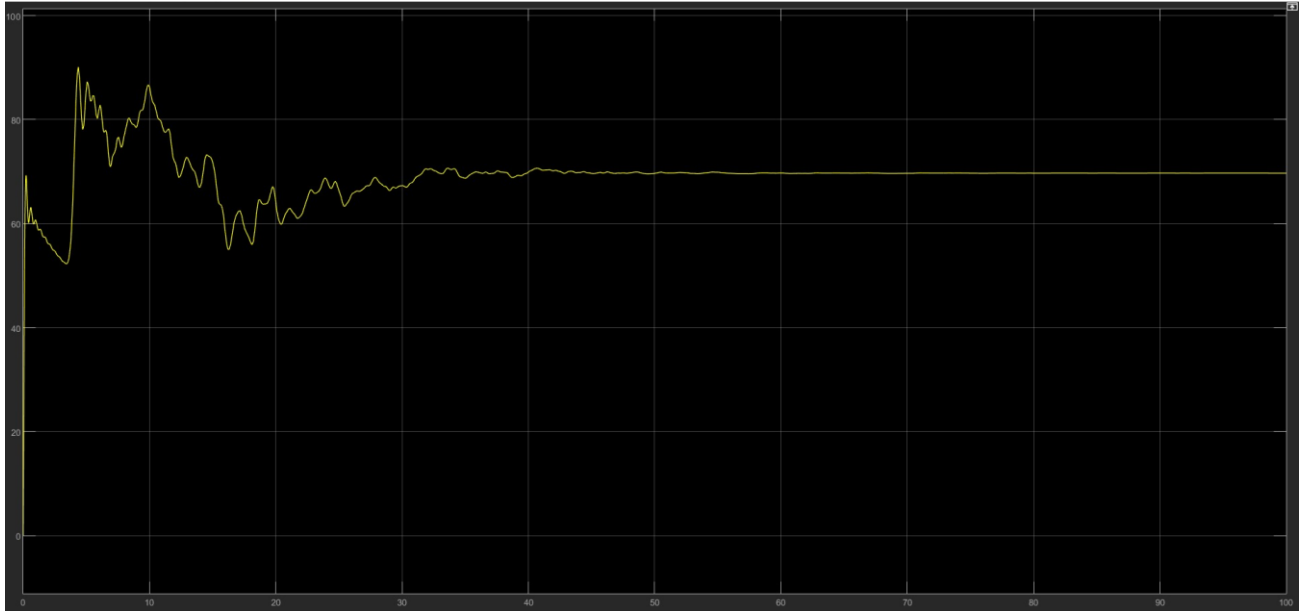


Рисунок 3.12 - Перехідні процеси $Q8(v, t)$

3.3 Вирішувачі j-гілок на основі послідовної С-мови програмування

3.3.1 Дискретна Simulation-модель j-гілки

Оберемо для дискретної Simulation-моделі (ДСМ) j-гілки метод Ейлера і побудуємо відповідний вирішувач рівнянь на прикладі Q3-рівнянь (3.3):

$$\begin{cases} Q_{3,1}(i+1) = Q_{3,1}(i) + h[\alpha_3 (P_{3,0}(i) - P_{3,1}(i)) - \beta_3 Q_{3,1}(i)|Q_{3,1}(i)|] \\ P_{3,1}(i+1) = P_{3,1}(i) + h\gamma_3(Q_{3,1}(i) - Q_{3,2}(i)) \end{cases}$$

$$\begin{cases} Q_{3,2}(i+1) = Q_{3,2}(i) + h[\alpha_3 (P_{3,1}(i) - P_{3,2}(i)) - \beta_3 Q_{3,2}(i)|Q_{3,2}(i)|] \\ P_{3,2}(i+1) = P_{3,2}(i) + h\gamma_3(Q_{3,2}(i) - Q_{3,3}(i)) \end{cases}$$

.....

$$\begin{cases} Q_{3,47}(i+1) = Q_{3,47}(i) + h[\alpha_3 (P_{3,46}(i) - P_{3,47}(i)) - \beta_3 Q_{3,47}(i)|Q_{3,47}(i)|] \\ P_{3,47}(i+1) = P_{3,47}(i) + h\gamma_3(Q_{3,47}(i) - Q_{3,48}(i)) \end{cases}$$

$$\begin{cases} Q_{3,48}(i+1) = Q_{3,48}(i) + h[\alpha_3 (P_{3,47}(i) - P_{3,48}(i)) - \beta_3 Q_{3,48}(i)|Q_{3,48}(i)|] \\ P_{3,48} = \text{const} = -202 \end{cases}$$

3.3.2 Блок-схема і текст програми С-вирішувача рівнянь j -гілки, процеси $Q_j(v, t)$

Блок-схема програми С-вирішувача рівнянь j -гілки (рис. 3.13) містить цикли відносно часових інтервалів і та пунктів просторової координати v . Текст програми – в додатку Д2 (Додаток Д2 - С-програма вирішувача для j -гілок). Для дослідження працездатності С-вирішувача використаємо дані тесту, наведені в п. 3.1.4.

Перехідні процеси $Q_3(v, t)$ показано на рис. 3.14. Як видно з графіків, результати роботи С-вирішувача відповідають фізичному змісту динамічних процесів: хвиля тиску розповсюджується вздовж гілки, викликає аналогічні коливання потоку в різних пунктах v , потік виходить на стаціонарний режим з тестовою величиною $Q_{\text{test}} = 10 \text{ м}^3/\text{сек}$. Чітко ілюструється особливість об'єкту з розподіленими параметрами: процеси в різних пунктах v починаються з запізненням, зумовленим різною відстанню від пункту розташування джерела тиску.

Відлагодимо цю програму стосовно всіх інших гілок, щоб використати С- j -вирішувачі для розробки С-МДОРП-вирішувача. Динамічні процеси $Q_j(v, t)$ гілок $Q_1, \dots, Q_8$ наведено в додатку В3 (ДОДАТОК В3 - Динамічні процеси j -гілок).

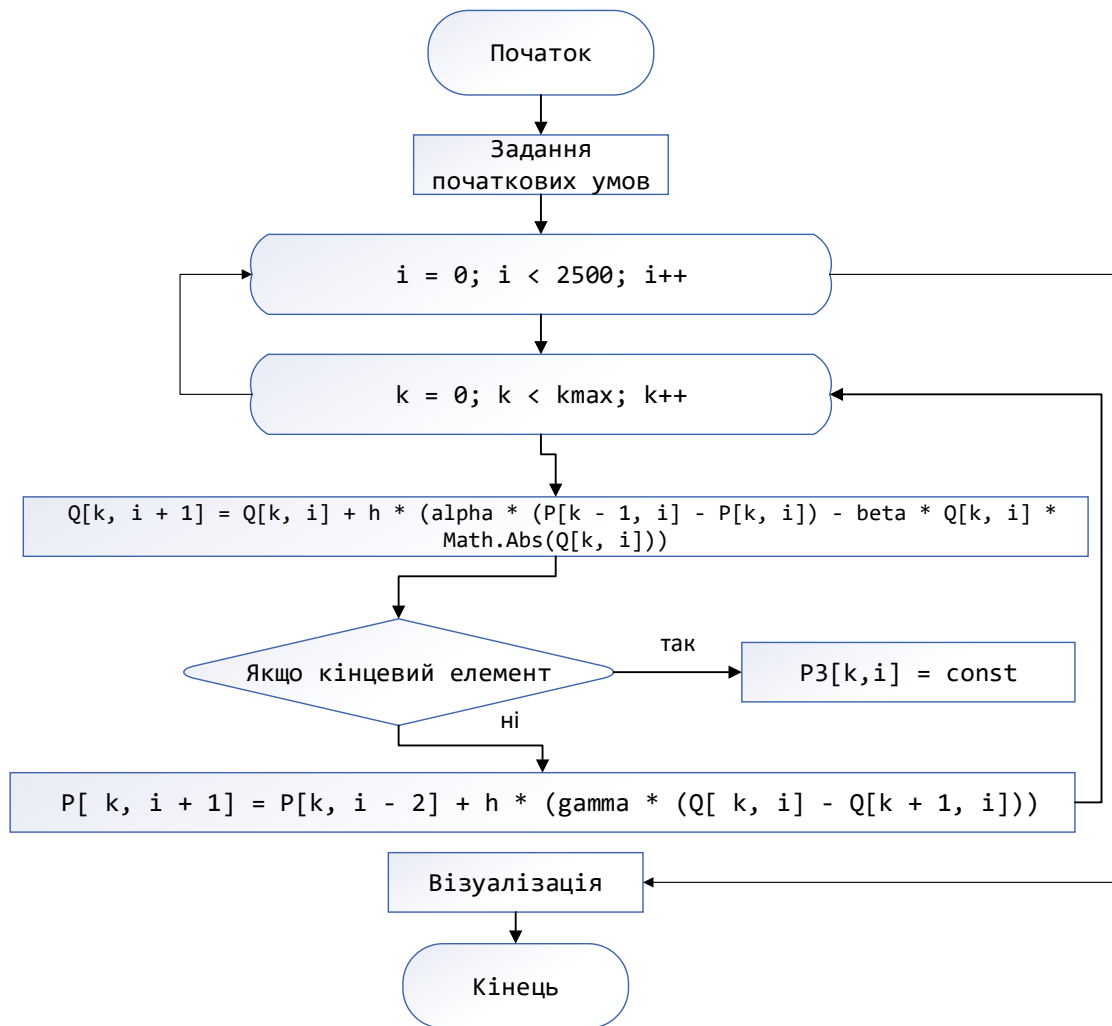
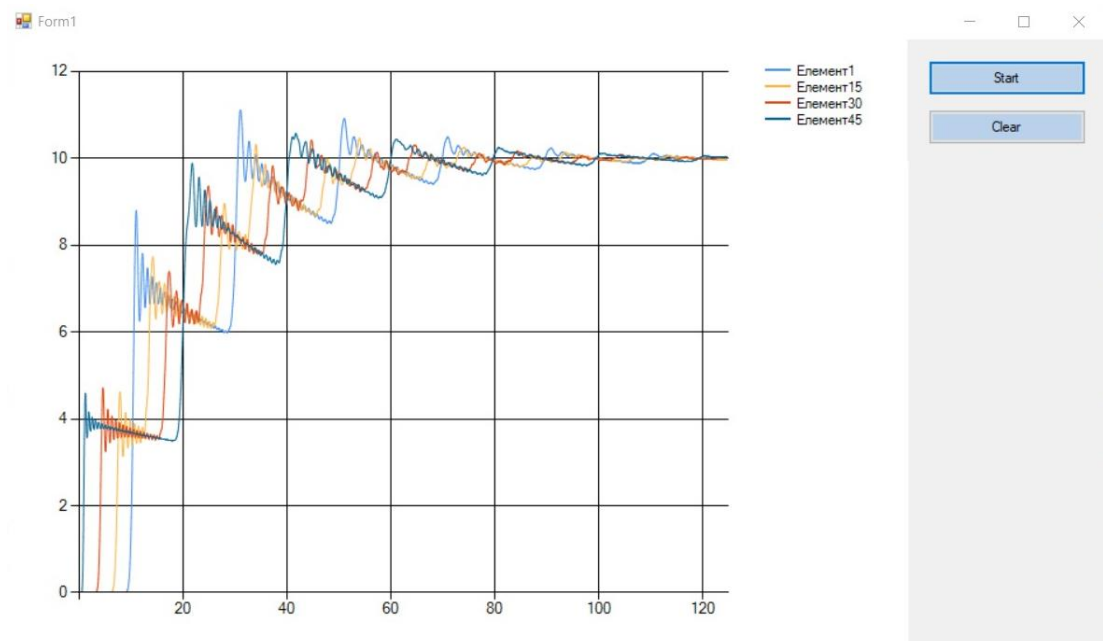


Рисунок 3.13 – Блок-схема програми С-вирішувача рівнянь j-гілки

Рисунок 3.14 – Динамічні процеси $Q_3(v, t)$: 1-й, 15-й, 30-й та 45-й елементи

3.4 Вирішувачі рівнянь МДОРП на основі послідовної С-мови програмування

3.4.1 Дискретна Simulation-модель як основа С-вирішувача МДОРП

Для побудови С-вирішувача запишемо рівняння Simulation-моделі МДОРП стосовно початкових і кінцевих елементів гілок з врахуванням граничних умов, що презентуються відповідними вузловими рівняннями.

Гілка Q1, вузли $P_{1,0}$, P_{w1} :

$$\left. \begin{aligned} \frac{dQ_{1,1}}{dt} &= \alpha_1(P_{1,0} - P_{1,1}) - \beta_1 Q_{1,1} |Q_{1,1}| \\ \frac{dP_{1,1}}{dt} &= \gamma_1(Q_{1,1} - Q_{1,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{1,20}}{dt} &= \alpha_1(P_{1,19} - P_{w1}) - \beta_1 Q_{1,20} |Q_{1,20}| \\ \frac{dP_{w1}}{dt} &= \gamma_{w1}(Q_{1,20} - Q_{2,1} - Q_{5,1} - Q_{6,1}) \end{aligned} \right\}$$

Гілка Q2, вузли P_{w1} , P_{w2} :

$$\left. \begin{aligned} \frac{dQ_{2,1}}{dt} &= \alpha_2(P_{w1} - P_{2,1}) - \beta_2 Q_{2,1} |Q_{2,1}| \\ \frac{dP_{2,1}}{dt} &= \gamma_2(Q_{2,1} - Q_{2,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{2,4}}{dt} &= \alpha_2(P_{2,3} - P_{w2}) - \beta_2 Q_{2,4} |Q_{2,4}| \\ \frac{dP_{w2}}{dt} &= \gamma_{w2}(Q_{2,4} - Q_{3,1} - Q_{4,1}) \end{aligned} \right\}$$

Гілка Q3, вузли P_{w2} , P_{w3} :

$$\left. \begin{aligned} \frac{dQ_{3,1}}{dt} &= \alpha_3(P_{w2} - P_{3,1}) - \beta_3 Q_{3,1} |Q_{3,1}| \\ \frac{dP_{3,1}}{dt} &= \gamma_3(Q_{3,1} - Q_{3,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{3,48}}{dt} &= \alpha_3(P_{3,47} - P_{w3}) - \beta_3 Q_{3,48} |Q_{3,48}| \\ \frac{dP_{w3}}{dt} &= \gamma_{w3}(Q_{3,48} + Q_{4,35} - Q_{7,1}) \end{aligned} \right\}$$

Гілка Q4, вузли P_{w2} , P_{w3} :

$$\left. \begin{aligned} \frac{dQ_{4,1}}{dt} &= \alpha_4(P_{w2} - P_{4,1}) - \beta_4 Q_{4,1} |Q_{4,1}| \\ \frac{dP_{4,1}}{dt} &= \gamma_4(Q_{4,1} - Q_{4,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{4,35}}{dt} &= \alpha_4(P_{4,34} - P_{w3}) - \beta_4 Q_{4,35} |Q_{4,35}| \\ P_{w3} &= \text{з системи } Q_{3,48} \end{aligned} \right\}$$

Гілка Q5, вузли P_{w1} , P_{w4} :

$$\left. \begin{aligned} \frac{dQ_{5,1}}{dt} &= \alpha_5(P_{w1} - P_{5,1}) - \beta_5 Q_{5,1} |Q_{5,1}| \\ \frac{dP_{5,1}}{dt} &= \gamma_5(Q_{5,1} - Q_{5,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{5,43}}{dt} &= \alpha_5(P_{5,42} - P_{w4}) - \beta_5 Q_{5,43} |Q_{5,43}| \\ \frac{dP_{w4}}{dt} &= \gamma_{w4}(Q_{7,4} + Q_{5,43} + Q_{6,23} - Q_{8,1}) \end{aligned} \right\}$$

Гілка Q6, вузли P_{w1} , P_{w4} :

$$\left. \begin{aligned} \frac{dQ_{6,1}}{dt} &= \alpha_6(P_{w1} - P_{6,1}) - \beta_6 Q_{6,1} |Q_{6,1}| \\ \frac{dP_{6,1}}{dt} &= \gamma_6(Q_{6,1} - Q_{6,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{6,23}}{dt} &= \alpha_6(P_{6,22} - P_{w4}) - \beta_6 Q_{6,23} |Q_{6,23}| \\ P_{w4} &= \text{з системи } Q_{5,43} \end{aligned} \right\}$$

Гілка Q7, вузли P_{w3} , P_{w4} :

$$\left. \begin{aligned} \frac{dQ_{7,1}}{dt} &= \alpha_7(P_{w3} - P_{7,1}) - \beta_7 Q_{7,1} |Q_{7,1}| \\ \frac{dP_{7,1}}{dt} &= \gamma_7(Q_{7,1} - Q_{7,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{7,4}}{dt} &= \alpha_7(P_{7,3} - P_{w4}) - \beta_7 Q_{7,4} |Q_{7,4}| \\ P_{w4} &= \text{з системи } Q_{5,43} \end{aligned} \right\}$$

Гілка Q8, вузли P_{w4} , P_v :

$$\left. \begin{aligned} \frac{dQ_{8,1}}{dt} &= \alpha_8(P_{w4} - P_{8,1}) - \beta_8 Q_{8,1} |Q_{8,1}| \\ \frac{dP_{8,1}}{dt} &= \gamma_8(Q_{8,1} - Q_{8,2}) \end{aligned} \right\}$$

.....

$$\left. \begin{aligned} \frac{dQ_{8,16}}{dt} &= \alpha_8(P_{8,15} - P_v) - \beta_8 Q_{8,16} |Q_{8,16}| \\ P_v &= -4100, \text{const} \end{aligned} \right\}$$

Дискретна Simulation-модель МДОРП на основі методу Ейлера:

Гілка Q1, вузли $P_{1,0}$, P_{w1} :

$$\begin{cases} Q_{1,1}(i+1) = Q_{1,1}(i) + h[\alpha_1(P_{1,0}(i) - P_{1,1}(i)) - \beta_1 Q_{1,1}(i)|Q_{1,1}(i)| \\ P_{1,1}(i+1) = P_{1,1}(i) + h\gamma_1(Q_{1,1}(i) - Q_{1,2}(i)) \end{cases}$$

.....

$$\begin{cases} Q_{1,20}(i+1) = Q_{1,20}(i) + h[\alpha_{1,19}(P_{1,19}(i) - P_{w1}(i)) - \beta_1 Q_{1,20}(i)|Q_{1,20}(i)| \\ P_{w1}(i+1) = P_{w1}(i) + h\gamma_{w1}[Q_{1,20}(i) - Q_{2,1}(i) - Q_{5,1}(i) - Q_{6,1}(i)] \end{cases}$$

Гілка Q2, вузли P_{w1} , P_{w2} :

$$\begin{cases} Q_{2,1}(i+1) = Q_{2,1}(i) + h[\alpha_2(P_{w1}(i) - P_{2,1}(i)) - \beta_2 Q_{2,1}(i)|Q_{2,1}(i)| \\ P_{2,1}(i+1) = P_{2,1}(i) + h\gamma_2(Q_{2,1}(i) - Q_{2,2}(i)) \end{cases}$$

.....

$$\begin{cases} Q_{2,4}(i+1) = Q_{2,4}(i) + h[\alpha_2(P_{2,3}(i) - P_{w2}(i)) - \beta_2 Q_{2,4}(i)|Q_{2,4}(i)| \\ P_{w2}(i+1) = P_{w2}(i) + h\gamma_{w2}[Q_{2,4}(i) - Q_{3,1}(i) - Q_{4,1}(i)] \end{cases}$$

Для гілок Q3, Q4 дискретні Simulation-моделі будуються аналогічно з врахуванням граничних умов P_{w2} та P_{w3} .

