

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ
НАВЧАННЯ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

на тему:

**«Комп'ютерне забезпечення для моделювання
роботи двопозиційного регулятора температури в
теплообмінному апараті»**

Завідувач

випускаючої кафедри

Керівник роботи

Студента групи ІНЗ – 72с

Довбиш А.С.

Авраменко В.В.

Сіробаба О.М

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Центр заочної, дистанційної і вечірньої форм навчання

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 г.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІНз-72С спеціальності “Комп'ютерні науки” заочної форми навчання Сіробаби Олександра Миколайовича.

Тема: “ Комп'ютерне забезпечення для моделювання роботи двопозиційного регулятора температури в теплообмінному апараті ”

Затверджена наказом по СумДУ

№ _____ от _____ 2021 г.

Зміст пояснювальної записки: 1) аналітична частина; 2) проектна частина; 3) програмна частина; 4) висновки..

Дата видачі завдання “ _____ ” _____ 2021 г.

Керівник випускної роботи _____ Авраменко В.В

Завдання прийняв до виконання _____ Сіробаба О.

Зміст

ВСТУП	3
Розділ 1. АНАЛІТИЧНА ЧАСТИНА	4
1.1 Аналіз понять, пов'язаних з математичним моделюванням	4
1.2 Класифікація моделей, способи моделювання	7
1.3 Цілі і завдання комп'ютерного моделювання	12
1.4 Комп'ютерне математичне моделювання: поняття, цілі та етапи	18
1.5 Різні класифікації математичних моделей	27
1.6 Тестування	30
1.7 C++, як інструмент для комп'ютерного моделювання	31
1.8 Висновки	33
Розділ 2. ПРОЕКТНА ЧАСТИНА	33
2.1 Постановка задачі	33
2.2 Математична постановка задачі	33
2.3 Вибір методу розв'язання задачі	35
2.4 Хід розв'язання задачі	35
2.5 Алгоритм розв'язання задачі	35
2.6 Алгоритми роботи програми	37
2.7 Налаштування властивостей проекту	39
2.8 Інструкція для користувача	41
2.9 Контрольні приклади	41
2.10 Висновки	48
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ	52
Додаток А	52

РЕФЕРАТ

Записка: 55 стор., 17 рис., 1 табл., 5 блок-схем., 1 додаток, 11 джерел.

Об'єкт дослідження — двопозиційний регулятор температури.

Мета роботи — Розробити алгоритм і комп'ютерну програму для моделювання роботи двопозиційного регулятора температури в теплообмінному апараті. Процес дослідити для різних співвідношень між сумарними теплоємностями середовищ в кожусі, через який підводиться тепло, і в кожусі, де відбувається реакція. В залежності від конструктивних особливостей теплообмінника мають місце різні величини відтоку тепла. Дослідити, як впливають ці особливості на процес регулювання температури в середовищі, де знаходяться реагуючі речовини.

Методи дослідження — метод Рунге-Кутта четвертого порядку.

Результати — У бакалаврській дипломній роботі було розроблено програму для моделювання роботи двопозиційного регулятора температури в теплообмінному апараті. Розроблене програмне забезпечення має практичне значення в моделюванні роботи двопозиційного регулятора температури в теплообмінному апараті.

ВСТУП

В багатьох галузях промисловості(зокрема в хімічній та хімікофармацевтичній) знаходять широке застосування теплообмінні апарати, в яких реагуючі речовини нагріваються через зовнішній кожух, який охоплює посудину, в якій відбувається реакція. В ній потрібно підтримувати постійне значення температури. Здебільшого для цього застосовуються двопозиційні регулятори температури, які включають повну подачу теплоносія в кожух, коли температура нижче заданої, і перекриває подачу при перевищенні. При цьому допускається деяке відхилення від заданого значення. Ця робота присвячена створенню програмного забезпечення для комп'ютерного моделювання процесів регулювання температури для різних співвідношеннях сумарних теплоємностей кожуха і посудини із реагуючими речовинами, а також для різних конструктивних особливостей теплообмінників.

Розділ 1. АНАЛІТИЧНА ЧАСТИНА

1.1 Аналіз понять, пов'язаних з математичним моделюванням

Моделювання являє собою один з основних методів пізнання, є формою відображення дійсності і полягає в з'ясуванні або відтворенні тих або інших властивостей реальних об'єктів, предметів і явищ за допомогою інших об'єктів, процесів, явищ, чи за допомогою абстрактного опису у вигляді зображення, плану, карти, сукупності рівнянь, алгоритмів і програм [1].