Гілка Q5, вузли P_{w1} , P_{w4} :

$$\begin{cases} Q_{5,1}(i+1) = Q_{5,1}(i) + h[\alpha_5(P_{w1}(i) - P_{5,1}(i)) - \beta_5 Q_{5,1}(i)|Q_{5,1}(i)| \\ P_{5,1}(i+1) = P_{5,1}(i) + h\gamma_5(Q_{5,1}(i) - Q_{5,2}(i)) \end{cases}$$

.....

$$\begin{cases} Q_{5,43}(i+1) = Q_{5,43}(i) + h[\alpha_5(P_{5,42}(i) - P_{w4}(i)) - \beta_5 Q_{5,43}(i)|Q_{5,43}(i)| \\ P_{w4}(i+1) = P_{w4}(i) + h\gamma_{w4}[Q_{7,4}(i) - Q_{5,43}(i) + Q_{6,23}(i) - Q_{8,1}(i)] \end{cases}$$

Для гілок Q6, Q7 дискретні Simulation-моделі будуються аналогічно з врахуванням граничних умов P_{w1} та P_{w4} , P_{w3} .

Гілка Q8, вузли P_{w4} , P_v :

$$\begin{cases} Q_{8,1}(i+1) = Q_{8,1}(i) + h[\alpha_8(P_{w4}(i) - P_{8,1}(i)) - \beta_8 Q_{8,1}(i)|Q_{8,1}(i)| \\ P_{8,1}(i+1) = P_{8,1}(i) + h\gamma_8(Q_{8,1}(i) - Q_{8,2}(i)) \end{cases}$$

.....

$$\begin{cases} Q_{8,15}(i+1) = Q_{8,15}(i) + h[\alpha_8(P_{8,14}(i) - P_{8,15}(i)) - \beta_8 Q_{8,15}(i)|Q_{8,15}(i)| \\ P_{8,15}(i+1) = P_{8,15}(i) + h\gamma_8[Q_{8,15}(i) - Q_{8,16}(i)] \\ \begin{cases} Q_{8,16}(i+1) = Q_{8,16}(i) + h[\alpha_8(P_{8,15}(i) - P_v) - \beta_8 Q_{8,16}(i)|Q_{8,16}(i)| \\ P_v = -4100 = \text{const} \end{cases} \end{cases}$$

3.4.2 Блок-схема і текст програми С-вирішувача рівнянь МДОРП, процеси $Q_j(v, t)$

На рис. 3.15 показано блок-схему С-програми МДОРП-вирішувача, текст програми – в додатку В4 (ДОДАТОК В4 - Текст С-програми вирішувача рівнянь МДОРП). Графіки отриманих перехідних процесів $Q_j(v, t)$ показано на рисунках 3.16 – 3.17.

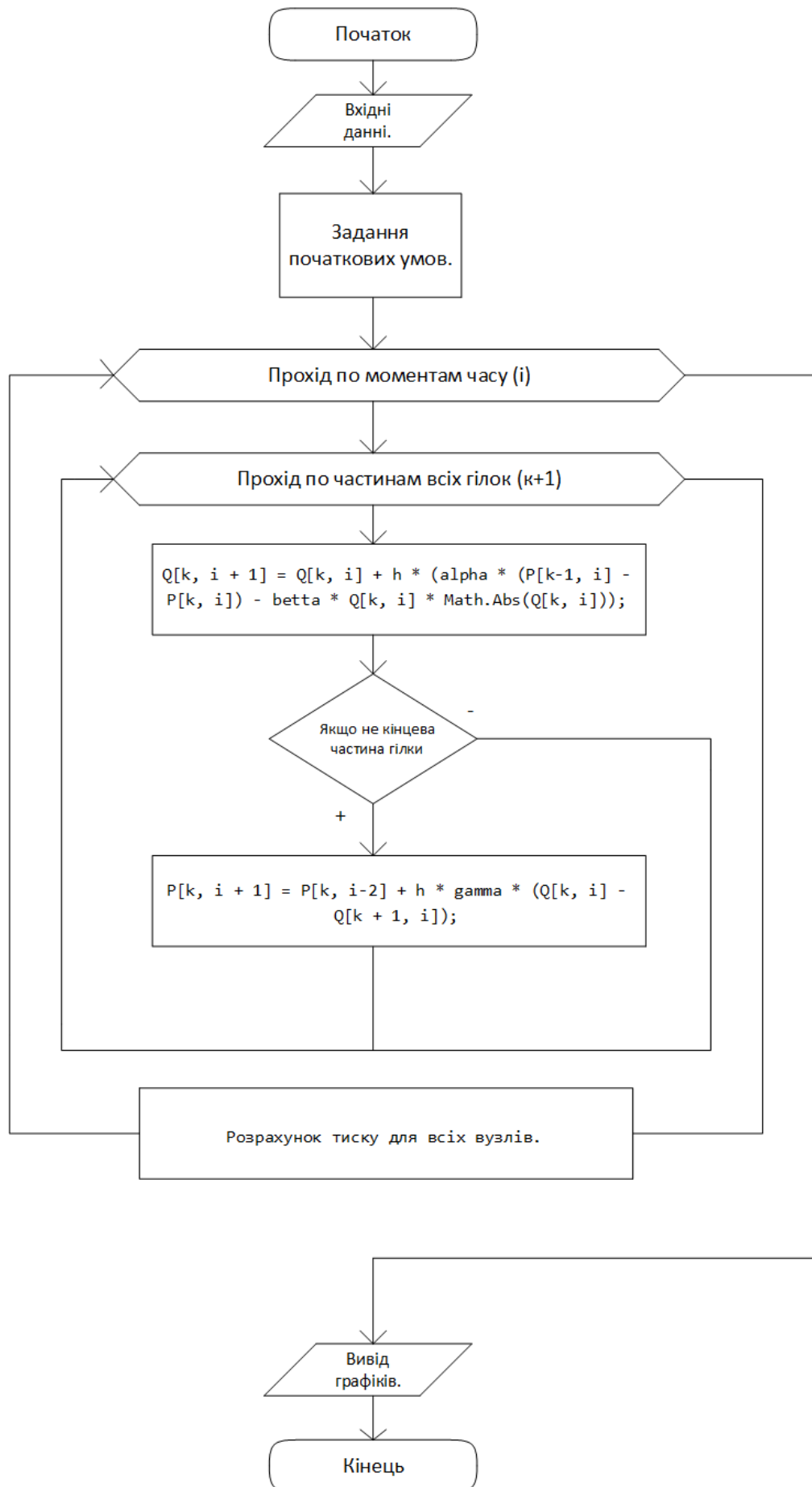


Рисунок 3.15 – Блок-схема С-вирішувача рівнянь МДОРП

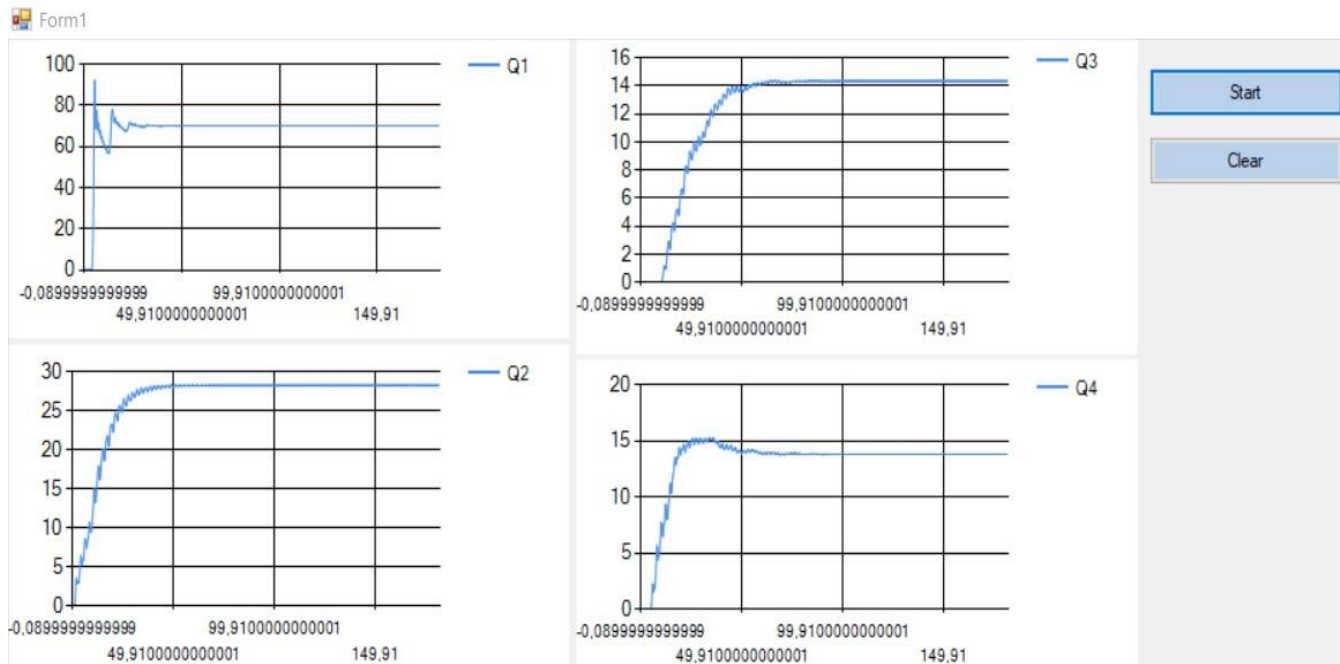


Рисунок 3.16 – Графіки процесів $Q1(v, t)$, $Q2(v, t)$, $Q3(v, t)$, $Q4(v, t)$

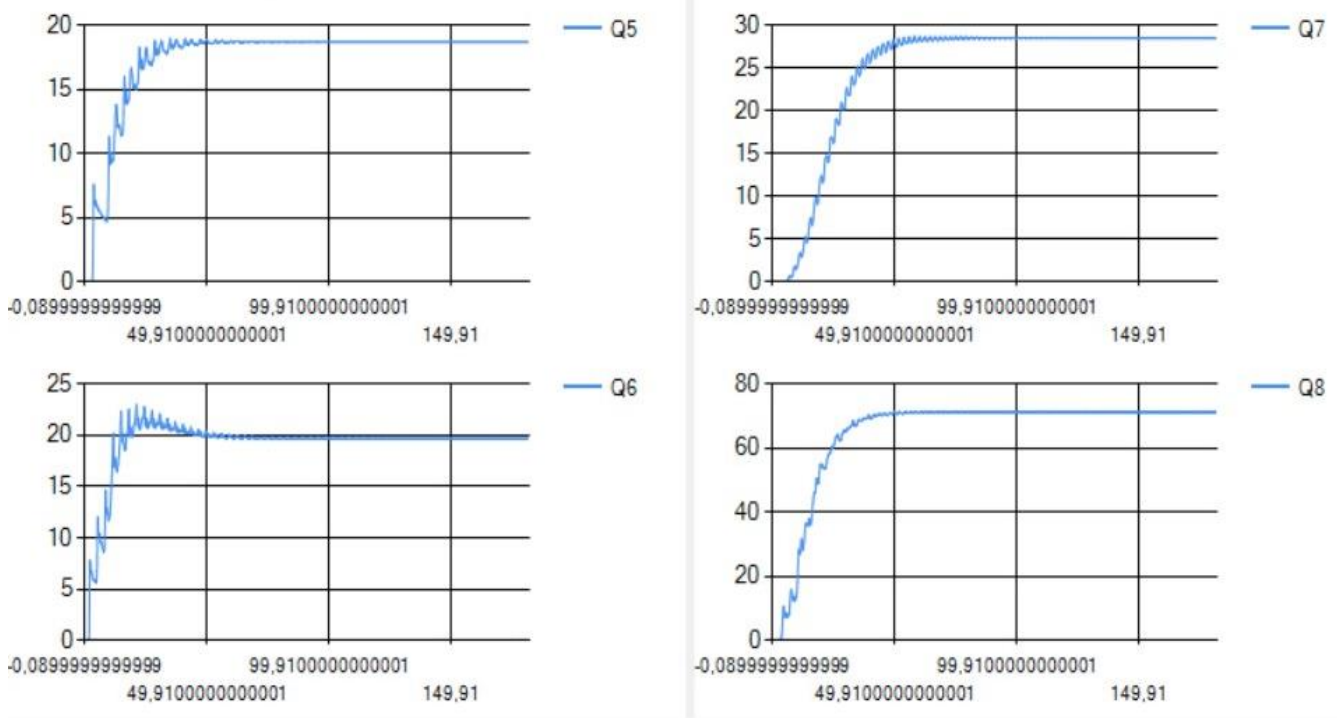


Рисунок 3.17 – Графіки процесів $Q5(v, t)$, $Q6(v, t)$, $Q7(v, t)$, $Q8(v, t)$

С-вирішувач рівнянь МДОРП дає результати, що відповідають фізичному змісту динамічних процесів. Після коливань потоки виходять на сталий режим, їх величини співпадають з тестовими значеннями, виконується динамічний «закон Кірхгофа» для вузлів.

3.5 Висновки

1) Згідно з вимогами до повнофункціонального програмного забезпечення проблемно орієнтованого паралельного моделюючого середовища та визначеними функціями підсистеми розроблено послідовні вирішувачі рівнянь на основі БО-мови моделювання і С-мови програмування.

2) Запропоновано й реалізовано методику побудови вирішувачів рівнянь від формування Simulation-моделей гілок та вузлів до отримання працездатних симуляторів як окремих гілок, так і всього МДОРП.

3) Отримані послідовні вирішувачі є основою для розробки паралельних симуляторів і засобами співставлення результатів паралельного і послідовного моделювання МДОРП та інших складних динамічних систем.

4 АЛГОРИТМИ ФУНКЦІОНУВАННЯ ПІДСИСТЕМИ ПАРАЛЕЛЬНИХ ВИРІШУВАЧІВ

4.1 Модель і Simulation-модель мережевого динамічного об'єкту з зосередженими параметрами.

В роботі [1] показано, що МДОРП може розглядатись і як об'єкт з зосередженими параметрами (МДОЗП). В цьому випадку j -гілка описується диференціальним рівнянням

$$K_j \frac{dQ_j}{dt} + R_j Q_j |Q_j| = H_j, \quad (4.1)$$

де $K_j = \frac{\rho L_j}{F_j}$ – коефіцієнт інерційності потоку; ρ – щільність повітря; L_j , F_j – довжина і площа поперечного перетину гілки-повітропроводу; R_j – аеродинамічний опір; Q_j – потік повітря; H_j – різниця тисків в початковому й кінцевому вузлах гілки (депресія).

Для побудови математичної моделі МДОЗП вводяться матриці інцидентів A та контурів S , вектор потоків повітря

$$Q = (X \ Y)^T, \quad (4.2)$$

діагональні матриці параметрів

$$K = \begin{pmatrix} K_x & 0 \\ 0 & K_y \end{pmatrix}, R = \begin{pmatrix} R_x & 0 \\ 0 & R_y \end{pmatrix}, \quad (4.3)$$

вектор депресій вентиляторів

$$H = (H_x, H_y)^T \quad (4.4)$$

Тут X, Y – підвектори потоків в гілках дерева й антидерева графа МДОЗП/РП. Модель МДОЗП в матрично-векторній формі описується системою алгебраїчних і диференціальних рівнянь [1, 2]

$$AQ = 0 \quad (4.5)$$

$$SK \frac{dQ}{dt} + SRZ = SH \quad (4.6)$$

В рівнянні (4.6) Z є вектором з елементами $Z_j = Q_j|Q_j|$, тобто $Z = (Z_x \ Z_y)^T$.

Simulation-модель МДОЗП – це модель у формі, яку вимагають обчислювальні методи розв’язання системи рівнянь (4.5), (4.6). Зробивши підстановки векторів і матриць в цю систему і виконавши відповідні перетворення [1], отримаємо рівняння Simulation-моделі:

$$\begin{cases} X = -W \cdot Y \\ \frac{dY}{dt} = TP \cdot H - RU \cdot Z \end{cases} \quad (4.7)$$

де

$$TP = (S_y \cdot K_y - S_x \cdot K_x) - 1 \cdot S, \quad RU = (S_y \cdot K_y - S_x \cdot K_x) - 1 \cdot S \cdot R, \\ W = Ax - 1 \cdot Ay$$

Алгоритм генерування системи (4.7) виражається за допомогою операцій над матрицями $A_x, A_y, S_x, S_y, K_x, K_y, R_x, R_y, S, R$ [1, 2]. Вирішувач рівнянь (ВР) повинний являти собою програму, що реалізує циклічний алгоритм чисельного розв’язання систем рівнянь Simulation-моделі.

4.2 Можливі підходи до розпаралелювання програми ВР.

Програма ВР розподіляється між n процесами $T_1, \dots, T_n$. Процес T_1 призначається керуючим процесом (Host-Prozess), який виконує певне число послідовних операцій, щоб підготувати до запуску процеси $T_2, \dots, T_n$ (рис. 4.1).

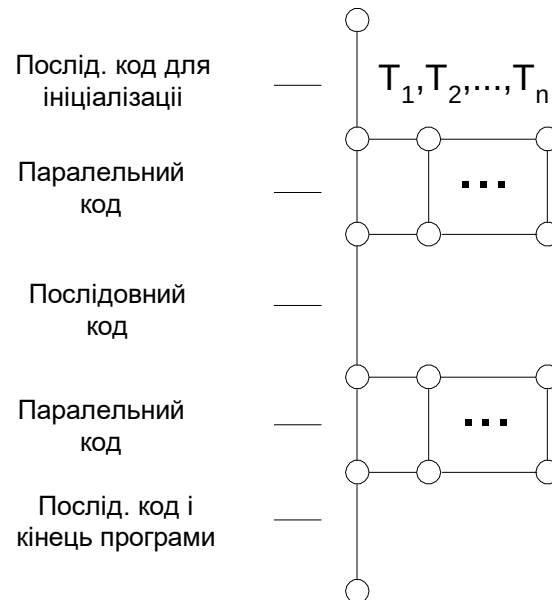


Рисунок 4.1 – *MIMD*-підхід до розпаралелювання.

Далі всі процеси працюють паралельно і кожен досягає закінчення своєї програми й операції об'єднання, що веде до синхронізації з процесом T_1 і до термінування процесів. Для кожної паралельної нитки в програмі керуючі потоки генеруються заново і закінчуються.

Генерування і детермінування керуючих потоків викликає великі витрати ресурсів. Їхня економія досягається в *SPMD*-моделі розпаралелювання (рис 4.2).

У *SPMD* – моделі паралельні процеси породжуються тільки один раз при запуску програми. Ці процеси термінуються наприкінці виконання програми.

Послідовні частини програми виконуються всіма процесами, тобто мова йде про багаторазове виконання послідовних кодів програми. Кожен процес виконує одну паралельну гілку програми над своїми локальними даними.