Можливості моделювання, тобто перенесення результатів, отриманих в процесі побудови і дослідження моделі, на оригінал засновані на тому, що модель в деякому сенсі відображає (відтворює, моделює, описує, імітує) деякі цікаві для дослідника риси об'єкта. Моделювання як форма відображення дійсності широко поширене, і достатньо повну класифікацію можливих видів моделювання вкрай скрутна, хоча б в силу багатозначності поняття «модель», широко використовується не тільки в науці і техніці, але і в творчості (мистецтві), і в повсякденному житті [2].

Слово «модель» походить від латинського слова «modulus», означає «міра», «зразок». Його первинне значення було пов'язане з будівельним мистецтвом, і майже у всіх європейських мовах воно використовувалося для позначення способу або прообразу, або речі, подібною в якомусь відношенні з іншою річчю [2].

Моделювання в наукових дослідженнях стало використовуватися ще в глибоку давнину і поступово захоплювало все нові області наукових знань: технічне конструювання, будівництво і архітектуру, астрономію, фізику, хімію, біологію і, нарешті, суспільні науки. Великих успіхів і визнання практично у всіх галузях сучасної науки приніс методу моделювання ХХ століття. Проте методологія моделювання довгий час розвивалася окремими науками незалежно один від одного. Була відсутня єдина система понять, єдина термінологія. Лише з часом стала усвідомлюватись роль моделювання як універсального методу наукового пізнання [2]. Термін «модель» широко

використовується в різних сферах людської діяльності і має безліч значень. У цьому розділі ми будемо розглядати тільки такі моделі, які є інструментами отримання знань. Модель - це такий матеріальний чи подумки представлений об'єкт, який у процесі дослідження заміщає об'єкт-оригінал так, що його безпосереднє вивчення дає нові знання про об'єкт-оригіналі. Під моделюванням розуміється процес побудови, вивчення і застосування моделей. Воно тісно пов'язане з такими категоріями, як абстракція, аналогія, гіпотеза та ін. Процес моделювання обов'язково включає і побудову абстракцій, і висновки за аналогією, і конструювання наукових гіпотез. Головна особливість моделювання в тому, що це метод опосередкованого пізнання за допомогою об'єктів-заступників[2]. Модель виступає як деякий інструмент пізнання, який дослідник ставить між собою і об'єктом, і за допомогою якого вивчає те, що його цікавить. Саме ця особливість методу моделювання визначає специфічні форми використання абстракцій, аналогій, гіпотез, інших типів і методів пізнання. У найзагальнішому випадку при побудові моделі дослідник відкидає ті характеристики, параметри об'єкта-оригіналу, які не важливі для вивчення об'єкта. Вибір характеристик об'єкта-оригіналу, які при цьому зберігаються і увійдуть в модель, визначається цілями моделювання. Зазвичай такий процес абстрагування від несуттєвих параметрів об'єкта називають формалізацією. Більш детально, формалізація - це заміна реального об'єкта або процесу його формальним описом. Основна вимога до моделей - це їх відповідність реальним процесам або об'єктам, які заміщає модель[2]. Практично у всіх науках про природу, живої та неживої, про суспільство, побудова та використання моделей є потужним знаряддям пізнання. Реальні об'єкти і процеси бувають настільки багатогранні і складні, що найкращим (а іноді і єдиним) способом їх вивчення досить часто є побудова і дослідження моделі, що відображає лише якусь грань реальності і тому в багато разів більш простий, ніж ця реальність. Багатовіковий досвід розвитку науки довів на практиці корисність такого підходу. Більш конкретно, необхідність використання методу моделювання залежить від того, що багато

об'єктів (або проблеми, які стосуються цих об'єктів) безпосередньо досліджувати чи взагалі неможливо, або ж це дослідження вимагає занадто багато часу або коштів. У моделюванні є два різних підходи[2].

Модель може бути схожою копією об'єкта, виконаної з іншого матеріалу, в іншому масштабі, з відсутністю ряду деталей. Наприклад, це іграшковий кораблик, будиночок з сірників, дерев'яна модель літака в натуральну величину, яка використовується в авіаконструюванні і ін. Моделі такого роду називають натурними. Модель може, однак, відображати реальність більш абстрактно - словесним описом у вільній формі, описом, формалізованим з якихось правил, математичними співвідношеннями і т.п. Будемо називати такі моделі абстрактними[2].

1.2 Класифікація моделей, способи моделювання

Моделлю об'єкта можна назвати будь-який інший об'єкт, деякі властивості якого повністю або частково співпадають з властивостями вихідного. Слід розуміти, що вичерпно повної модель бути не може. Вона завжди обмежена і має лише відповідати цілям, які поставлені перед моделюванням, відображаючи рівно стільки властивостей вихідного об'єкта і в такій повноті, скільки необхідно для конкретного дослідження[2].

Математична модель - це образ досліджуваного об'єкта, що створюється за допомогою певних формальних (математичних) систем з метою вивчення (оцінки) певних властивостей даного об'єкту[2].