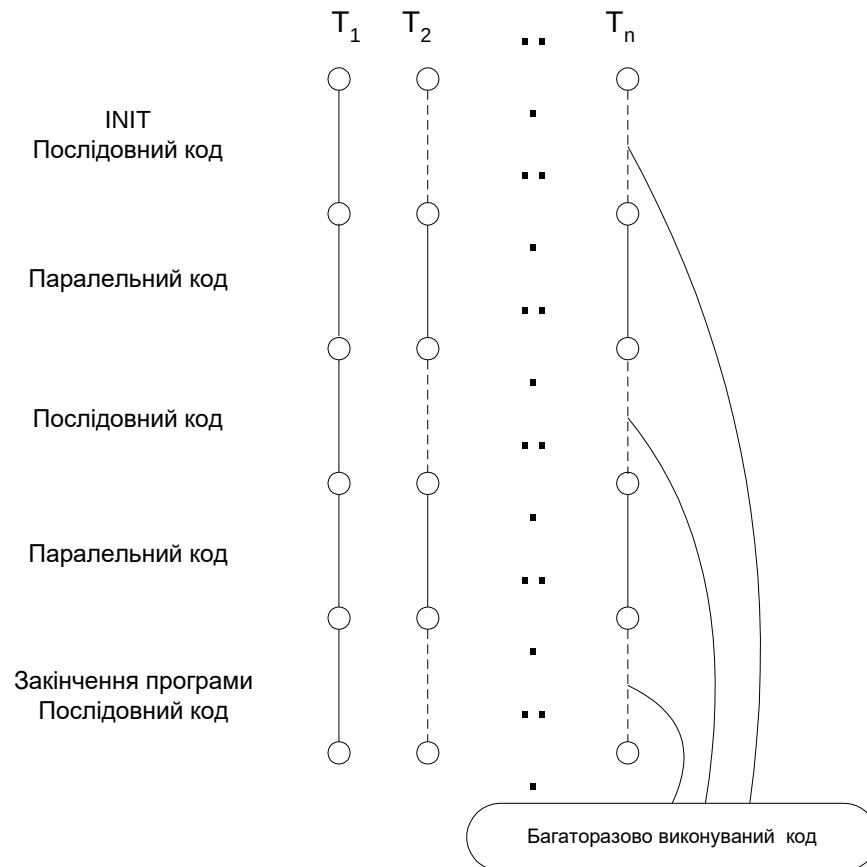


Рисунок 4.2 – *SPMD*-підхід до розпаралелювання.

За допомогою цього підходу вдається скоротити обсяг робіт, пов'язаних з багаторазовим генеруванням і термінуванням керуючих потоків (процеси) та виключити багаторазове виконання послідовних частин програми.

Основна ідея полягає в тому, що нові процеси генеруються, якщо вони запитуються в програмі, тобто до початку паралельної частини програми. Наприкінці паралельного блоку програми керуючі потоки переводяться в стан «недіючий» і до початку наступної паралельної ділянки знову активізуються. У такий спосіб послідовна частина програми виконується тільки Host-процесом.

Аналіз цих структур показує, що треба орієнтуватися на *SPMD*-організацію вирішувача, виконувати програму у декількох *MIMD*-процесорах [13]. При цьому в кожному процесорі будуть свої вхідні і кінцеві дані.

4.3 Дискретна Simulation-модель МДОЗП і послідовний вирішувач.

Скориставшись обчислювальним методом Адамса-Башфорта (п. 2.2), отримаємо наступні розрахункові формули дискретної Simulation-моделі МДОЗП:

$$\begin{aligned} Y(i+1) &= Y(i) + h * [TP * H(i) - RU * Z(i)] \text{ для } i = 0 \\ Y(i+1) &= Y(i) + 1.5h * [TP * H(i) - RU * Z(i)] - \end{aligned} \quad (4.8)$$

$$\begin{aligned} &- 0.5h * [TP * H(i-1) - RU * Z(i-1)] \text{ для } i > 0 \\ X(i) &= -W * Y(i) \end{aligned} \quad (4.9)$$

На рис.4.3 показано блок-схему програми виконання матрично-векторних операцій послідовного вирішувача системи рівнянь (4.8), (4.9).

4.4 Розпаралелювання за принципом «Х- та Y-процеси»

Призначимо для обчислення компонентів векторів X та Y n-1 X-процесів та m-(n-1) Y-процесів і доручимо кожному процесу виконувати програму за блок-схемою рис. 4.3. Блок-схема функціонування такого паралельного вирішувача має m процесів (рис. 4.4) і склад операційних блоків, що відповідає послідовній програмі (рис. 4.3). В таблиці 4.1 наведено зміст операцій, що мають виконуватись в послідовному і паралельному вирішувачах рівнянь. Паралельний ВР рис. 4.4 має два недоліки: нерівномірність завантаження процесів в зв'язку з тим, що обчислення компонент X-вектора проводиться за простішими формулами в порівнянні з Y-вектором; для МДОЗП/РП промислової складності може не вистачити ресурсів для реалізації вирішувача за принципом «1 процес – 1 процесор».

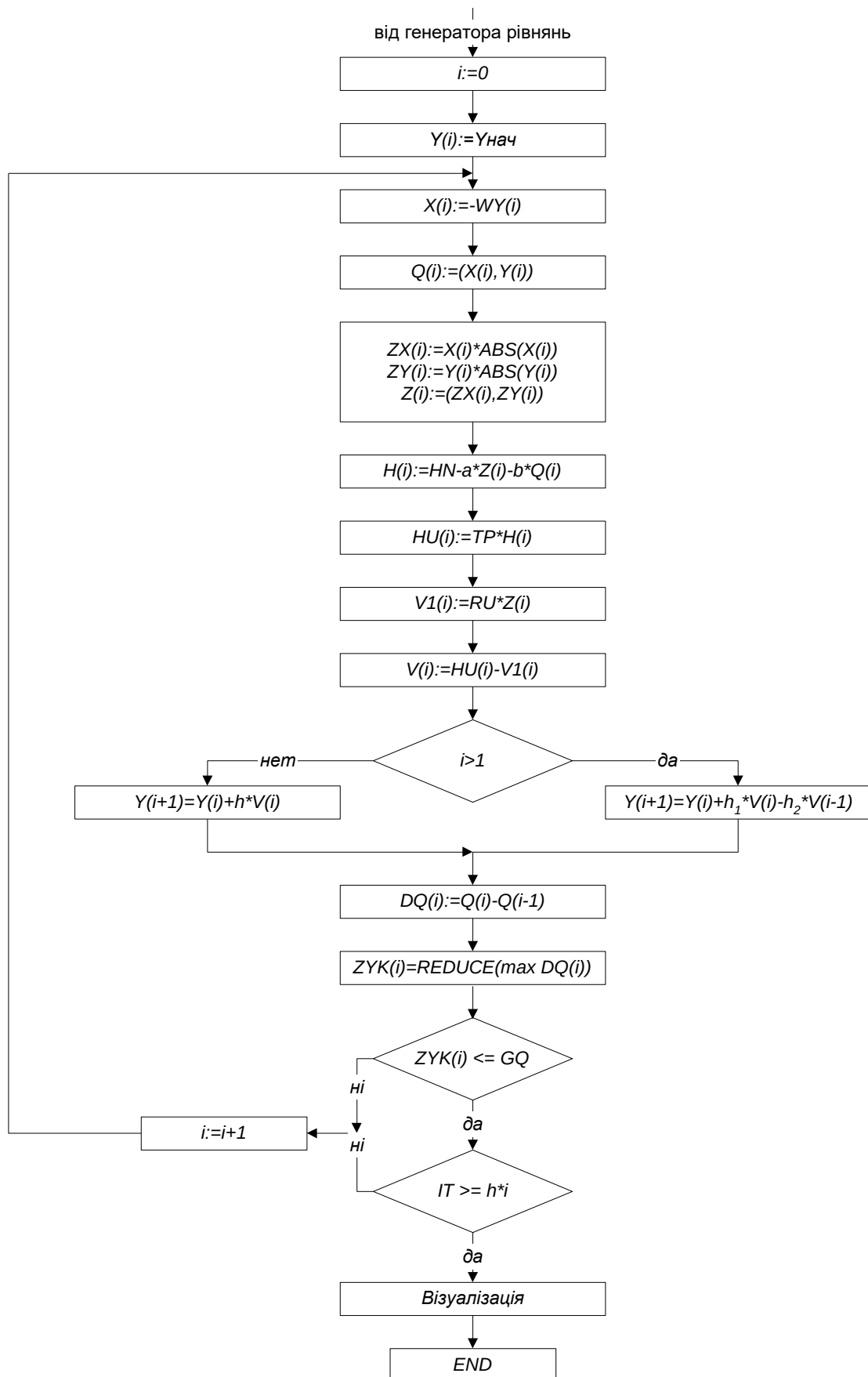


Рисунок 4.3 – Блок-схема програми послідовного ВР

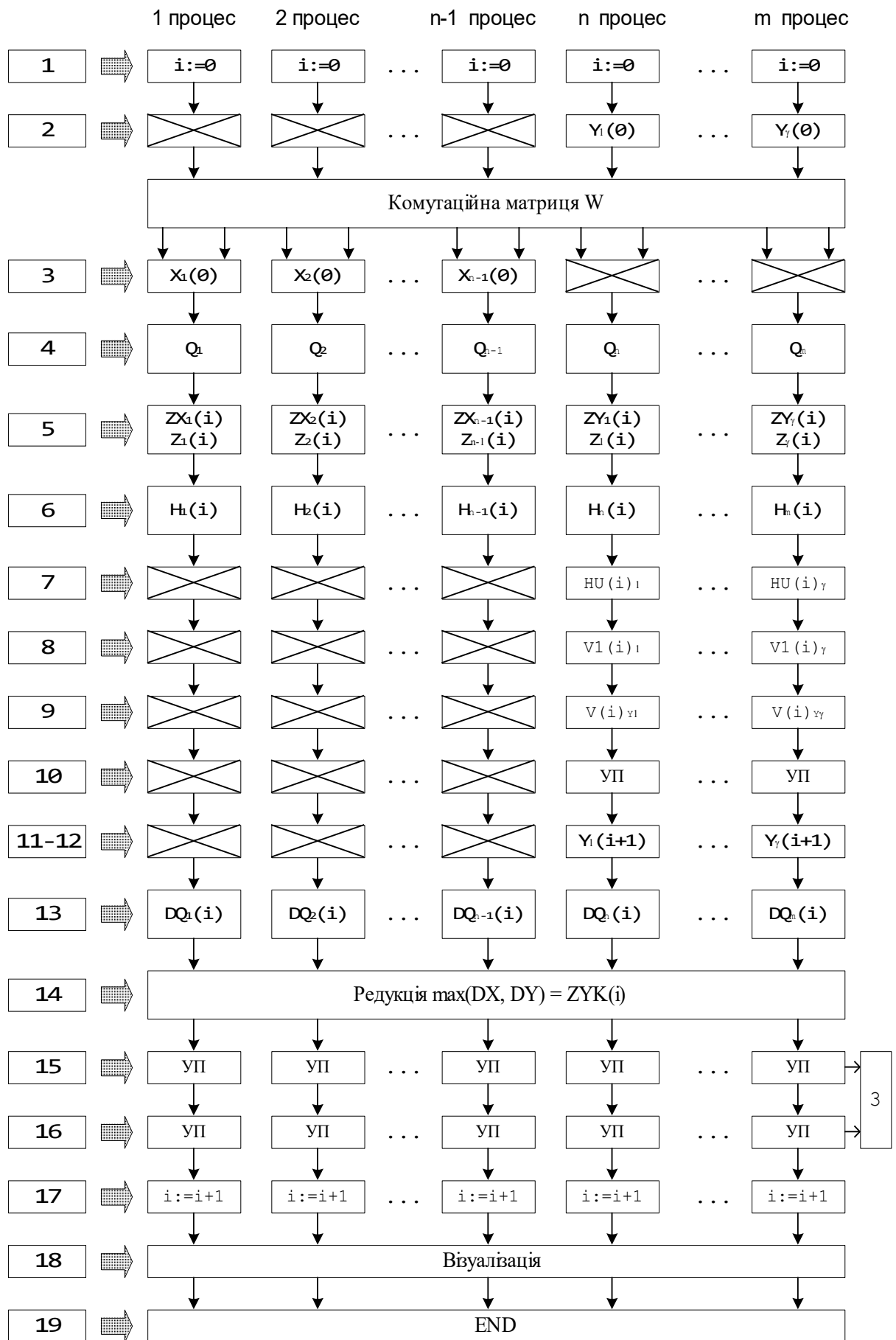


Рисунок 4.4 – Блок-схема програми паралельного ВР з X- та Y-процесами

Таблиця 4.1. Зміст операцій послідовного і паралельного ВР.

Номер блока	Виконувана операція	Опис
1	$i:=0$	Має сенс загального послідовного блока для усієї програми.
2	$Y(0):=YHAC$	Присвоєння компонентам вектора потоків Q у гілках антидерева початкових значень. Розмірність вектора Y - $\gamma \times 1$. У зв'язку з тим, що система диференціальних рівнянь будується відносно гілок антидерева графа і вектор Y утворює базу незалежних контурів, за основу узято процеси $T_1, T_2, \dots, T_\gamma$. Зв'язків між процесами немає, тому що вектор початкових умов задається ззовні та величини Y_k ($k=1, 2, \dots, \gamma$) безпосередньо між собою не зв'язані. Їх фізична реалізація у ШВМ повинна гарантуватися постановкою задачі моделювання. Операції $i:=0$ та $Y(0):=YHAC$ можна віднести до ініціалізації моделі.
3	$X(i):=-WY(i)$	Для $i=0$ знаходяться початкові значення витрат у гілках дерева $X(0):=-WY(0)$. Матриця W формується у генераторі рівнянь як $W:=A_X^{-1}A_Y$, має розмірність $(n-1) \times \gamma$ і дає у результаті операції множення вектор $X(0)$ розмірністю $[(n-1) \times \gamma][\gamma \times 1] := [(n-1) \times 1]$. Компоненти вектора X утворюються як результати по-компонентного множення рядків матриці W на вектор Y та додавання добутків компонент. Таким чином, блок 3 розвертається в процесах $T_{31}, T_{32}, \dots, T_{3\gamma}$. Слід зазначити, що якщо $(n-1) \neq \gamma$, то для зберігання $X(0)$ будуть брати участь не всі процеси γ , а лише ті, що задіяні для запам'ятовування підвектора X .
4	$Q(i):=(X(i), Y(i))$	Формує вектор Q у формі $Q:=(X_1, X_2, \dots, X_{n-1}, Y_1, Y_2, \dots, Y_\gamma)^T$. Вектор запам'ятовується парами X та Y і окремо залишковими Y або $X: (X_1; Y_1), (X_2; Y_2), \dots, (X_{n-1}; Y_{n-1}), (Y_n), (Y_\gamma)$. Паралельне зчитування буде проводитися як X ($X_1, X_2, \dots, X_{n-1}$) за допомогою процесів $T_{41}, T_{42}, \dots, T_{4(n-1)}$, а потім Y ($Y_1, Y_2, \dots, Y_\gamma$) процесами $T_{41}, T_{42}, \dots, T_{4\gamma}$.
5	$Z(i):=(ZX(i), ZY(i))$	За даними $X(i)$ та $Y(i)$ процеси обчислюють елементи вектора Z : $ZX_k(i):=X_k(i)*ABS(X_k(i))$, $k = 1, 2, \dots, n-1$; $ZY_j(i):=Y_j(i)*ABS(Y_j(i))$, $j = 1, 2, \dots, \gamma$. Результируючий вектор $Z:=(Z_X, Z_Y)^T$ зберігається у процесах $T_1, T_2, \dots, T_\gamma$ таким же чином, як і вектор Q .
6	$H(i):=HN-a*Z(i) - b*Q(i)$	Як і в блоці 5, по-компонентне знаходження $H(i)$. Аналогічно Q процеси запам'ятовують $H_X(i)$ та $H_Y(i)$.
7	$HU(i):=TP*H(i)$	Ця операція має розмірність $[HU(i)] = [\gamma \times m][m \times 1] = \gamma \times 1$. Таким чином, проводиться «пакунок» m по-компонентних операцій у γ операцій. «Пакунок» здійснюється топологічною матрицею TP . Оскільки вона має розмірність $\gamma \times m$, то в ній чітко видно частини TP_X та TP_Y . Одержуємо

Номер блока	Виконувана операція	Опис
		$HU(i)_Y := (TP_X + TP_Y) * (H_X(i) + H_Y(i)) \Rightarrow TP_X H_X(i) + TP_Y H_Y(i)$. Для кожного T_k маємо $HU(i)_k := TP_X H_X(i)_k + TP_Y H_Y(i)_k$.
8	$V1(i) := RU * Z(i)$	Ця операція має розмірність $[V1(i)] = [\gamma m][m \times 1] = \gamma \times 1$.
9	$V(i) := HU(i) - V1(i)$	Операція має розмірність $\gamma \times 1$ та виконується по-компонентно.
10	$i > 1$	У кожному процесі операція умовного переходу.
11	$Y(i+1) := Y(i) + h * V(i)$	Обчислення вектора $Y(i+1)$ для $i=1$. Виконується лише один раз. У процесах T_k є : 1. Елементи вектора $Y(0) := Y_{НАЧ}$. 2. Діагональна матриця кроків h . 3. Обчислений вектор $V(0)$. Обчислення ведуться паралельно γ процесами по-компонентно в розгорнутій формі : $Y_k(i+1) := Y_k(i) + h * V_k(i)$, $k=1, 2, \dots, \gamma$. Далі результат передається у блок 13.
12	$Y(i+1) := Y(i) + h1V(i) - h2V(i-1)$	Обчислення вектора $Y(i+1)$ для $i>1$. У процесах T_k є : 1. Елементи вектора $Y(i)$. 2. Діагональні матриці кроків $h1, h2$. 3. Обчислений вектор $V(i)$. 4. Обчислений на попередньому кроці вектор $V(i-1)$. Обчислення ведуться паралельно γ процесами по-компонентно в розгорнутій формі : $Y_k(i+1) := Y_k(i) + h1V_k(i) - h2V_k(i-1)$, $k=1, 2, \dots, \gamma$. Результат передається у блок 13.
13	$DQ(i) := Q(i) - Q(i-1)$	У процесах є рішення на i -ому та $(i-1)$ -ому кроках. Для здійснення умови закінчення циклу утворюються різниці $DX := X(i) - X(i-1)$; $DY := Y(i) - Y(i-1)$.
14	$ZY_k(i) := REDUCE(max(DQ(i)))$	Здійснюється операція редукції вектора.
15,16	$ZY_k(i) \leq GQ, i \geq h * i$	Блоки умовних переходів, здійснюються в кожному процесі, щоб не ускладнювати керування.
17	$i := i + 1$	Інкрементування ітерації, значення якої однакове для усіх процесів T_k .
18	Візуалізація	Здійснюється після переходу з блока 17, тобто після завершення циклу.

4.5 Розпаралелювання за принципом «XY-процеси»

Для збалансованого завантаження процесорів пропонується обчислювати пари компонент цих векторів одночасно, як показано на рис. 4.5.