Комп'ютерна модель - це програмна реалізація математичної моделі, доповнена службовими програмами (наприклад: графічними). Дана модель має дві складові - програмну і апаратну. Програмна складова так само є абстрактною знаковою моделлю. Комп'ютерна модель проявляє властивості фізичної моделі, коли вона, а точніше її абстрактні складові - це програми - інтерпретуються фізичним пристроєм, комп'ютером [2]. В роботі сукупність комп'ютера і моделі програми називається "електронним еквівалентом досліджуваного об'єкта".

На сьогоднішній день класифікації різних авторів по одній і тій же підставі можуть відрізнятися. Часто вони відрізняються один від одного тільки використовуваною термінологією. Моделі можна умовно розділити на дві групи: матеріальні і ідеальні. Моделювання, таким чином, поділяється на предметне і абстрактне. Основними різновидами предметного моделювання є фізичне і аналогове моделювання[3].

Фізичним прийнято називати таке моделювання, при якому реальному об'єкту ставиться у відповідність його збільшена або зменшена копія. Ця копія створюється на основі теорії подібності, що і дає підстави стверджувати, що в моделі збереглися необхідні властивості. У фізичних моделях крім геометричних пропорцій може бути збережений матеріал або колірна гамма вихідного об'єкта, а також будь-які інші властивості, які потрібні для конкретного дослідження. Аналогове моделювання ґрунтується на заміні вихідного об'єкта об'єктом іншої фізичної природи, що володіє аналогічним поведінкою. І фізичне, і аналогове моделювання в якості основного методу дослідження припускають проведення натурального експерименту з моделлю, але цей експеримент виявляється в якомусь сенсі більш привабливим, ніж експеримент з вихідним об'єктом[3].

На ідеї моделювання базується будь-який метод наукового дослідження, як теоретичний, так і експериментальний(практичний). Найбільш загальним розподілом всіх видів моделей буде поділ за методом наукового дослідження (за законом функціонування). Експериментальний метод пізнання використовує матеріальне моделювання (воно ж - предметне, натурне, фізичне, реальне). Матеріальні моделі функціонують за законами об'єктивної природи. Теоретичний метод пізнання використовує умовне моделювання (воно ж - ідеальне, як протилежність матеріального, воно ж - логічне, так як моделі, отримані таким методом, функціонують за законами логіки в свідомості людини) [3].

Уявне моделювання так само називається інформаційним моделюванням. Інформаційна модель протиставляється матеріальній і визначається як "сукупність інформації, що характеризує властивості і стан об'єкту, процесу, явища, а також взаємозв'язок із зовнішнім світом". Але існує й інше розуміння терміну "інформаційна модель". Зміна раніше сформованого значення терміну може призвести до термінологічної плутанини[3]. Розглянемо, що раніше з'явилися визначення. У словнику, заснованому на багатьох авторитетних джерелах, інформаційна модель визначається як "формалізоване опис інформаційних структур і операцій над ними" і ототожнюється з "моделлю даних", а також більш вузько - як "параметричне представлення процесу циркуляції інформації, яка підлягає автоматизованій обробці в системі управління "[4]. Вимога формалізації вже має на увазі більш вузьке розуміння, ніж ідеальна модель. Аналогічне висловлювання зробили А.В. Могильов і Е.К. Хеннер. З огляду на сказане, на перший погляд доречніше було б назвати моделі, використовувані при теоретичному методі пізнання, уявними (відштовхуючись від визначення моделі) або логічним (по імені науки, що вивчає форми і закони мислення) [4]. Однак, останнім часом вживання поняття "інформаційна модель" в значенні уявної моделі стає вже звичним. Старі значення цього поняття цілком переносяться на поняття "модель даних", яке його, по суті, дублює. Таким чином, будемо вважати, що всі моделі за законом функціонування діляться на матеріальні і інформаційні.

Інформаційна(ідеальна) модель - це в широкому сенсі будь-який образ об'єкта, уявний або умовний. Уявне моделювання зводиться до інформаційних процесів. Форма існування інформації визначається двома факторами: способом кодування (алфавітом і комбінаторикою) і матеріальним носієм. Алфавіт кодування частково визначає ступінь вивченості модельованого об'єкту. З цього випливає, що при класифікації уявних моделей важливо розрізняти їх саме за способом подання (схема 1) [4].

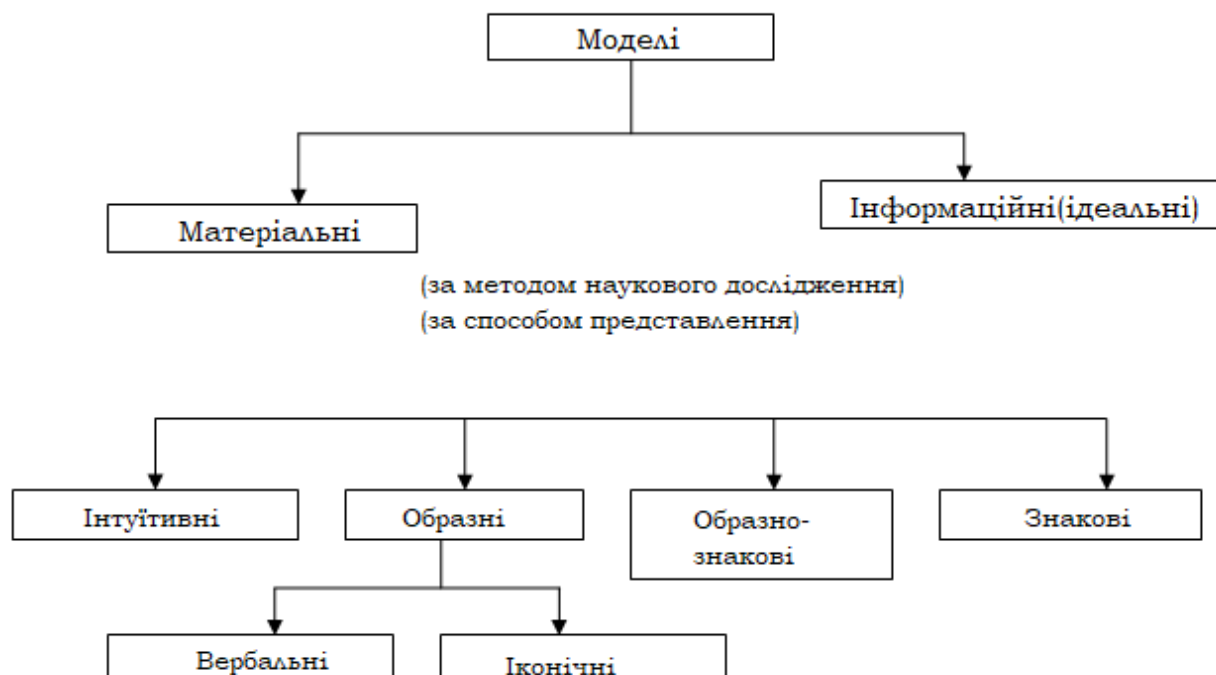


Схема 1 – Класифікація моделей

Інтуїтивне моделювання - це уявне представлення про об'єкт. Алфавітом кодування інформації для інтуїтивних моделей є система понять, а носієм - нервова система людини, мозок[4]. Поняття інтуїтивного моделювання використовується, при неможливості дати опис використовуваної моделі, хоча робляться спроби описати якась подія з використанням цієї моделі. Життєвий досвід кожної людини - його інтуїтивна модель навколишнього світу; музична тема в мозку композитора – це інтуїтивна модель музичного твору[4].

Образне моделювання - це вираження властивостей оригіналу за допомогою наочних чуттєвих образів, описаних природною мовою або зображених малюнком. Носій інформації може знаходитися і поза людиною. Приклади: художні полотна, фотографії, текст, кінофільми, усні розповіді, фізичні моделі: модель атома, запропонована Резерфордом і Бором, інші кульки молекул в кінетичної теорії газів. Оскільки при науковому моделюванні поняття найчастіше кодуються словами і малюнками, то цей вид моделювання ще називають іконічним або вербальним. Стає можливим

вважати вербальне і іконічне моделювання різновидами образного моделювання за способом кодування (за способом подання) [4].

Образно-знакове моделювання використовує знакові образи будь-якого виду: схему, граф, креслення, графік, план, карту і т.п. Приклади: шкільна карта, план квартири, стовпчаста діаграма співвідношення голосів виборців, семантична мережа понять, родовідне дерево, блок-схема алгоритму, класифікаційна схема. Глобус можна розглядати як сукупність двох моделей в одному реальному об'єкті: матеріальну подібну модель земної кулі як фізичного тіла і інформаційну образно-знакову модель розташування будь-чого на його поверхні [4]. Знакове (символічне) моделювання використовує умовні знаки, спеціальні символи, літери, цифри і передбачає сукупність законів оперування з вибраними знаками. Приклади: загальна схема опису системи мови або будь-якої його підсистеми, фізичні або хімічні формули, математичні вирази і рівняння, теорія музики, нотний запис і т.д.[5]. З цього видно, що образно-знакове моделювання є проміжним між образним і знаковим в різного ступеня для кожної конкретної моделі має риси і того й іншого. Найважливішим видом знакового моделювання є математичне моделювання. Абстрагуючись від фізичної природи об'єктів, математика вивчає точні(ідеальні) об'єкти. Далі зупинимося докладніше на математичному моделюванні і його найважливішою різновиди - комп'ютерному моделюванні[5].