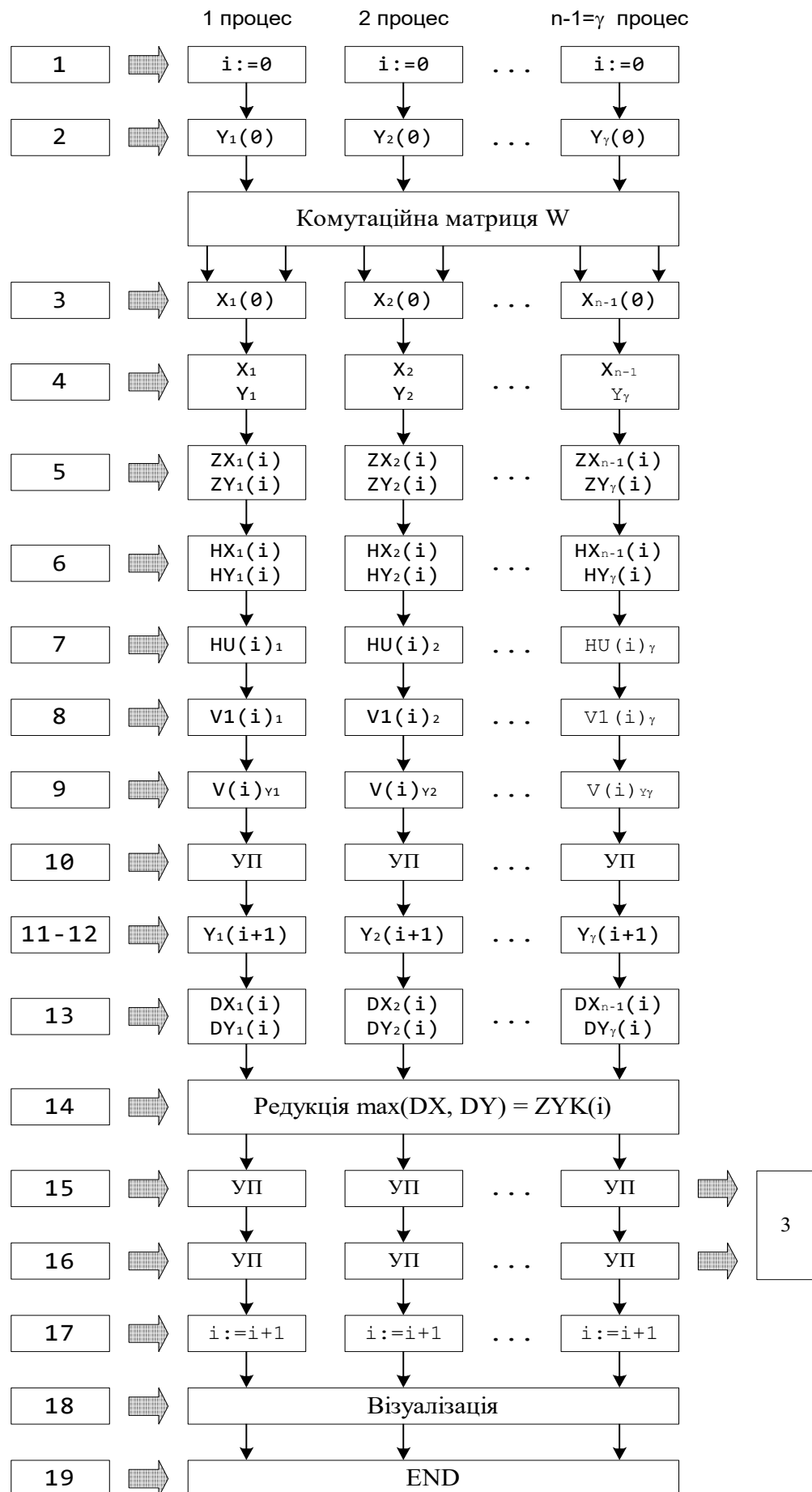


Рисунок 4.5 – Блок-схема програми паралельного ВР з ХУ-процесами

4.6 Підготовка до імплементації паралельного вирішувача.

Визначимо функції MPI (Message Passing Interface), необхідні для виконання матрично-векторних операцій, наведених в таблиці 4.1.

Ініціалізація матриці

```
int InputMatrix( char* FileName, Matrix Mat, const int m, const int n )
```

Ініціалізація вектора

```
int InputArray( char* FileName, Array Arr, const int n )
```

Елемент вектора, отриманий множенням i -ого рядка матриці на вектор

```
float CreateElement( Matrix Mat, Array Arr, int NoLine ).
```

Функції ініціалізації зчитують з підготовлених заздалегідь файлів дані і підпорядковують елементам векторів і матриць отримані дані.

Функція *CreateElement(...)* обчислює i -ий елемент вектора, де i – номер процесу (процесора), у якому проводяться обчислення.

В основній частині програми декларуються необхідні для роботи змінні і здійснюється ініціалізація MPI. Для цього необхідні наступні функції:

```
MPI_Init (&argc, &argv);
```

```
MPI_Comm_rank (MPI_COMM_WORLD, &my_rank);
```

```
MPI_Comm_size (MPI_COMM_WORLD, &p).
```

За допомогою цих функцій виділяється заявлена користувачем кількість процесорів і їм надаються логічні імена від 0 до $n-1$.

Після цього в 0 процесорі (процесі) відбувається ініціалізація початкових даних. Для більш зручної реалізації програми розмірності матриць визначимо як $[M][M]$, а масивів – $[M]$.

У той час як 0-ий процес зайнятий ініціалізацією, інші процеси знаходяться в стані чекання. Для того, щоб гарантувати запуск інших процесів тільки після закінчення роботи 0-го, використовується функція:

```
MPI_Barrier(MPI_COMM_WORLD),
```

тобто усі процеси чекають закінчення ініціалізації.

Алгоритм виконання програми передбачає, що значення матриць і векторів відомі не тільки 0-му, але і всім іншим процесам. Для передачі повідомлень служить функція *Broadcast*:

*MPI_Bcast(W, M*M, MPI_FLOAT, 0, MPI_COMM_WORLD).*

Тут же варто згадати, що 0-ий процес посилає матрицю *W* всім іншим процесам.

Функція *MPI_Barrier(MPI_COMM_WORLD)* як і раніш дозволяє гарантувати закінчення всіх операцій *Broadcast* до початку наступного програмного сегмента.

Отримані в результаті моделювання дані повинні бути збережені для наступного аналізу і візуалізації. Для цього кожен процес відкриває файли з іменами *OutFile"номер процесу"* і записує туди свої дані.

Після проведення підготовчих операцій починається безпосереднє рішення рівнянь (їх інтегрування). Спочатку обчислюються компоненти вектора *X* (з назвою *localX*) та векторів *Z_x* і *Z_y* (відповідно *localZ*). Потім у кожному процесі обчислюється права частина диференціального рівняння (змінна *Vlocal*), що використовується для інтегрування системи рівнянь за методом Адамса-Башфорта. Процес інтегрування ведеться з кроком *h* до моменту часу *ITl* чи до виходу на стаціонарний режим. Для перевірки умови стаціонарності ($\max(\Delta X, \Delta Y) \leq \Delta$), де ΔX і ΔY максимальна зміна компонентів векторів *X* і *Y* між *i* і *i + 1* кроками інтегрування, використовується ще одна функція *MPI*:

MPI_Allreduce(&localDQ, &localZDK, 1, MPI_FLOAT, MPI_MAX, MPI_COMM_WORLD).

У даному випадку ця функція знаходить максимальне значення серед перемінних *localDQ* і зберігає його в змінній *localZDK* у кожному процесі.

Формування і розсилання вектора *Y* відбувається за допомогою функції *MPI*:

MPI_Allgather(&localY, 1, MPI_FLOAT, Buf, 1, MPI_FLOAT, MPI_COMM_WORLD).

Ця функція збирає всі значення *localY* в один масив і розсилає всім процесам.

Після закінчення інтегрування закриваються файли заданих даних і виконується завершальна для всіх процесів функція *MPI_Finalize()*.

4.7 Априорний аналіз ефективності паралельних вирішувачів на рівні "Гілка МДОРП - MIMD-процес"

Як показано в розділі 2, динамічні процеси в МДОРП можуть розглядатись на чотирьох рівнях паралельності. Послідовні С-вирішувачі гілок і МДОРП, побудовані в розділі 3, мають готові програмні модулі, про паралельне виконання яких має йти мова. Маючи С-вирішувачі для гілок, розглянемо наступні питання роботи МДОРП-вирішувача рівня РП-3 «j-гілка – MIMD-процес »:

- Оцінка об'єму операцій в віртуальному процесі;
- Порівняння процесів за критерієм «Величина завантаження процесу»;
- Пропозиції щодо можливо потрібного балансування завантаження;
- Аналіз співвідношення між операціями обчислень і обміну даними;
- Оцінка очікуваного прискорення обчислень паралельного вирішувача;
- Можливість реалізації в цільовій паралельній обчислювальній системі за принципом «1 процес – 1 процесор»

4.7.1 Розподіл завантаження між віртуальними процесами і оцінка нерівномірності

МДОРП-С-вирішувач рівня РП-3 має складатись з m QR-процесів, які вирішують системи рівнянь Simulation- моделі відносно змінних Q_{jv} , P_{jv} гілок

($v=1,2,\dots, M_j$) та ще n P_w -процесів для розв'язання вузлових рівнянь. Завантаження процесів L залежить від кількості розв'язуваних рівнянь типу (3.2) і на рівні гілки складає

$$L_j = 2M_j = \frac{2l_j}{\Delta\xi}. \quad (4.10)$$

У зв'язку з різною довжиною гілок від $l_{\min}$ до $l_{\max}$ завантаження відповідних процесів має діапазон

$$L_{\min} \leq L_j \leq L_{\max} \quad (4.11)$$

Нерівномірність завантаження можна оцінити як

$$\Delta L = L_{\max} - L_{\min} = 2(M_{j\max} - M_{j\min}) \quad (4.12)$$

Якщо тривалість виконання QR-процесу позначити як T_{QR} , то максимальна тривалість очікування $T_{W\max}$ для всього МДОРП складе

$$T_{W\max} = \Delta L \cdot T_{QR} \quad (4.13)$$

Тривалість очікування процесу j -гілки відносно процесу $l_{\max}$ -гілки буде

$$T_{Wj} = \Delta L_j \cdot T_{QR} \quad (4.14)$$

Тут маємо $\Delta L_j = L_{\max} - L_j$.

4.7.2 Порівняння об'ємів обчислювальних і допоміжних операцій, віртуальне прискорення обчислень паралельного вирішувача.

На рівні РП-3 кількість операцій обміну даними менша, ніж на рівнях РП-1 і РП-2, тобто співвідношення між об'ємами обчислювальних і допоміжних операцій покращується. Розглянемо це співвідношення для j -гілок з посиланням на гілку Q_1 (тестовий МДОРП рис. 3.1) Q_j -процес вирішує $2M_j$ рівнянь і виконує наступні допоміжні операції:

- Ввід P_{j0} з вузлового процесу того вузла, який є початковим для j -гілки (для Q_1 : $P_{10} = P_A$ – атмосферний тиск);
- Ввід потоків початкових елементів тих гілок, які інцидентні кінцевому вузлу j -гілки (для Q_1 – вузол w_1 і потоки Q_{21} , Q_{51} , Q_{61});
- Відправка Q_{j1} процесу початкового вузла (для Q_1 не виконується);
- Відправка тиску $P_{jmj} = P_{wi}$, обчисленого j -QP-процесом по рівнянню для вузла w_i , що співпадає з кінцевим пунктом j -гілки (для $j = 1$: $P_{1M1} = P_{w1}$), тим процесам, які обчислюють потоки Q_{j1} в початкових елементах гілок, інцидентних вузлу w_i (для $j = 1$ це потоки Q_{21} , Q_{51} та Q_{61} : рис. 3.1).

Таким чином, на кожні $N_{RO} = 8M_j$ обчислювальних операцій для правих частин рівнянь типу (3.2)

$$\left. \begin{aligned} \frac{dQ_{jv}}{dt} &= \alpha_j(P_{jv-1} - P_{jv}) - \beta_j Q_{jv} |Q_{jv}| \\ \frac{dP_{jv}}{dt} &= \gamma_j(Q_{jv} - Q_{jv+1}) \end{aligned} \right\} \quad (4.15)$$

має виконуватись $N_{DA} = 4$ допоміжних операцій, тобто маємо співвідношення:

$$M_j \leq N_{RO}/N_{DA} \leq 4M_j. \quad (4.16)$$

В [17] показано, що для РП-1 це співвідношення виглядає як $1 \leq N_{RO}/N_{DA} \leq 2$, а для РП-2 – як $2 \leq N_{RO}/N_{DA} \leq 4$. Таким чином, на рівні РП-3 це важливе співвідношення покращується в M_j раз для всіх гілок.

Аналіз показує, що час (тривалість) циклу розв'язання (LZ – Lösenszeit) рівнянь j -гілок на рівні РП-3 можна оцінити формулою:

$$T_{jLZ} = M_j T_{QPj}, \quad (4.17)$$

де T_{QPj} – тривалість j -QR-процесу, який розв'язує пару рівнянь (4-15). Реалізація m процесів в віртуальній однопроцесорній системі потребує для всього МДОРП наступну тривалість обчислень:

$$T_{LZ}(1) = \sum_{j=1}^m M_j T_{QPj}. \quad (4.18)$$

При паралельній реалізації на віртуальній m -процесорній системі ця тривалість залежить від тривалості процесу для $l_{\max}$ -гілки і часу T_{DA} обміну даними $T_{3DA}^{(1)}$ (РП-3, 1- базовий варіант «гілка – процес», DA - Datenaustausch):

$$T_{LZ}(m) = \mathbf{MAX}_{j=1}^m (M_j T_{QPj}) + T_{3DA}^{(1)}. \quad (4.19)$$

Віртуальне прискорення на рівні РП-3 визначимо як

$$S_{v3}^{(1)} = \frac{T_{LZ}(1)}{T_{LZ}(m)} = \frac{\sum_{j=1}^m M_j T_{QPj}}{\mathbf{MAX}_{j=1}^m (M_j T_{QPj}) + T_{3DA}^{(1)}}. \quad (4.20)$$

Операція MAX виконується стосовно параметрів МДОРП (табл. 3.1) і дає величину

$$\mathbf{MAX}_{j=1}^m (M_j T_{QPj}) = M_{j_{\max}} T_{QPj_{\max}}$$

З врахуванням цього результату можна трансформувати формулу (4.20) так:

$$S_{v3}^{(1)} = \frac{1 + (\sum_{k=1}^{m-1} M_k T_{QPk}) / M_{jmax} T_{QPjmax}}{1 + \frac{T_{3DA}^{(1)}}{M_{jmax} T_{QPjmax}}}, \quad (4.21)$$

де $k \neq jmax$ та $jmax$ є номер l_{max} -гілки.

З формули (4.21) слідує, що організація обміну є вирішальним фактором ефективності підходів до розпаралелювання вирішувачів рівнянь. Максимальна кількість операцій обміну на рівні РП-3 залежить від кількості вузлів. В найнесприятливішому випадку послідовної організації обміну отримаємо:

$$T_{3DA}^{(1)} = n \cdot T_{UDA} \cdot T_{3DA} = n \cdot T_{wDA} \quad (4.22)$$

При паралельному одночасному виконанні цих операцій на всіх вузлах вирішувач буде мати

$$T_{3DA}^{(1)} = T_{UDA}, \quad T_{3DA} = T_{wDA}$$

де T_{wDA} – це час виконання операцій обміну процесом QR-j-гілки, який обчислює вузловий тиск P_{wi} ($i = 1, 2, \dots, n$).

4.7.3 Можливість реалізації за принципом «процес – процесор»

На рівні РП-3 для реалізації паралельного С-вирішувача потрібно

$$N_{РП-3} = m \quad (4.23)$$

процесорів. Крок апроксимації Δx впливає на множину обчислювальних операцій в процесах і на оцінку прискорення (4.21). Кількість процесорів

(обчислювальних вузлів) в сучасних MIMD-системах дозволяє реалізувати цей підхід. Якщо промисловий МДОРП має кількість гілок, що перевищує число наявних обчислювальних вузлів, то треба скористатись рівнем РП-4 або комбінацією розглянутих підходів.

4.7 Висновки.

1) Розроблено алгоритми функціонування підсистеми паралельних вирішувачів рівнянь (ВР) мережевого динамічного об'єкту з розподіленими й зосередженими параметрами (МДОРП/ЗП).

2) Розглянуто можливі підходи до розпаралелювання програми ВР, запропоновано дискретну Simulation-модель МДОЗП і послідовний вирішувач.

3) Розроблено блок-схеми паралельних вирішувачів з розпаралелюванням за принципами «Х- та Y-процеси», «ХY-процеси», проведено підготовку до імплементації паралельного вирішувача з визначенням потрібних операцій MPI-стандарту.

4) Виконано апріорний аналіз ефективності паралельних вирішувачів на рівні «Гілка МДОРП - MIMD-процес» за показниками «Розподіл завантаження між віртуальними процесами і оцінка нерівномірності», «Порівняння об'ємів обчислювальних і допоміжних операцій, віртуальне прискорення обчислень паралельного вирішувача», «Можливість реалізації з підпорядкуванням віртуальних процесів процесорам цільової паралельної обчислювальної системи»

ЗАГАЛЬНІ ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1) Дано визначення підсистеми вирішувачів рівнянь, сформульовано функціональність, розроблено структуру підсистеми. Розглянуто формальні підходи до побудови дискретних Simulation-моделей МДОРП, на основі яких мають розроблюватись вирішувачі рівнянь та симулятори.

2) Розробки вирішувачів зорієнтовано на можливі рівні розпаралелювання Simulation-моделей МДОРП.

3) Згідно з вимогами до повнофункціонального програмного забезпечення проблемно орієнтованого паралельного моделюючого середовища та визначеними функціями підсистеми запропоновано й реалізовано методику побудови вирішувачів рівнянь від формування Simulation-моделей гілок та вузлів до отримання працездатних симуляторів як окремих гілок, так і всього МДОРП на основі БО-мови моделювання і С-мови програмування. Отримані послідовні вирішувачі є основою для розробки паралельних симуляторів і засобами співставлення результатів паралельного і послідовного моделювання МДОРП та інших складних динамічних систем.

4) Розроблено алгоритми функціонування підсистеми паралельних вирішувачів рівнянь (ВР) мережевого динамічного об'єкту з розподіленими й зосередженими параметрами (МДОРП/ЗП). Розглянуто можливі підходи до розпаралелювання програми ВР, запропоновано дискретну Simulation-модель МДОЗП і послідовний вирішувач.

5) Розроблено блок-схеми паралельних вирішувачів з розпаралелюванням за принципами «Х- та Y-процеси», «XY-процеси», проведено підготовку до імплементації паралельного вирішувача з визначенням потрібних операцій МРІ-стандарту.

6) Виконано апріорний аналіз ефективності паралельних вирішувачів на рівні «Гілка МДОРП - MIMD-процес» за показниками «Розподіл завантаження між віртуальними процесами і оцінка нерівномірності», «Порівняння об'ємів обчислювальних і допоміжних

операцій, віртуальне прискорення обчислень паралельного вирішувача», «Можливість реалізації з підпорядкуванням віртуальних процесів процесорам цільової паралельної обчислювальної системи».

Подальші розробки й дослідження вирішувачів рівнянь мають бути зорієнтованими на їх функціонування на всіх рівнях розпаралелювання, на взаємодію підсистеми вирішувачів з іншими підсистемами та експериментальну реалізацію на кластерах ДонНТУ і Штутгартського університету, на організацію застосувань в Українському дослідницько-навчальному центрі технологій паралельного моделювання [18, 19].

ПЕРЕЛІК ПОСИЛАНЬ

1. Святный В. А. Моделирование аэрогазодинамических процессов и разработка систем управления проветриванием угольных шахт. Дисс. докт. техн. наук, ДПИ, Донецк, 1985, 407 с.
2. Абрамов Ф. А., Фельдман Л. П., Святный В. А. Моделирование динамических процессов рудничной аэрологии. К. – Наук. думка, 1981, 284 с.
3. Galasov R., Lapko V., Pererva A., Rasinkov D., Svjatnyj V.:
Parallele Simulation von industriellen Grubenbewetterungsnetzen. In: Hohmann, G. (Hrsg.): Tagungsband 13. ASIM-Symposium Simulationstechnik, Weimar 1999, 343-348.
4. Молдованова О. В. Способи організації паралельних обчислень в задачах моделювання шахтних вентиляційних мереж. Дис. к. т. н. ДонНТУ, Донецьк, 2008,- 170 с.
5. Svjatnyj V.A., Nadeev D.V., Solonin O. M., Rothermel K., Zeitz M.:
Subsysteme einer verteilten parallelen Simulationsumgebung für dynamische Systeme. 16. Symposium ASIM 2002, Tagungsband, 2002, S. 64 – 69.
6. Масюк А. Л. Інтерактивні засоби паралельного моделюючого середовища, орієнтованого на мережеві динамічні об'єкти. Дис. к. т. н. ДонНТУ Покровськ, 2018,- 168 S.
7. Теплинський К. С.: Паралельне моделююче середовище, орієнтоване на ідентифікацію параметрів моделей складних динамічних систем. Дис. к. т. н., ДонНТУ Покровськ, 2016, - 130 S.
8. Svyatnyy V.A., Kushnarenko V.G., Resch M., Miroshkin O.M., Wesner S.:
Problematik der parallelen Simulationstechnik. Наукові праці ДонНТУ №2 (23), 2016 Серія “Інформатика, кібернетика та обчислювальна техніка”, ISSN 1996-1588, Покровськ, 2016, С. 5-20.
9. Svyatnyy V.A., Kushnarenko V.G., Miroshkin O.M.: Parallele Modellierung und Simulation der luftdynamischen Prozesse in den

Grubenbewetterungs-netzen. In: Вісті Донецького гірничого інституту, №1(42), 2018, с. 116-129.