1.3 Цілі і завдання комп'ютерного моделювання

Модель створюється, головним чином, для досліджень, які на реальному об'єкті проводити або неможливо, або незручно, або економічно не вигідно. Виділимо кілька основних цілей створення моделей, потім зупинимося докладніше на типах дослідження. Отже, цілі комп'ютерного моделювання:

- модель як засіб осмислення допомагає виявити взаємозалежності змінних, характер їх зміни в часі, знайти існуючі закономірності[5]. При складанні моделі стає більш зрозумілою структура досліджуваного об'єкта,

стають очевидними важливі причинно-наслідкові зв'язки. В процесі моделювання поступово відбувається поділ властивостей вихідного об'єкта на більш важливі і другорядні з точки зору поставлених до моделі вимог. У роботі з вихідним об'єктом необхідно виділити тільки ті риси, які мають безпосереднє відношення до сторони функціонування, що представляє інтерес для дослідження[5];

- модель як засіб прогнозування дозволяє навчитися передбачати поведінку об'єкта і керувати ним, відчуваючи різні варіанти управління на моделі. Експериментувати з реальним об'єктом буває незручно і навіть небезпечно. Експеримент може виявитися неможливим в силу наступних причин: великий тривалості експерименту, ризику пошкодження, або знищення об'єкта, відсутність об'єкта, коли він знаходиться на стадії проектування[5];

- побудовані моделі можуть використовуватися для знаходження оптимальних співвідношень параметрів, а також для дослідження особливих (критичних) режимів роботи реального об'єкта;

- також модель може в деяких випадках замінювати вихідний об'єкт при навчанні, наприклад, використовуватися в якості тренажера для підготовки користувачів до подальшої роботи в реальних умовах, або виступати в якості досліджуваного об'єкта в віртуальній лабораторії[5]. Моделі, реалізовані у вигляді виконуваних модулів, застосовуються і як імітатори об'єктів управління при стендових випробуваннях систем управління, і, на ранніх стадіях проектування, замінюють самі майбутні апаратно реалізовані системи управління. Процес дослідження закінчується, коли дослідник знаходить сукупність значень параметрів об'єкта, що задовольняє заданому критерію із заданою вірогідністю. Проведення таких досліджень називається обчислювальним експериментом [5]. Його можна уявити як послідовність наступних основних кроків:

- Виділення істотних для даного дослідження властивостей вихідного об'єкта і побудова математичної моделі.

- Проектування і налагодження комп'ютерної моделі.

- Оцінка адекватності побудованої комп'ютерної моделі. Як правило, оцінка адекватності приводить до перегляду вимог до моделі і поверненню на етап 1 - модель будується спочатку з урахуванням внесених змін. Дослідження моделі. Аналіз отриманих результатів[6]. Підведення підсумків моделювання не виключає можливих додаткових змін у разі, коли запланованих експериментів виявилось недостатньо для завершення робіт. Побудова математичної моделі, як правило, виконується вручну. Наведемо приклад побудови: нехай деякий об'єкт Q володіє деякою властивістю, що цікавить нас - S_0 . Для отримання математичної моделі, яка описує дане властивість необхідно:

- Визначити показник даної властивості (тобто визначити міру властивості в деякій системі вимірювання) [6].

- Встановити перелік властивостей S_1, \dots, S_m , з якими властивість S_0 пов'язано деякими відносинами (це можуть бути внутрішні властивості об'єкта і властивості зовнішнього середовища, що впливають на об'єкт).

- Описати в обраній форматній системі властивості зовнішнього середовища, як зовнішні чинники x_1, \dots, x_n , що впливають на шуканий показник Y , внутрішні властивості об'єкта, такі як параметри z_1, \dots, z_r , а невраховані властивості віднести до групи невраховуваних факторів (w_1, \dots, w_s) [6].

- З'ясувати, за можливості, закономірні відносини між Y і всіма враховуються факторами і параметрами, і скласти математичний опис (модель). В узагальненому вигляді схема такого опису (моделювання) показана на рис. 1[6].

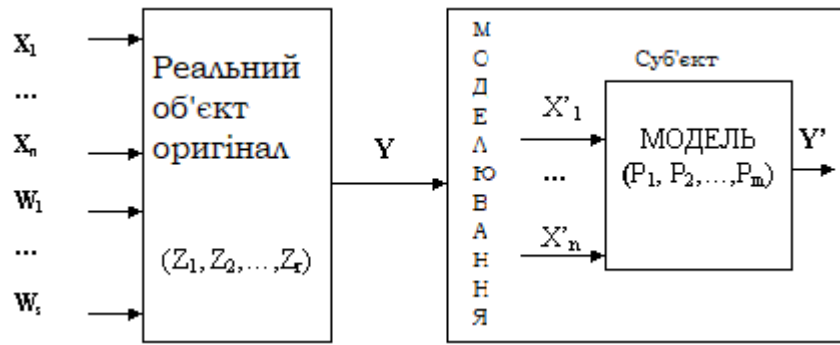


Рисунок 1.1 - Моделювання способу використовуваного об'єкта

Як показано на рисунку 1.1 реальний об'єкт характеризується наступними функціональними ставленням між показниками його властивостей:

$$Y = f(x_1, \dots, x_n, z_1, \dots, z_r, w_1, \dots, w_s). \quad (1)$$

Однак в моделі відображаються тільки ті фактори і параметри оригінального об'єкта, які мають істотне значення для розв'язання досліджуваної проблеми[6]. Крім того, вимірювання істотних факторів і параметрів практично завжди містять помилки, викликані неточністю вимірювальних приладів і незнанням деяких факторів. В силу цього математична модель є тільки наближеним описом властивостей досліджуваного об'єкта[7]. А математичну модель можна визначити ще і як абстракцію досліджуваної реальної суті[6]. Моделі зазвичай відрізняються від оригіналів по природі своїх внутрішніх параметрів. Подоба полягає в адекватності реакції Y моделі і оригіналу на зміну зовнішніх факторів x_1, \dots, x_n . Тому в загальному випадку математична модель являє собою функцію

$$Y' = f(x'_1, \dots, x'_n, p_1, \dots, p_m), \quad (2)$$

де p_1, \dots, p_m - це внутрішні параметри моделі, адекватні параметрам оригіналу [6]. Залежно від застосовуваних методів математичного опису досліджуваних об'єктів (процесів) математичні моделі бувають аналітичні, імітаційні, логічні, графічні, автоматні та т.д[7].

Головним питанням математичного моделювання є питання про те, наскільки точно складена математична модель відображає відносини між

факторами, які враховуються, параметрами і показником Y оцінюється властивості реального об'єкта, тобто на скільки точно рівняння (2) відповідає рівнянню (1) [7].

Іноді рівняння (2) може бути отримано одразу в явному вигляді, наприклад, у вигляді системи диференціальних рівнянь, або у вигляді інших явних математичних співвідношень[7].

У більш складних випадках вид рівняння (2) невідомий і завдання дослідника полягає, перш за все, в тому, щоб знайти це рівняння. При цьому до числа варійованих параметрів x^1, \dots, x^n , відносять все враховуються зовнішні фактори та параметри досліджуваного об'єкта, а до числа шуканих параметрів відносять внутрішні параметри моделі p_1, \dots, p_m , що зв'язують фактори x^1, \dots, x^n , з показником Y 'найбільш правдоподібним ставленням. Вирішенням цієї проблеми займається теорія експерименту. Суть цієї теорії полягає в тому, щоб, ґрунтуючись на вибіркових вимірах значень параметрів x^1, \dots, x^n , і показника Y ', знайти параметри p_1, \dots, p_m , при яких функція (2) найбільш точно відображає реальну закономірність (1) [7].

Крім рівнянь і формул в математичних моделях часто беруть участь залежності, отримані експериментально і задаються у вигляді таблиць. У цьому випадку додатково доводиться вивчати вплив наведених залежностей на властивості шуканого рішення. На даному етапі доцільно прагнути до того, щоб, з огляду на всі істотні фактори, модель, проте, була якомога простіше[7].

Далі необхідно побудувати комп'ютерну модель. В даний час існує безліч систем або пакетів моделювання, що автоматизують процес перекладу математичної моделі в моделює програму (жирної стрілкою на рис.1.2). В системі моделювання математична модель записується на деякій формальній вхідній мові моделювання і потім автоматично, при допомогою відповідного транслятора, перекладається на машинний мову[7].

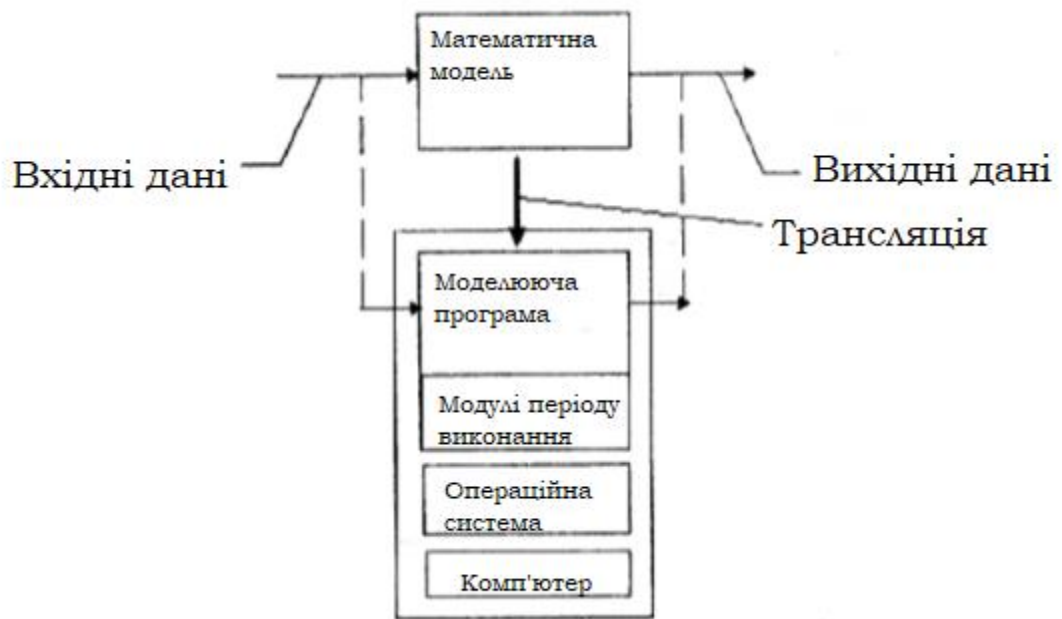


Рисунок 1.2 Побудова комп'ютерної моделі

У моделюючу програму крім згенерованого коду, необхідного для реалізації конкретної моделі, включають вже готові модулі підтримки періоду виконання, що надаються системою моделювання (бібліотеки, графічні функції). Система моделювання повинна також автоматично переводити вхідні дані моделі, записані в формі, зручній для аналізу людиною, в машинну форму подання і аналогічну операцію виробляти над вхідними даними в машинної формі; тобто інтерпретувати вхідні і вихідні дані [7].

Оцінка адекватності комп'ютерної моделі передбачає в якості обов'язкового етапу проведення спеціальних чисельних експериментів, результати яких заздалегідь відомі. Для перевірки правильності моделі можуть використовуватися відомі експериментальні залежності, існуючі оцінки рішення, а також вручну знайдені частинні розв'язки модельних рівнянь.

В результаті проведення цих експериментів виявляються помилки і неточності математичної моделі. Неточність в демонструваній поведінці може означати, що неврахований який-небудь істотний фактор. Після внесення виправлень доведеться повторити всі експерименти з моделлю заново. Сучасні

системи моделювання дозволяють здійснювати ці багаторазово повторювані дії набагато швидше[8].

Відтворення поведінки модельованої системи на інтервалі модельного часу $[0, T]$ при фіксованих значеннях параметрів моделі називаються елементарним досвідом або виконанням моделі. Результатом виконання моделі є знаходження значень всіх змінних моделі в кінцевий момент часу T і побудова таблиць значень змінних на зазначеному інтервалі для проміжних значень часу[8].

Кінцевий момент може бути побічно прив'язаний до якоїсь конкретної події. Природно, що в більшості випадків для досягнення бажаного результату одного досвіду або одноразового виконання моделі недостатньо. Виконання моделі так само використовується для вирішення завдань параметричної оптимізації.

Окремим завданням є знаходження особливих значень коефіцієнтів рівнянь моделі, якісно змінюючи характер її поведінки. Такі дослідження зазвичай проводять, в режимі, який є для моделі аварійним і може привести до краху[8].

Успіх дослідження багато в чому залежить від можливості автоматизованого обчислювального експерименту. Сучасні пакети моделювання організують його, спираючись на концепцію віртуального стенду. Обчислювальний експеримент можна трактувати як роботу на віртуальному випробувальному стенді, де розміщуються блоки модельованої системи, а також віртуальна вимірювальна апаратура[8].

З опису процесу обчислювального експерименту випливає, що моделювання - процес циклічний, в якому одні й ті ж операції повторюються багаторазово.

1.4 Комп'ютерне математичне моделювання: поняття, цілі та етапи

Почнемо з терміну «комп'ютерна модель». В даний час під комп'ютерною моделлю найчастіше розуміють:

- умовний образ об'єкта чи деякої системи об'єктів (або процесів), описаний за допомогою взаємозалежних комп'ютерних таблиць, блок-схем, діаграм, графіків, малюнків, анімаційних фрагментів, гіпертекстів і т. д., і відображає структуру і взаємозв'язки між елементами об'єкта. Комп'ютерні моделі такого виду ми будемо називати структурно-функціональними[8];

- окрему програму, сукупність програм, програмний комплекс, що дозволяє за допомогою послідовності обчислень і графічного відображення їх результатів, відтворювати (імітувати) процеси функціонування об'єкта, системи об'єктів за умови впливу на об'єкт різних, як правило випадкових, факторів. Такі моделі ми будемо далі називати імітаційними моделями[8].