10. Svyatnyy V.A., Kushnarenko V.G., Miroshkin O.M., Resch M., Wesner S.: Virtuelle parallele Simulationsmodelle und ein Devirtualisierungsvorgang der Entwicklung von MIMD-Simulatoren für dynamische Netzobjekte mit verteilten Parametern. Tagungsband der Konferenz SIMULATION-2018, Kyiw, ISBN 978-966-02-8587-3, S. 55-66 (Plenarvortrag).

11. Svjatnyj, V. Dekomposition der verteilten parallelen Simulationsumgebung / V. Svjatnyj, V. Kushnarenko, O. Shcherbakov, M. Resch // Наукові праці Донецького національного технічного університету. Серія: «Проблеми моделювання та автоматизації проектування» № 1(10) - 2(11) – Донецьк: ДонНТУ, 2012.

12. Svyatnyy V., Kushnarenko V. Ein Ansatz zur gleichmäßigen Lastverteilung zwischen Prozessoren des MIMD-Simulators der dynamischen Netzobjekte mit verteilten Parametern. Наукові праці ДонНТУ, серія «Проблеми моделювання та автоматизації проектування», № 1 (15), 2020, с.5-14.

13. Бройнль Т. Паралельне програмування (переклад з німецької В. А. Святного). Київ, Вища школа, 1997.

14. Метод Ейлера для розв'язання звичайних диференціальних рівнянь, 2012, [online]. Available at:<<https://www.mathros.net.ua/metod-ejlera-dlja-rozvjazannja-zvyhajnyh-dyferencialnyh-rivnjan.html>> [Дата звернення: 06.10.2022].

15. Метод Рунге-Кутты как класс численных методов решения задачи Коши, блок-схема и примеры решений и реализации, [online]. Available at:<<https://intellect.icu/metod-runge-kutty-kak-klass-chislennykh-metodov-resheniya-zadachi-koshi-blok-skHEMA-i-primery-reshenij-i-realizatsii-5249>>[Дата звернення: 09.10.2022].

16. Метод Адамса, [online]. Available at: <<https://intellect.icu/metod-adamsa-8575>> [Дата звернення: 09.10.2022].

17. Svjatnyj V. Virtuelle parallele Simulationsmodelle und ein Devirtualisierungsvorgang der Entwicklung von parallelen Simulatoren für dynamische Netzobjekte mit verteilten Parametern. HLRS, Universität Stuttgart, Forschungsbericht-2008.

18. Мірошкін О. М. Системна організація ресурсів обчислювального кластеру Донецького національного технічного університету. Наукові праці ДонНТУ, серія «Проблеми моделювання та автоматизації проектування», № 1 (17), 2021, с.13-25.

19. A. S. Tanenbaum, M. van Stehen. Verteilte Systeme: Prinzipien und Paradigmen. Pearson Studium, München, 2008, 761 S.

ДОДАТОК А – Заява на затвердження теми і керівника роботи

Завідувачу кафедри
Електронної техніки
Вовні Олександрову Володимировичу
студента 2 курсу, групи КІМ-21,
спеціальності 123 Комп'ютерна інженерія
Литвинова Микити Олександровича

ЗАЯВА

Прошу призначити керівником моєї магістерської роботи
д. т. н, професора кафедри ЕТ Святного Володимира Андрійовича

(вчена ступінь, вчене звання, посада, П.І.Б викладача)

і закріпити за мною наступну тему:

«Підсистема вирішувачів рівнянь проблемно орієнтованого паралельного
моделюючого середовища»

(найменування теми)

« __ » __ 2022 р.

(підпис студента)

ПОГОДЖЕНО:

Науковий керівник



Святний В. А.

ДОДАТОК Б – Охорона праці

ОХОРОНА ПРАЦІ В УКРАЇНСЬКОМУ ДОСЛІДНИЦЬКО-НАВЧАЛЬНОМУ ЦЕНТРІ ТЕХНОЛОГІЙ ПАРАЛЕЛЬНОГО МОДЕЛЮВАННЯ (ДОННТУ, М. ПОКРОВСЬК)

1. Заходи щодо створення сприятливих умов при роботі з персональним комп'ютером (ПК)

Згідно з НПАОП 0.00-1.31-99 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», експлуатація персонального комп'ютера супроводжується впливом на організм комплексу факторів трудового процесу та виробничого середовища. У результаті настають зміни функціонального стану центральної системи організму, що характеризуються вираженою напругою. Зміни, що накопичуються в процесі роботи, призводять до ризику розвитку певних захворювань. З метою мінімізації впливу негативних факторів на організм людини при роботі із ПК варто дотримуватися правил охорони праці при роботі із ПК. На рис. 5.1 наведено схему приміщення лабораторії.

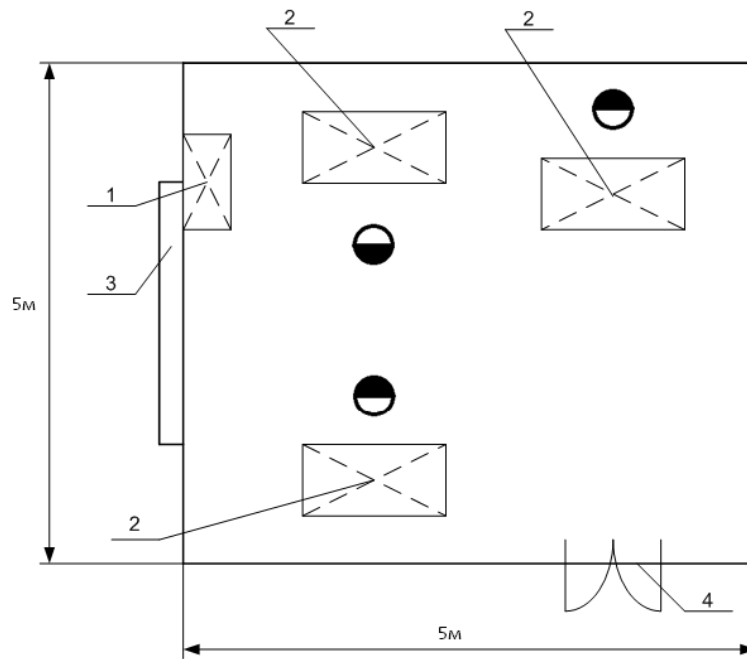


Рисунок 5.1 – Схема приміщення лабораторії

Умовні позначення: 1-кондиціонер; 2-комп'ютер (3 шт.); 3-вікно; 4-двері;

У приміщеннях, в яких перебувають робочі місця з ПК, є освітлення, з однобічним розміщенням світопрорізів, площа яких не перевищує 25% від площі стіни. Віконні прорізи в приміщенні з ПК мають регульовані жалюзі, або штори, або інші сонцезахисні пристрої.

Конструкцію робочого місця згідно з ГОСТ 12.2.032-78, «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования», маємо забезпечити підтримання оптимальної робочої пози з такими ергономічними характеристиками: ступні ніг – на підлозі або на підставці для ніг; стегна – в горизонтальній площині; передпліччя - вертикально; лікті – під кутом 90 градусів до вертикальної площини; зап'ястя зігнуті під кутом не більш ніж 20 градусів відносно горизонтальної площини; нахил голови 15 - 20 градусів відносно вертикальної площини.

Екран та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів, загалом 600-1000 мм.

Клавіатуру слід розміщувати на поверхні столу або на спеціальній, регульованій за висотою, робочій поверхні окремо від столу на відстані 100 - 300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути в межах 15 градусів.

Висота робочої поверхні столу має бути 800 мм, а ширина – повинна забезпечувати можливість виконання операцій в зоні досяжності моторного поля.

Робочий стіл повинен мати простір для ніг висотою не менше 600 мм, шириною не менше 500 мм, глибиною на рівні колін не менше 450 мм, на рівні витягнутої ноги – не менше 650 мм.

Ширина та глибина сидіння повинні бути не меншими за 400 мм. Висота поверхні сидіння має регулюватися в межах 400 - 500 мм, а кут нахилу поверхні – від 15 градусів вперед до 5 градусів назад.

Висота спинки сидіння має становити 300 ± 20 мм, ширина – не менше 380 мм, радіус кривизни в горизонтальній площині – 400 мм. Кут нахилу спинки повинен регулюватися в межах 0 - 30 градусів відносно вертикального положення. Відстань від спинки до переднього краю сидіння повинна регулюватись у межах 260 - 400 мм.

Для зниження статичного напруження м'язів рук необхідно застосовувати стаціонарні або знімні підлокітники довжиною не менше 250 мм, шириною – 50 - 70 мм, що регулюються по висоті над сидінням у межах 230 ± 30 мм та по відстані між підлокітниками в межах 350 - 500 мм.

У конструкцію сидіння можуть бути введені додаткові елементи, що не є обов'язковими: підголівник та підставка для ніг.

У приміщеннях, де розташовано монітори, потрібно виконувати заходи щодо боротьби зі статичною електрикою. Найбільш простим способом відповідно до рекомендацій, є підтримка відносної вологості повітря на рівні 50-60% за допомогою кондиціонера з функцією клімат-контроля.

Згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» у приміщеннях з ПК параметри мікроклімату повинні бути такими: температура навколишнього середовища повинна бути в межах 25-22°C, а швидкість руху повітря – 0,1- 0,2м/с.

2. Розрахунок кондиціювання повітря.

Мікроклімат впливає на самопочуття людини, його працездатність і протікання фізіологічних процесів, від яких залежить підтримка постійності температури тіла. Теплові дії на організм можуть з'явитися причиною швидкого стомлення, зниження працездатності, ослаблення, опірності організму до шкідливих дій, різних захворювань.

Кондиціювання повітря – це створення та автоматична підтримка в приміщеннях постійних чи визначених метеорологічних умов, що змінюються по програмі, найбільш сприятливих для людей, що працюють чи необхідних для нормального протікання технологічного процесу.

При виділенні тепла у виробничому приміщенні визначають необмежений повітрообмін виходячи з рівності виділеного тепла в помешканні та виділення вентиляцією.

$$L = \frac{Q_{изб}}{c \cdot p_n \cdot (t_{вуд} - t_{np})}, \text{ м}^3/\text{год}, \quad (5.1)$$

де $Q_{изб}$ – надлишкова кількість тепла, тобто різниця між його приходом і відходом, ккал/год;

c – теплообмінність повітря, кДж/кг °С;

ρ_n – щільність приточного повітря, кг/м³;

$t_{пр}$ і $t_{вид}$ – відповідно температура приточного повітря і того, що виділяється, °С.

У зв'язку з тим, що загально обмінна вентиляція повинна забезпечити допустиму температуру повітря на робочі місця, то

$$t_{вид} = t_{p.з.} + t (H - 2), \quad (5.2)$$

де $t_{p.з.}$ – припустима температура повітря в робочій зоні відповідно до ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»;

H – висота виробничого приміщення, м;

t – температурний градієнт по висоті приміщення, °С;

2 – висота робочої зони, м.

Звичайно приймають $t = 2-5$ °С.

$$t_{вид} = 21 + 2(3 - 2) = 23 \text{ °С}$$

Тепловиділення від електричного двигуна при нормальному завантаженні мають бути розраховані за формулою:

$$Q_1 = \frac{N (1 - \eta) \cdot 860}{\eta}, \text{ ккал/год} \quad (5.3)$$

де N – номінальна потужність електродвигуна,

$$Q_1 = \frac{0,7 (1 - 0,95) \cdot 860}{0,95} = 31,7 \text{ (ккал/год)}$$

Тепловиділення від джерел освітлення, ккал/год

$$Q_2 = q \cdot E_n \cdot S, \text{ ккал/год} \quad (5.4)$$

де Q_2 – тепло від джерел світла, ккал/год;

E_n – нормована освітленість, відповідно СНиП II-4-79.

«Строительные нормы и правила. Естественное и искусственное освещение.

Нормы проектирования».

$q = 0,05$ ккал – тепло, що виділяється на 1 м^2 освітлюваної поверхні.

$$Q_2 = 0,05 \cdot 400 \cdot 25 = 500 \text{ ккал/год}$$

Розрахуємо тепло, що поступає у приміщення від сонячної радіації за формулою:

$$Q_3 = F_{\text{ост}} \cdot q_{\text{ост}} \cdot A_{\text{ост}}, \text{ ккал/год} \quad (5.5)$$

де $F_{\text{ост}}$ – поверхня засклення, м^2 ;

$q_{\text{ост}}$ – величина радіації через 1 м^2 , ккал/ м^2 год;

$A_{\text{ост}}$ – коефіцієнт, що залежить від характеристики засклення.

$$Q_3 = 3 \cdot 18 \cdot 0,7 = 37,8 \text{ ккал/год}$$

Розрахуємо тепловиділення від працюючих за формулою:

$$Q_4 = q_1 \cdot n, \text{ ккал/год} \quad (5.6)$$

де q_1 – тепловиділення від одної людини в залежності від праці, що виконується, Вт;

n – кількість працюючих (3 людини).

Відповідно ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» приміщення з ПК відноситься до категорії робіт по важкості до категорії I-а (легка) $q = (90 - 120 \text{ ккал/год})$

$$Q_4 = 120 \cdot 3 = 360 \text{ ккал/год}$$

$$Q_{\text{ізб}} = 31,7 + 500 + 37,8 + 360 = 929,5 \text{ ккал/год}$$

Тоді L дорівнює:

$$L = \frac{929,5}{0,24 \cdot 1,2 \cdot (25 - 21)} = 806,8 \text{ (м}^3\text{/год)} \approx 1000 \text{ (м}^3\text{/год)}$$

Таким чином, обираємо кондиціонер ALASCA R-1000SE, який забезпечує регулювання температури повітря ($L = 1000 \text{ м}^3\text{/год}$), вологості та повітрообміну згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» для кімнати площею 20 м^2 .

3 Розрахунок освітлення у приміщенні.

Одним з основних питань охорони праці є організація раціонального освітлення виробничих приміщень робочих місць.

Для освітлення приміщення, у якому працюють фахівці, використовується змішане освітлення, тобто сполучення природного й штучного освітлення. Природне освітлення здійснюється через вікно. Штучне освітлення використовується при недостатньому природному освітленні й здійснюється за допомогою двох систем: загального й місцевого освітлення.

Розрахунок системи освітлення виробляється методом коефіцієнта використання світлового потоку, що виражається відношенням світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп.

Його величина залежить від характеристик світильника, розмірів приміщення, фарбування стін і стелі, яка характеризується коефіцієнтами відбиття стін і стелі.

Світловий потік при загальному освітленні визначається за формулою:

$$\Phi_{\text{заг}} = E_n \cdot S \cdot k \cdot z / \eta, \quad (5.7)$$

де E – нормована освітленість робочого місця по нормі, лк ($E=400$ лк);

S – площа приміщення, м^2 ;

k – коефіцієнт запасу, що враховує зношування й забруднення світильників ($k = 1,2$);

Z – коефіцієнт, що враховує нерівномірність висвітлення ($Z = 1,1$);

η – коефіцієнт використання світлового потоку, що вибирається з таблиць залежно від типу світильника, розмірів приміщення, коефіцієнта відбиття стін і стелі приміщення.

Оберемо з таблиці коефіцієнт використання світлового потоку за наступним даними:

- коефіцієнт відбиття для стелі приймемо $\rho_{\text{п}} = 60\%$;
- коефіцієнт відбиття стін $\rho_{\text{з}} = 30\%$;

$$i = \frac{A * B}{H_c (A + B)}, \quad (5.8)$$

де A і B – сторони приміщення, м;

H_c – розрахункова висота підвісу світильника, м.

Розрахункова висота визначається за формулою:

$$H_c = H - h_c - h_p, \quad (5.9)$$

де H – висота приміщення, м;

h_c – відстань світильників від перекриття ($h_c=0,1$ м);

h_p – висота робочої поверхні над підлогою ($h_p=0,8$ м).

Таким чином, розрахункова висота дорівнює:

$$H_c=3,5-0,1-0,8=2,6 \text{ (м)}.$$

Підставляючи значення у формулу (5.7), одержимо:

$$i = \frac{5 \cdot 5}{2,6 \cdot (5 + 5)} = 0,962.$$

Тоді визначаємо значення коефіцієнта використання світлового потоку $\eta = 0,45$.

Загальний світловий потік дорівнює:

$$\Phi_{zag} = \frac{400 \cdot 25 \cdot 1.3 \cdot 1.1}{0.45} = 31777 \text{ лм}$$

Найбільш прийнятними для приміщення є люмінесцентні лампи ЛБ (білого світла), потужністю 80 Вт які мають потік $\Phi_1=4320$ лм. Кількість ламп дорівнює:

$$N = \frac{\Phi_{zag}}{\Phi_1} \text{ (шт.);} \quad (5.10)$$

Підставляючи значення, отримуємо:

$$N = \frac{31777}{4320} = 7,35 = 8 \text{ (шт.)}$$

Якщо у світильники встановити по 2 лампи 80 Вт, з потоком 4320 лм, то потрібно розташувати 4 світильники на однаковій відстані до центру кімнати (близько 2 м).

ДОДАТОК В1 - БО-вирішувачі для гілок Q1...Q8 і процеси Q_J(t)

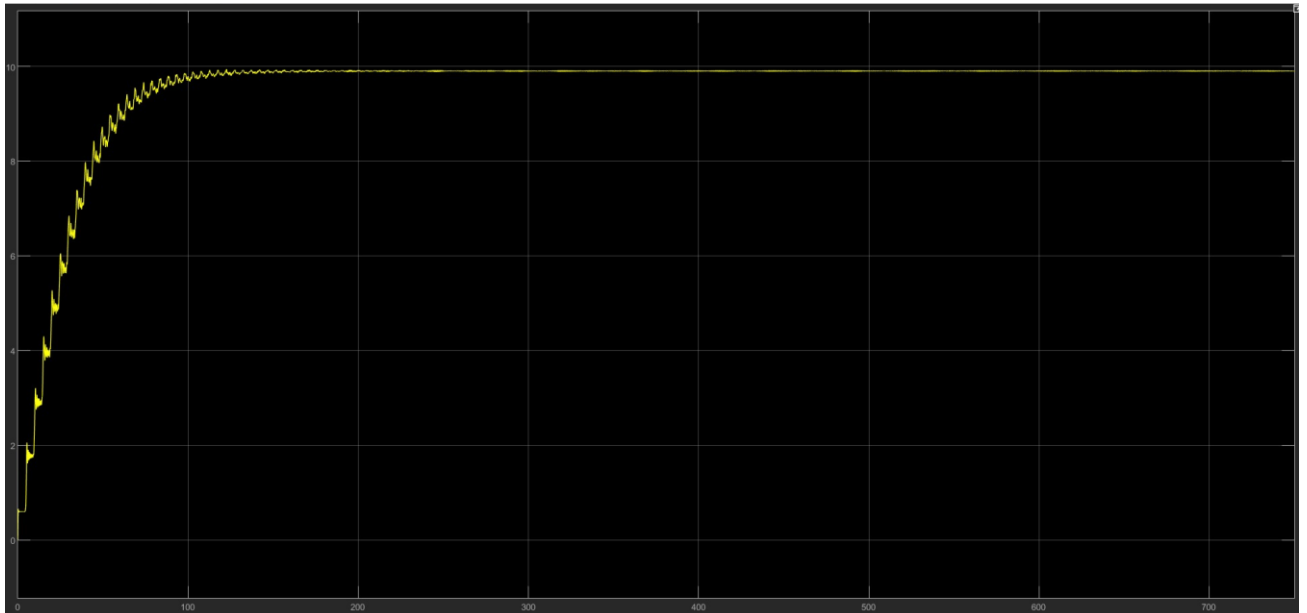
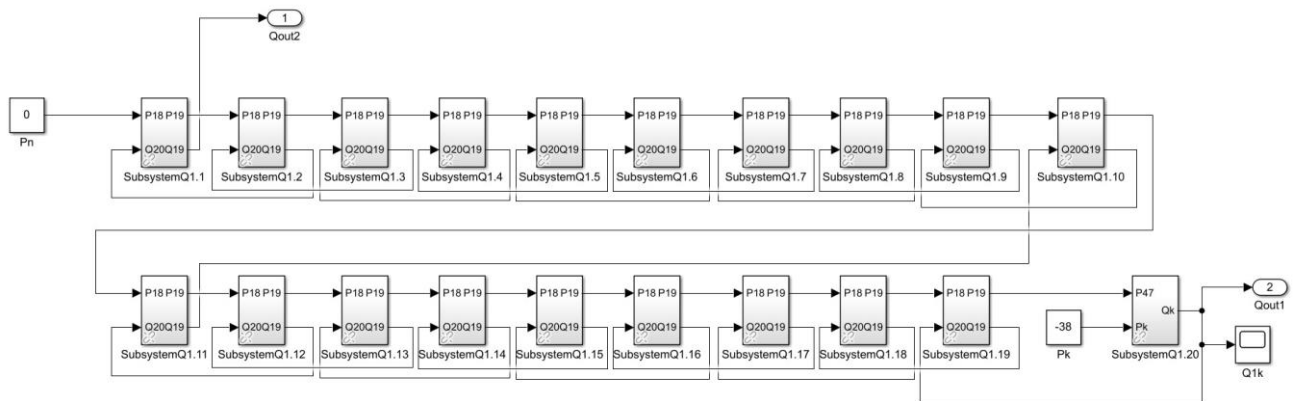


Рисунок В1.1 – БО-вирішувач для гілки Q1 і процес Q₁(t)

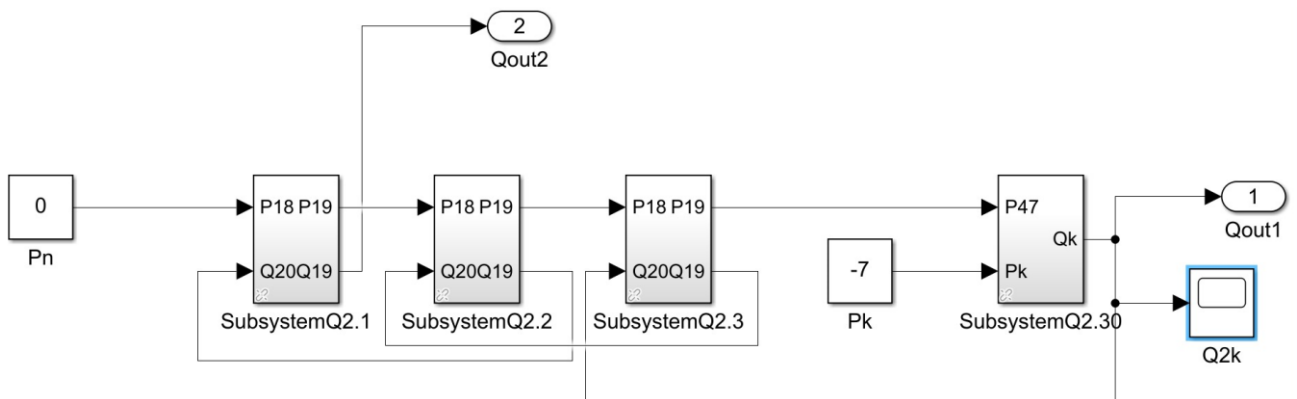
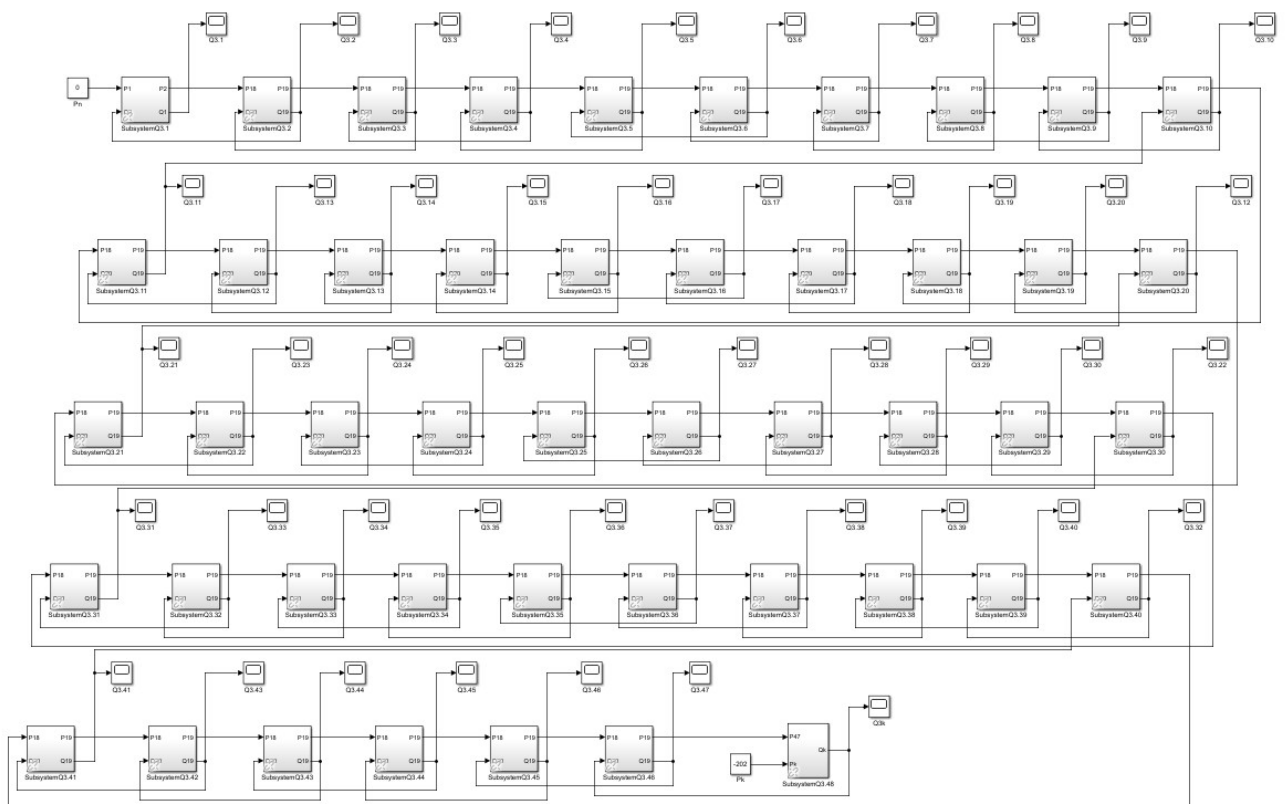




Рисунок В1.2 – БО-вирішувач для гілки Q2 і процес Q2(t)



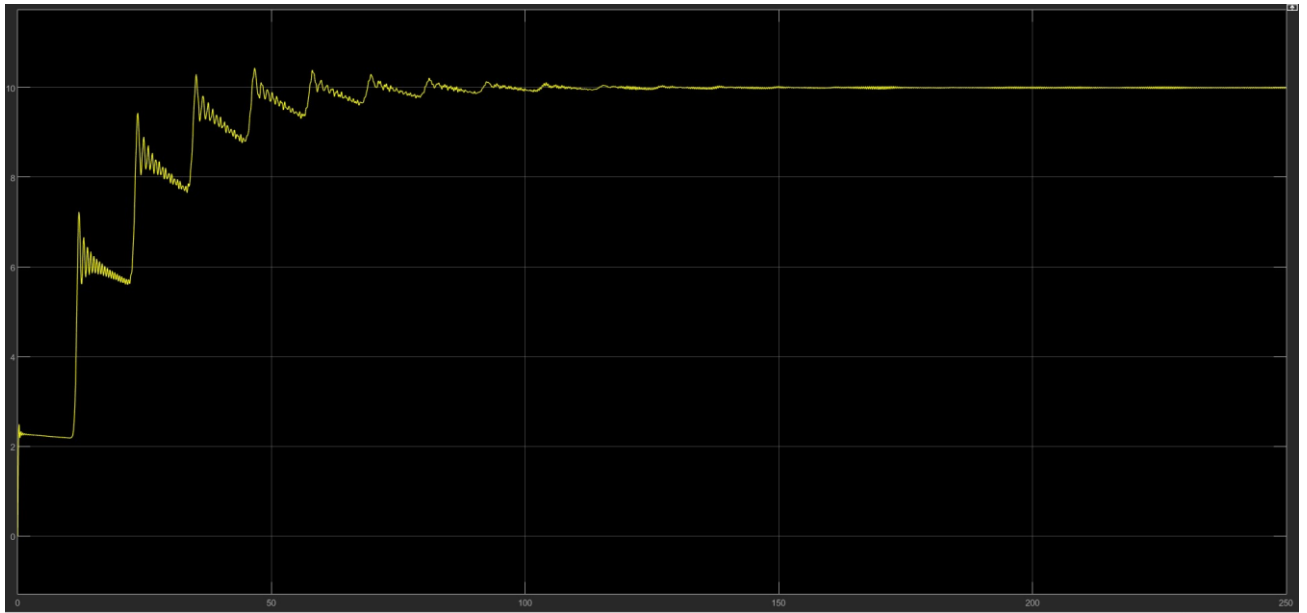


Рисунок В1.3 – БО-вирішувач для гілки Q3 і процес Q3(t)



Рисунок В1.4 – БО-вирішувач для гілки Q4 і процес Q4(t)

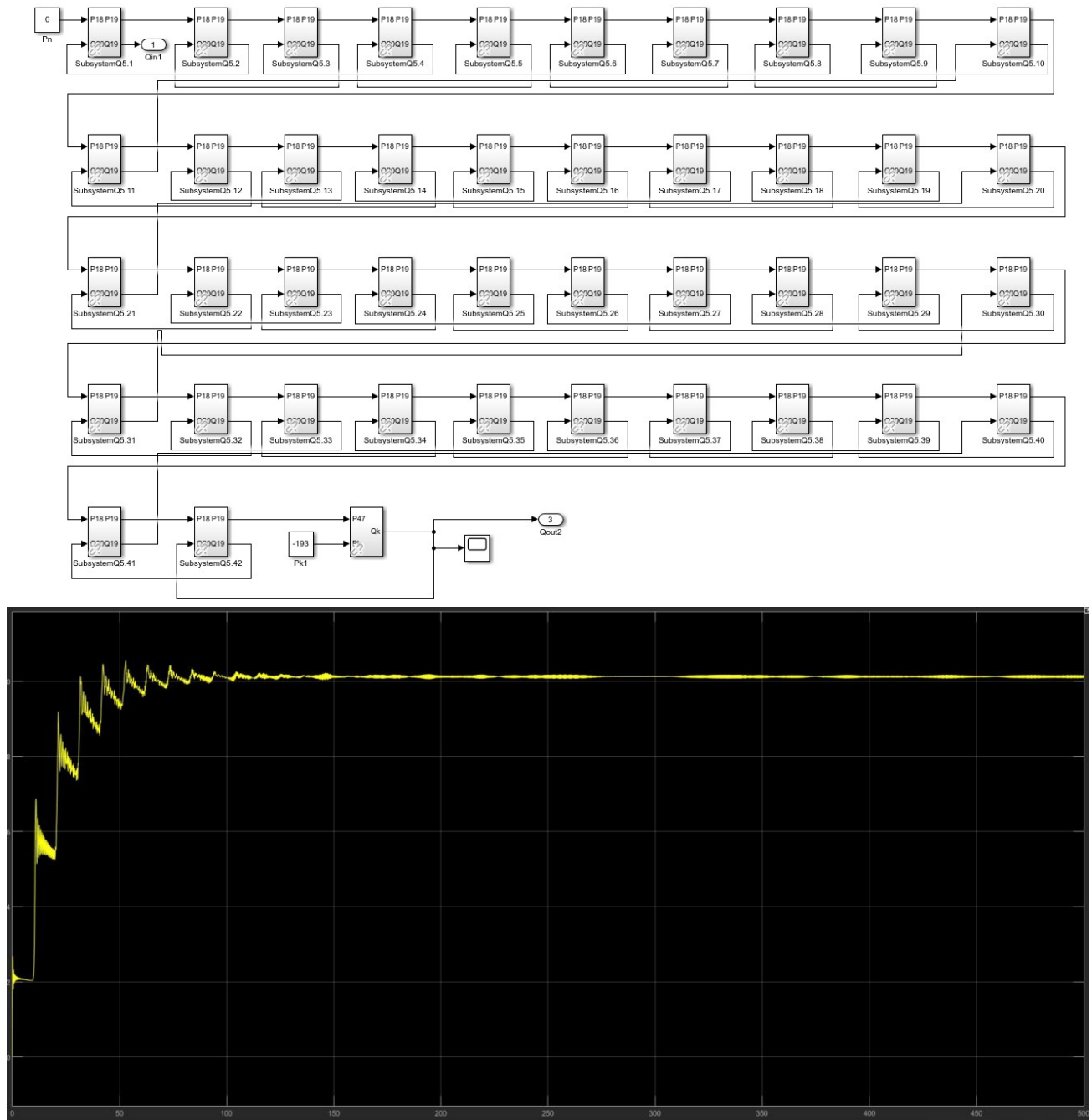


Рисунок В1.5 – БО-вирішувач для гілки Q5 і процес Q5(t)

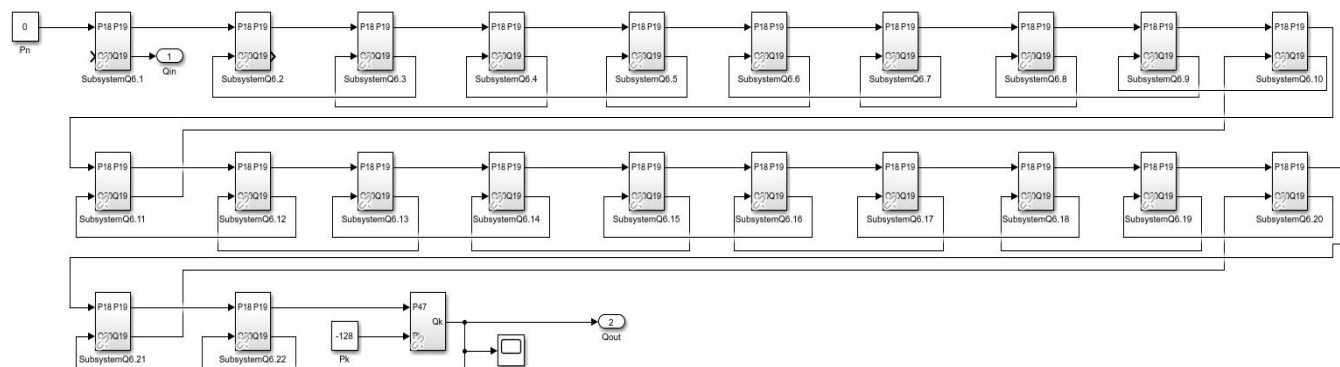




Рисунок В1.6 – БО-вирішувач для гілки Q6 і процес Q6(t)

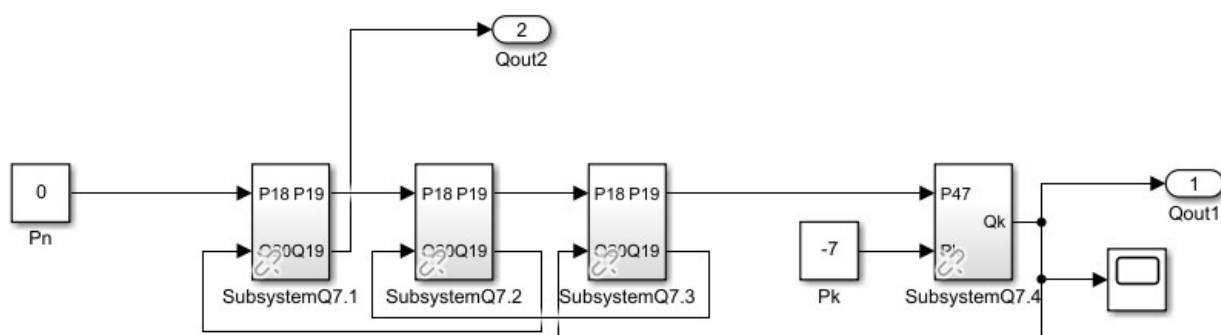


Рисунок В1.7 – БО-вирішувач для гілки Q7 і процес Q7(t)

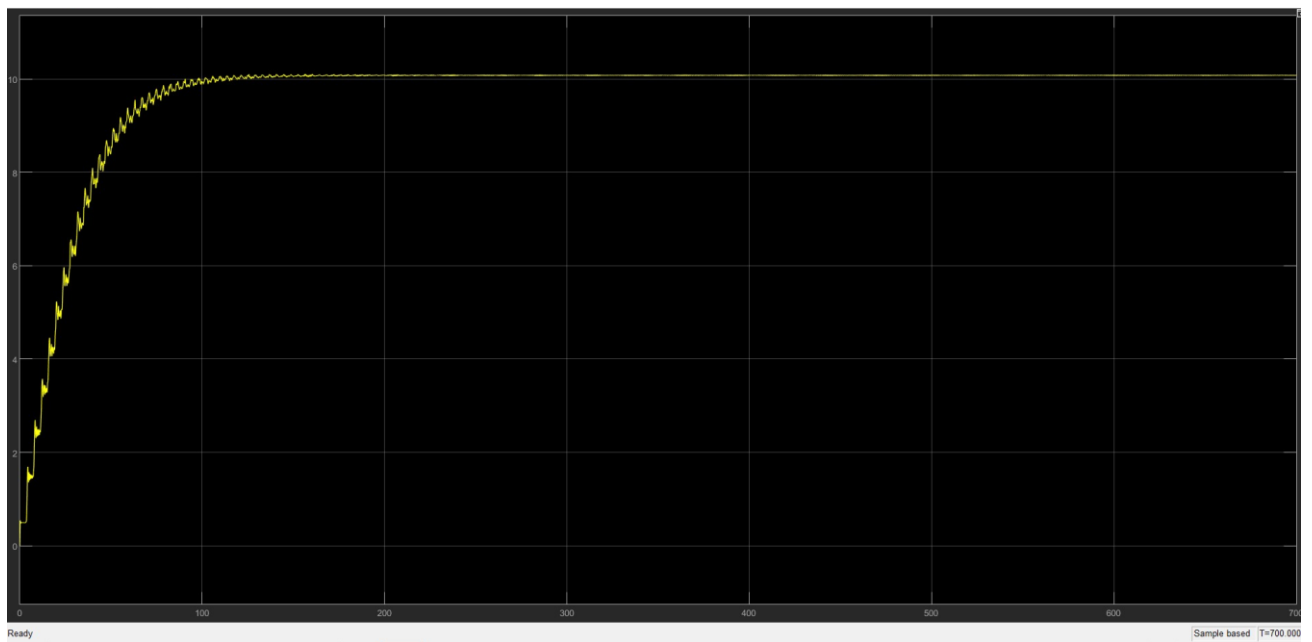
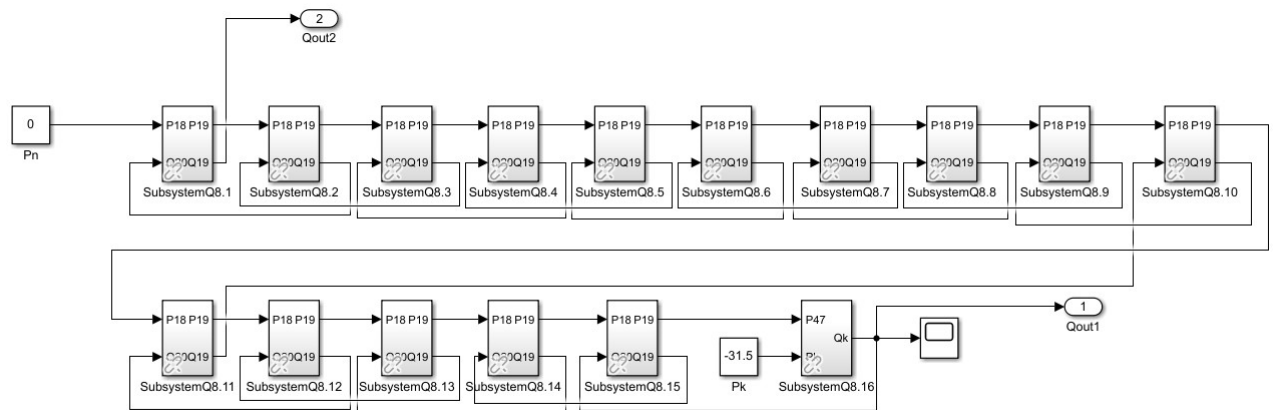


Рисунок В1.8 – БО-вирішувач для гілки Q8 і процес Q8(t)

ДОДАТОК В2 - С-програма вирішувача для j-гілок).

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace WindowsFormsApp2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent(); //Завантажує скомпільовану сторінку компонента
            chart1.Series[0].Points.Clear();
            chart1.Series[1].Points.Clear();
            chart1.Series[2].Points.Clear();
            chart1.Series[3].Points.Clear();
        }

        private void button1_Click_1(object sender, EventArgs e)
        { // Задання початкових параметрів
            List<double> list = new List<double>();
            double[,] Q = new double[48, 2500];
            double[,] P = new double[48, 2500];

            int i, kmax = 48;
            double alpha = 0.093;
            double beta = 0.004;
            double gamma = 728.6;
            double h = 0.05;
            double sum_of_h = 0;

            for (i = 0; i < 2500; i++) //Прохід по маментам часу(i)
            {
                for (int k = 0; k < kmax; k++) // Прохід по частинам гілки(k+1)
                {
                    if (k == kmax - 1)
                    {
                        Q[k, i] = 0;
                        P[k, i] = -202.0; // При Q = 10, P = -202
                    }
                    else
                    {
                        Q[k, i] = 0;
                        P[k, i] = 0;
                    }
                }
            }
            for (i = 2; i < 2498; i++)
            {
                list.Add(sum_of_h = sum_of_h + h);

                for (int k = 1; k < kmax; k++)
                {
                    if (k == kmax - 1) // Якщо кінцева частина гілки, тоді виконуємо:
                    {
                        Q[k, i + 1] = Q[k, i] + h * (alpha * (P[k - 1, i] - P[k, i]) - beta * Q[k, i] * Math.Abs(Q[k, i]));
                    }
                    else
                    {
                        // Якщо не кінцева частина гілки, тоді виконуємо:

```

```

        Q[k, i + 1] = Q[k, i] + h * (alpha * (P[k - 1, i] - P[k, i]) - beta * Q[k, i] * Math.Abs(Q[k, i]));
        P[k, i + 1] = P[k, i - 2] + h * (gamma * (Q[k, i] - Q[k + 1, i]));
    }

}

// Виведення графіків
int point = 0;
while (point < 2495)
{
    chart1.Series[0].Points.AddXY(list[point], Q[1, point]);
    chart1.Series[1].Points.AddXY(list[point], Q[15, point]);
    chart1.Series[2].Points.AddXY(list[point], Q[30, point]);
    chart1.Series[3].Points.AddXY(list[point], Q[45, point]);
    point++;
}

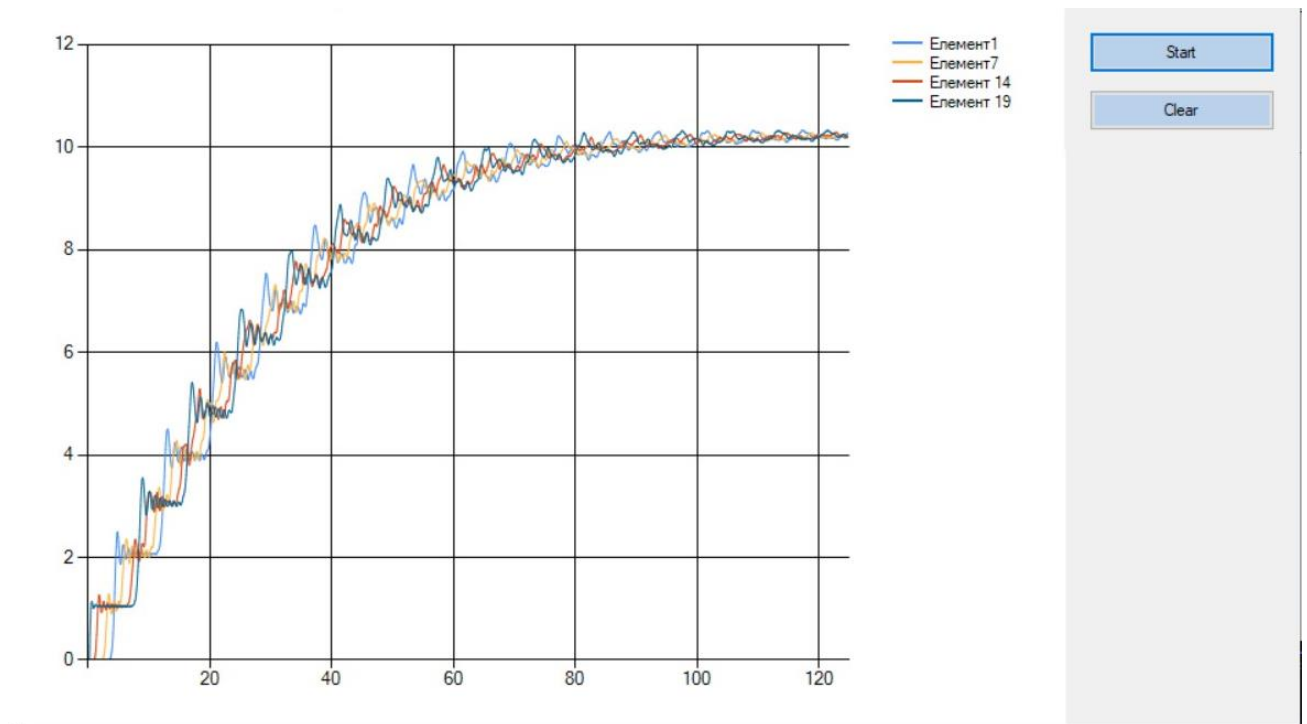
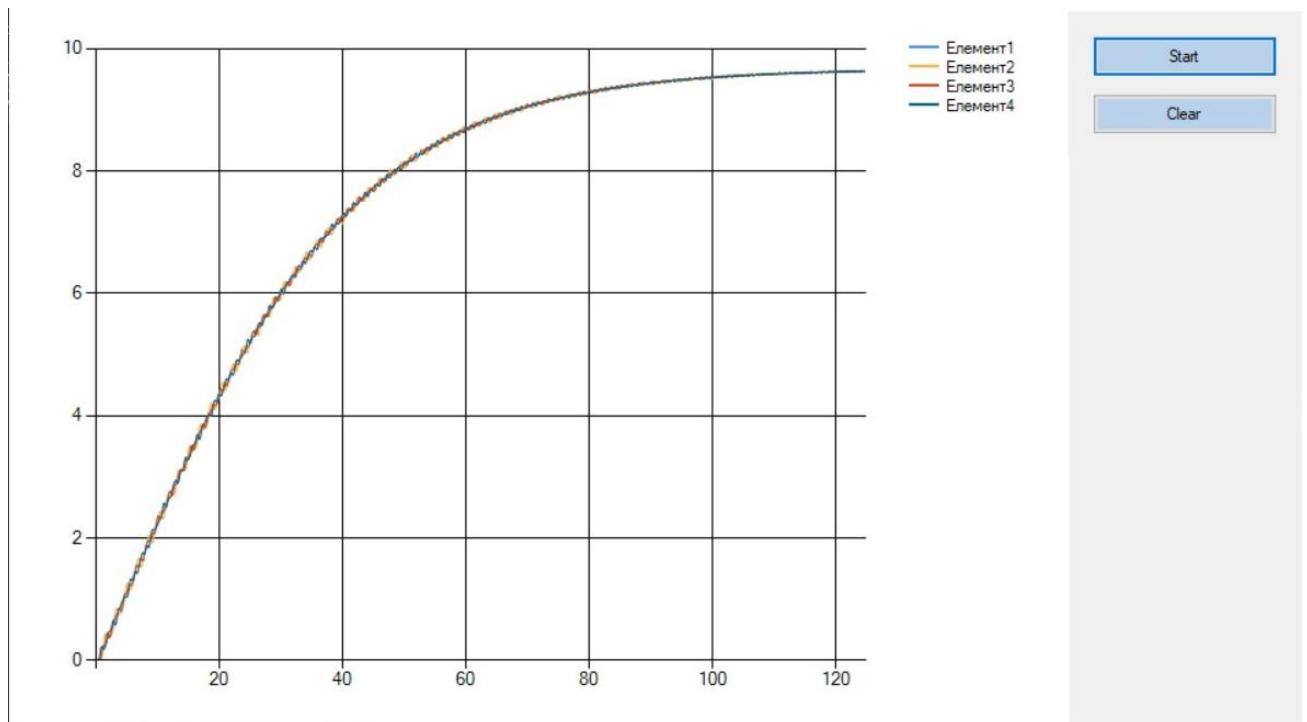
// Очищення графіків
private void button2_Click(object sender, EventArgs e)
{
    chart1.Series[0].Points.Clear();
    chart1.Series[1].Points.Clear();
    chart1.Series[2].Points.Clear();
    chart1.Series[3].Points.Clear();
}

private void chart1_Click(object sender, EventArgs e)
{
}

private void Form1_Load(object sender, EventArgs e)
{
}
}
}

```

ДОДАТОК В3 - Динамічні процеси j-гілок, отримані на С-вирішувачі

Рисунок В3.1 – Процеси $Q1(v, t)$, отримані на С-вирішувачіРисунок В3.2 – Процеси $Q2(v, t)$, отримані на С-вирішувачі

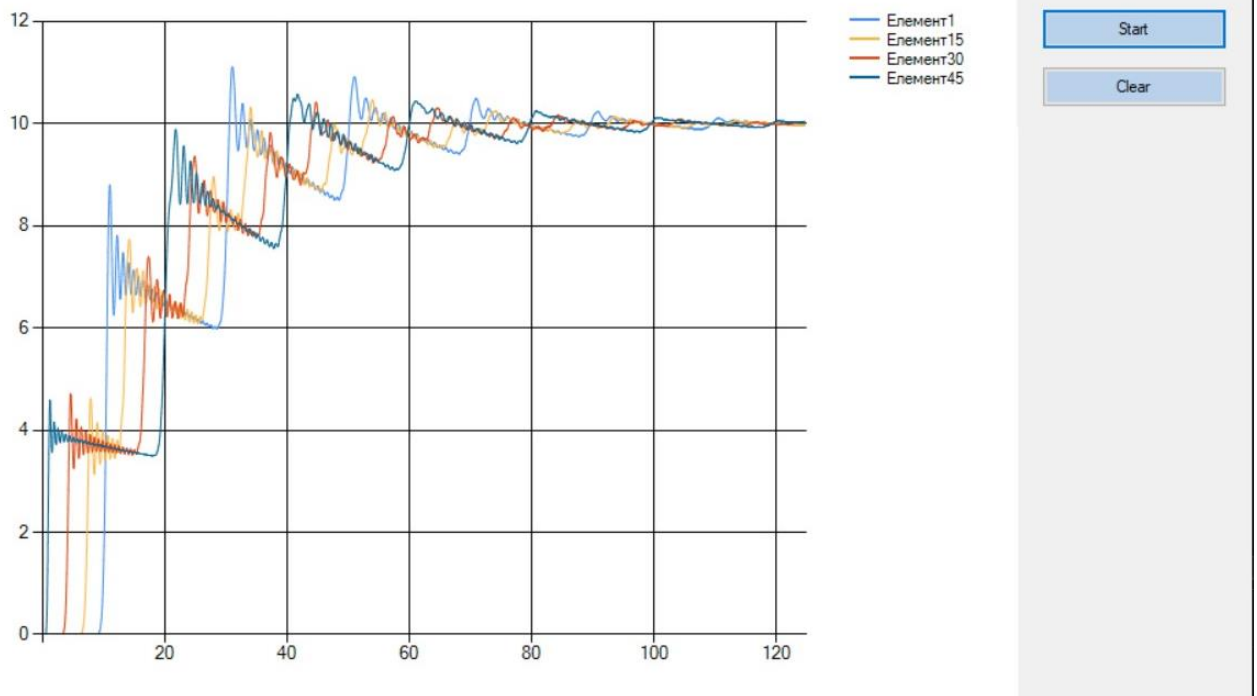


Рисунок В3.3 – Процеси $Q3(v, t)$, отримані на С-вирішувачі

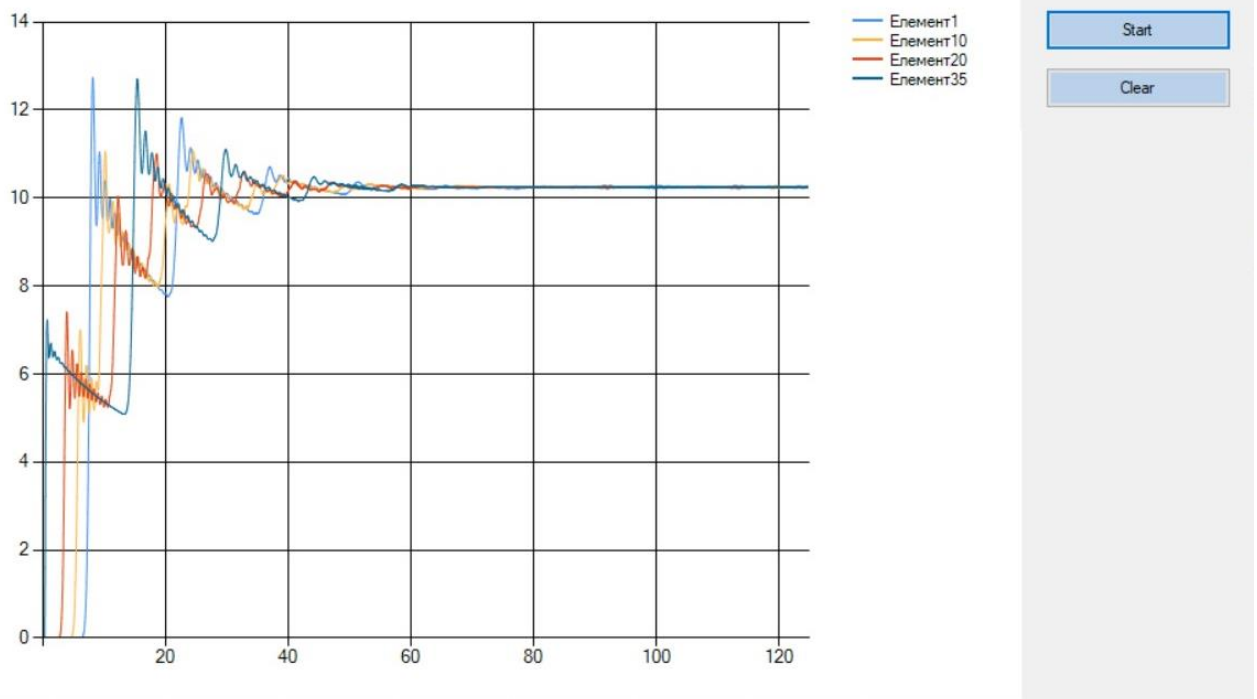


Рисунок В3.4 – Процеси $Q4(v, t)$, отримані на С-вирішувачі

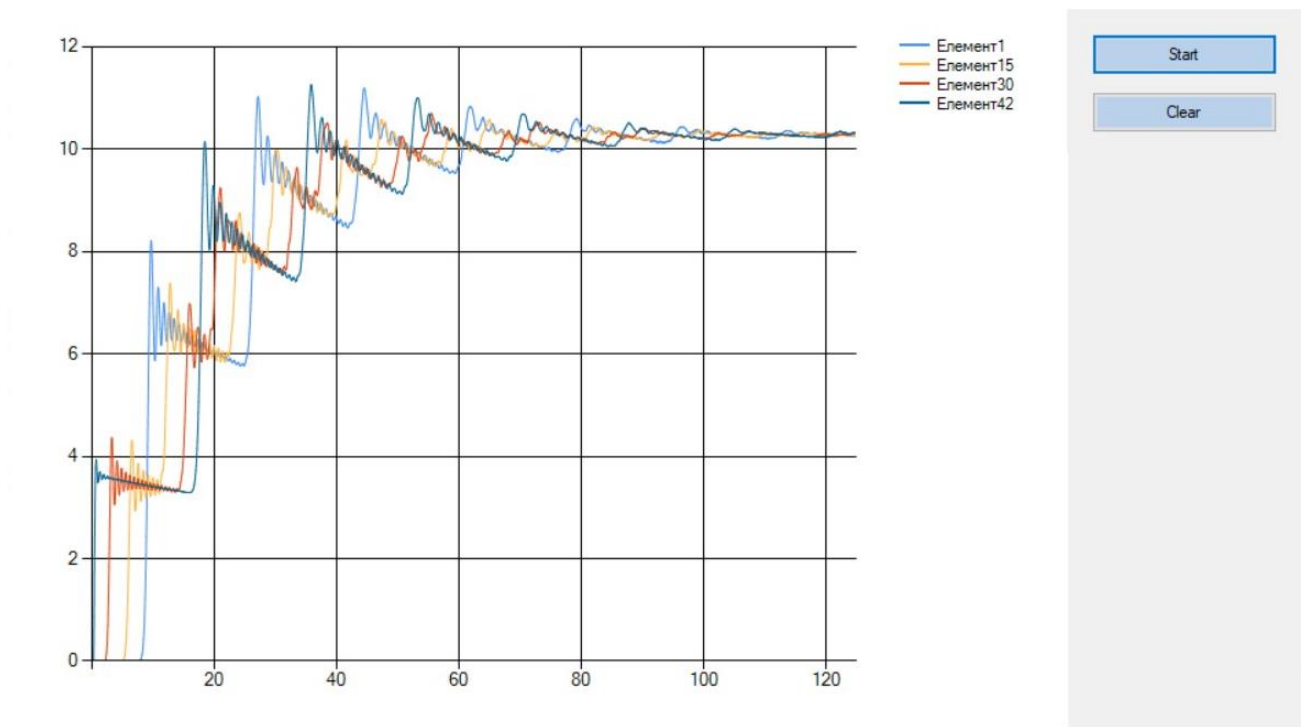


Рисунок В3.5 – Процеси $Q5(v, t)$, отримані на С-вирішувачі

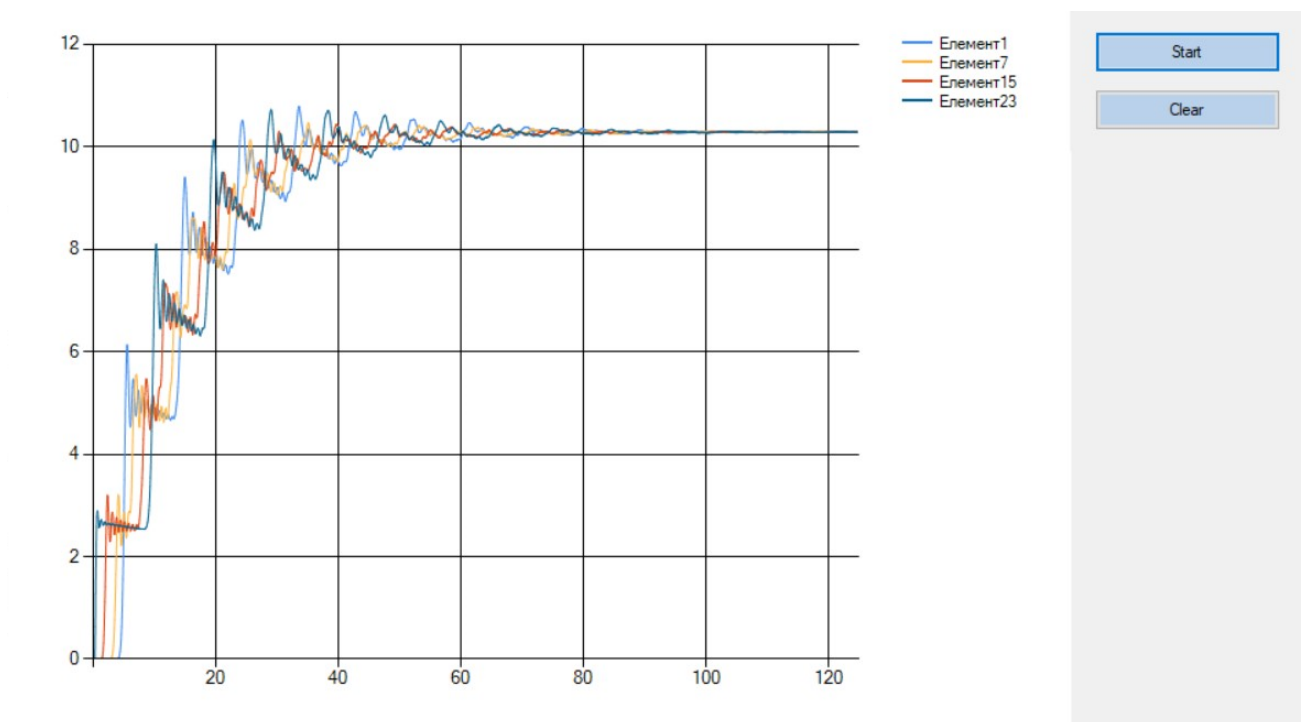


Рисунок В3.6 – Процеси $Q6(v, t)$, отримані на С-вирішувачі

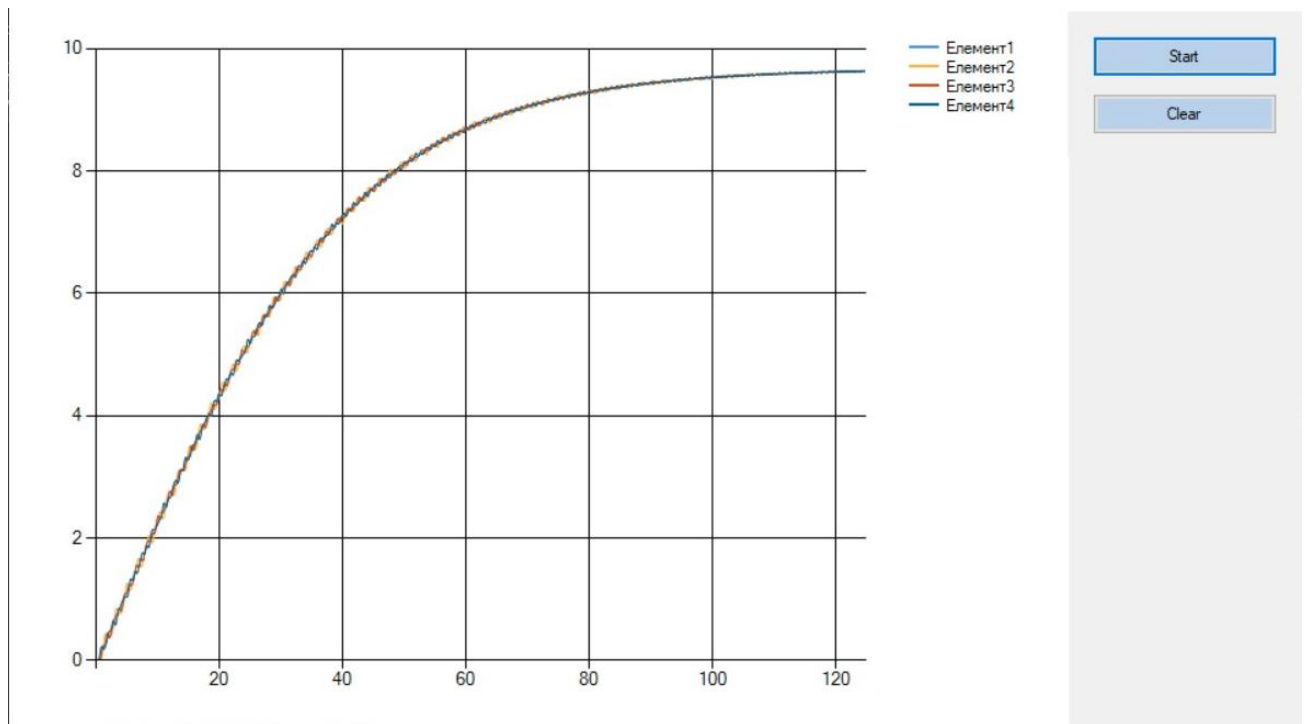


Рисунок В3.7 – Процеси $Q7(v, t)$, отримані на С-вирішувачі

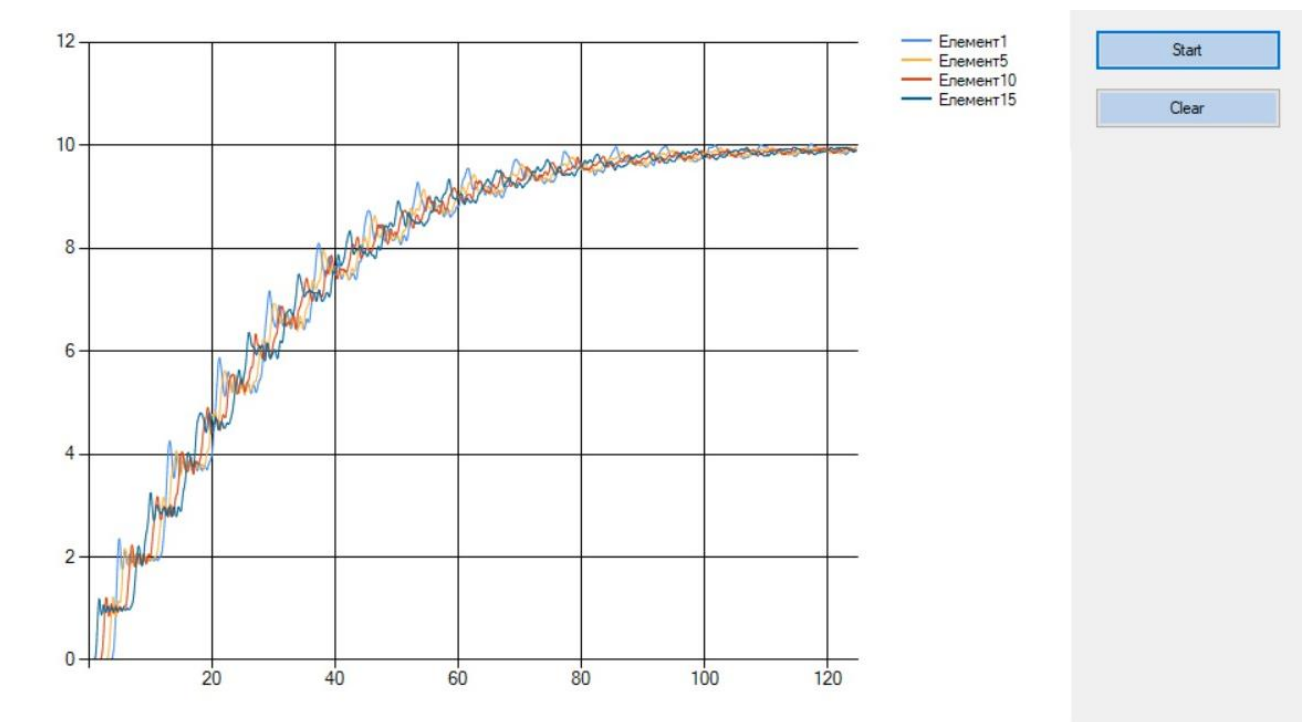


Рисунок В3.8 – Процеси $Q8(v, t)$, отримані на С-вирішувачі

ДОДАТОК В4 - Текст С-програми вирішувача рівнянь МДОРП

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace WindowsFormsApp2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent(); //Завантажує скомпільовану сторінку компонента
        }
        private void button1_Click_1(object sender, EventArgs e)
        {
            // Задання початкових параметрів
            List<double> t1 = new List<double>();
            double[,] Q1 = new double[22, 18200];
            double[,] P1 = new double[22, 18200];

            List<double> t2 = new List<double>();
            double[,] Q2 = new double[6, 18200];
            double[,] P2 = new double[6, 18200];

            List<double> t3 = new List<double>();
            double[,] Q3 = new double[50, 18200];
            double[,] P3 = new double[50, 18200];

            List<double> t4 = new List<double>();
            double[,] Q4 = new double[37, 18200];
            double[,] P4 = new double[37, 18200];

            List<double> t5 = new List<double>();
            double[,] Q5 = new double[45, 18200];
            double[,] P5 = new double[45, 18200];

            List<double> t6 = new List<double>();
            double[,] Q6 = new double[25, 18200];
            double[,] P6 = new double[25, 18200];

            List<double> t7 = new List<double>();
            double[,] Q7 = new double[6, 18200];
            double[,] P7 = new double[6, 18200];

            List<double> t8 = new List<double>();
            double[,] Q8 = new double[18, 18200];
            double[,] P8 = new double[18, 18200];

            double alpha1 = 0.13;
            double betta1 = 0.00252;
            double gamma1 = 523.6;
            double h1 = 0.01;
            double sum_of_h1 = 0;

            double alpha2 = 0.13;
            double betta2 = 0.00252;
            double gamma2 = 523.6;

            double alpha3 = 0.093;

```

```

double betta3 = 0.004;
double gamma3 = 728.6;

double alpha4 = 0.29;
double betta4 = 0.009;
double gamma4 = 569.1;

double alpha5 = 0.8;
double betta5 = 0.004;
double gamma5 = 756.2;

double alpha6 = 0.92;
double betta6 = 0.00252;
double gamma6 = 756.2;

double alpha7 = 0.13;
double betta7 = 0.00252;
double gamma7 = 523.6;

double alpha8 = 0.13;
double betta8 = 0.00252;
double gamma8 = 523.6;

double P_uzel1 = 0, P_uzel2 = 0, P_uzel3 = 0, P_uzel4 = 0;
double gamma_uzel0 = - 1.1;
double gamma_uzel1 = -15.4;
double gamma_uzel2 = -2.8;
double gamma_uzel3 = -25;

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 22; k++)
    {
        if (k == 21)
        {
            P1[k, i] = -4100;
            Q1[k, i] = 0;
        }
        else
        {
            Q1[k, i] = 0;
            P1[k, i] = 0;
        }
    }
}

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 6; k++)
    {
        Q2[k, i] = 0;
        P2[k, i] = 0;
    }
}
} //Q2

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 50; k++)
    {

```

```

        Q3[k, i] = 0;
        P3[k, i] = 0;
    }
} //Q3

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 37; k++)
    {
        Q4[k, i] = 0;
        P4[k, i] = 0;
    }
} //Q4

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 45; k++)
    {
        Q5[k, i] = 0;
        P5[k, i] = 0;
    }
} //Q5

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 25; k++)
    {
        Q6[k, i] = 0;
        P6[k, i] = 0;
    }
} //Q6

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 6; k++)
    {
        Q7[k, i] = 0;
        P7[k, i] = 0;
    }
} //Q7

for (int i = 0; i < 18200; i++)
{
    for (int k = 0; k < 18; k++)
    {
        Q8[k, i] = 0;
        P8[k, i] = 0;
    }
} //Q8

for (int i = 2; i < 18197; i++)
{
    t1.Add(sum_of_h1 = sum_of_h1 + h1); //заполнение оси времени для графика

    for (int k = 1; k < 22; k++)
    {

        Q1[k, i + 1] = Q1[k, i] + h1 * (alpha1 * (P1[k - 1, i] - P1[k, i]) - betta1 * Q1[k, i] * Math.Abs(Q1[k, i]));
        if (k < 21)
        {
            P1[k, i + 1] = P1[k, i - 2] + h1 * gamma1 * (Q1[k, i] - Q1[k + 1, i]);
        }
    }
}

```

```

    }
}

P_uzel1 = gamma_uzel0 * (Q1[21, i + 1] - Q2[1, i + 1] - Q5[1, i + 1] - Q6[1, i + 1]);
P2[5, i] = P_uzel1;

for (int k = 1; k < 6; k++)
{
    Q2[k, i + 1] = Q2[k, i] + h1 * (alpha2 * (P2[k - 1, i] - P2[k, i]) - beta2 * Q2[k, i] * Math.Abs(Q2[k, i]));
    if (k < 5)
    {
        P2[k, i + 1] = P2[k, i - 2] + h1 * gamma2 * (Q2[k, i] - Q2[k + 1, i]);
    }
}

P_uzel2 = gamma_uzel1 * (Q2[5, i + 1] - Q3[1, i + 1] - Q4[1, i + 1]);
P3[49, i] = P_uzel2;

for (int k = 1; k < 50; k++)
{
    Q3[k, i + 1] = Q3[k, i] + h1 * (alpha3 * (P3[k - 1, i] - P3[k, i]) - beta3 * Q3[k, i] * Math.Abs(Q3[k, i]));
    if (k < 49)
    {
        P3[k, i + 1] = P3[k, i - 2] + h1 * gamma3 * (Q3[k, i] - Q3[k + 1, i]);
    }
}

P_uzel2 = gamma_uzel1 * (Q2[5, i + 1] - Q3[1, i + 1] - Q4[1, i + 1]);
P5[44, i] = P_uzel1;

for (int k = 1; k < 45; k++)
{
    Q5[k, i + 1] = Q5[k, i] + h1 * (alpha5 * (P5[k - 1, i] - P5[k, i]) - beta5 * Q5[k, i] * Math.Abs(Q5[k, i]));
    if (k < 44)
    {
        P5[k, i + 1] = P5[k, i - 2] + h1 * gamma5 * (Q5[k, i] - Q5[k + 1, i]);
    }
}

P_uzel1 = gamma_uzel0 * (Q1[21, i + 1] - Q2[1, i + 1] - Q5[1, i + 1] - Q6[1, i + 1]);
P6[24, i] = P_uzel1;

for (int k = 1; k < 25; k++)
{
    Q6[k, i + 1] = Q6[k, i] + h1 * (alpha6 * (P6[k - 1, i] - P6[k, i]) - beta6 * Q6[k, i] * Math.Abs(Q6[k, i]));
    if (k < 24)
    {
        P6[k, i + 1] = P6[k, i - 2] + h1 * gamma6 * (Q6[k, i] - Q6[k + 1, i]);
    }
}

P_uzel2 = gamma_uzel1 * (Q2[5, i + 1] - Q3[1, i + 1] - Q4[1, i + 1]);
P4[36, i] = P_uzel2;

for (int k = 1; k < 37; k++)
{

```

```

    Q4[k, i + 1] = Q4[k, i] + h1 * (alpha4 * (P4[k - 1, i] - P4[k, i]) - betta4 * Q4[k, i] * Math.Abs(Q4[k, i]));
    if (k < 36)
    {
        P4[k, i + 1] = P4[k, i - 2] + h1 * gamma4 * (Q4[k, i] - Q4[k + 1, i]);
    }
}

P_uzel3 = gamma_uzel2 * (-Q7[1, i + 1] + Q3[36, i + 1] + Q4[1, i + 1]);
P7[5, i] = P_uzel3;

for (int k = 1; k < 6; k++)
{

    Q7[k, i + 1] = Q7[k, i] + h1 * (alpha7 * (P7[k - 1, i] - P7[k, i]) - betta7 * Q7[k, i] * Math.Abs(Q7[k, i]));
    if (k < 5)
    {
        P7[k, i + 1] = P7[k, i - 2] + h1 * gamma7 * (Q7[k, i] - Q7[k + 1, i]);
    }
}

P_uzel4 = gamma_uzel3 * (-Q8[1, i + 1] + Q7[5, i + 1] + Q5[44, i + 1] + Q6[24, i + 1]);
P8[17, i] = P_uzel4;

for (int k = 1; k < 18; k++)
{

    Q8[k, i + 1] = Q8[k, i] + h1 * (alpha8 * (P8[k - 1, i] - P8[k, i]) - betta8 * Q8[k, i] * Math.Abs(Q8[k, i]));
    if (k < 17)
    {
        P8[k, i + 1] = P8[k, i - 2] + h1 * gamma8 * (Q8[k, i] - Q8[k + 1, i]);
    }
}
}
int lp = 0;
while (t1.Count != lp)
{
    if (lp % 10 == 0)
    {
        chart1.Series[0].Points.AddXY(t1[lp], Q1[1, lp]);

        chart2.Series[0].Points.AddXY(t1[lp], Q2[1, lp]);

        chart3.Series[0].Points.AddXY(t1[lp], Q3[1, lp]);

        chart4.Series[0].Points.AddXY(t1[lp], Q4[1, lp]);

        chart5.Series[0].Points.AddXY(t1[lp], Q5[1, lp]);

        chart6.Series[0].Points.AddXY(t1[lp], Q6[1, lp]);

        chart7.Series[0].Points.AddXY(t1[lp], Q7[1, lp]);

        chart8.Series[0].Points.AddXY(t1[lp], Q8[1, lp]);
    }
}

```

```

        }
        lp++;
    }
}

// Очищення графіків
private void button2_Click(object sender, EventArgs e)
{
    chart1.Series[0].Points.Clear();
    chart2.Series[0].Points.Clear();
    chart3.Series[0].Points.Clear();
    chart4.Series[0].Points.Clear();
    chart5.Series[0].Points.Clear();
    chart6.Series[0].Points.Clear();
    chart7.Series[0].Points.Clear();
    chart8.Series[0].Points.Clear();

}

private void chart1_Click(object sender, EventArgs e)
{
}

private void Form1_Load(object sender, EventArgs e)
{
}
}
}

```