Історично сталося так, що перші роботи з комп'ютерного моделювання, або, як говорили раніше, моделювання на ЕОМ, були пов'язані з фізикою, де за допомогою моделювання вирішувався цілий ряд завдань гідравліки, фільтрації, теплопереносу і теплообміну, механіки твердого тіла і т.д., моделювання в основному представляло собою рішення складних нелінійних задач математичної фізики за допомогою ітераційних схем, за винятком хіба що тих завдань, де використовувався метод Монте-Карло, і по суті було воно, звичайним математичним моделюванням[8]. Успіхи математичного моделювання в фізиці сприяли поширенню його на завдання хімії, електроенергетики, біології та деякі інші дисципліни, причому схеми моделювання не дуже відрізнялися один від одного. Складність вирішуваних задач на основі моделювання завжди обмежувалася лише потужністю наявних ЕОМ.

Треба зауважити, що подібний вид моделювання досить широко поширений і в наш час. Більш того, за час розвитку методів моделювання на ЕОМ при вирішенні завдань фундаментальних дисциплін та суміжних

предметних областей накопичені цілі бібліотеки підпрограм і функцій, що полегшують застосування і розширюють можливості моделювання. І все ж в даний час поняття "комп'ютерне моделювання" зазвичай пов'язують не з фундаментальними дисциплінами, а в першу чергу з системним аналізом - напрямком кібернетики, вперше заявив про себе на початку 50-х років при дослідженні складних систем в біології, макроекономіці, при створенні автоматизованих економіко-організаційних систем управління[8].

Комп'ютерне моделювання вимагає абстрагування від конкретної природи явищ, побудови спочатку якісної, а потім і кількісної моделі. Після цього відбувається проведення серії обчислювальних експериментів на комп'ютері, інтерпретація результатів, зіставлення результатів моделювання з поведінкою досліджуваного об'єкта, подальше уточнення моделі і т.д.

До основних етапів комп'ютерного моделювання відносяться:

постановка задачі, визначення об'єкта моделювання;

розробка концептуальної моделі, виявлення основних елементів системи та елементарних актів взаємодії;

формалізація, тобто перехід до математичної моделі; створення алгоритму та написання програми;

планування та проведення комп'ютерних експериментів;

аналіз і інтерпретація результатів[8].

Розрізняють аналітичне та імітаційне моделювання. Аналітичними називаються моделі реального об'єкта, що використовують алгебраїчні, диференціальні та інші рівняння, а також передбачають здійснення однозначної обчислювальної процедури, що призводить до їх точного розв'язання. Імітаційними називаються математичні моделі, які відтворюють алгоритм функціонування досліджуваної системи шляхом послідовного

виконання великої кількості елементарних операцій. Принципи моделювання полягають у наступному:

1. Принцип інформаційної достатності. При повній відсутності інформації про об'єкт побудувати модель неможливо. При наявності повної інформації моделювання позбавлене сенсу. Існує рівень інформаційної достатності, при досягненні якого може бути побудована модель системи.

2. Принцип здійсненності. Створювана модель повинна забезпечувати досягнення поставленої мети дослідження за кінцевий час.

3. Принцип множинності моделей. Будь-яка конкретна модель відображає лише деякі сторони реальної системи. Для повного дослідження необхідно побудувати ряд моделей досліджуваного процесу, причому кожна наступна модель повинна уточнювати попередню[8].

4. Принцип системності. Досліджувана система подана в вигляді сукупності взаємодіючих між собою підсистем, які моделюються стандартними математичними методами. При цьому властивості системи не є сумою властивостей її елементів.

5. Принцип параметризації. Деякі підсистеми модельованої системи можуть бути охарактеризовані єдиним параметром: вектором, матрицею, графіком, формулою[8].

Комп'ютерне моделювання систем часто вимагає рішення диференціальних рівнянь. Важливим методом є метод сіток, що включає в себе метод кінцевих різниць Ейлера. Він полягає в тому, що область безперервної зміни одного або декількох аргументів замінюють кінцевим безліччю вузлів, що утворюють одновимірну або багатовимірну сітку, і працюють з функцією дискретного аргументу, що дозволяє наближено обчислити похідні і інтеграли. При цьому нескінченно малі збільшення функції $f = f(x, y, z, t)$ і збільшення її аргументів замінюються малими, але кінцевими різницями.

Математична модель висловлює суттєві риси об'єкту або процесу мовою рівнянь і інших математичних засобів. Власне кажучи, сама математика зобов'язана своїм існуванням тому, що вона намагається відбити, т. Е. Промодельовати на своєму специфічному мовою закономірності навколишнього світу. Величезний поштовх розвитку математичного моделювання дало поява ЕОМ, хоча сам метод зародився одночасно з математикою тисячі років тому[9].

Математичне моделювання як таке аж ніяк не завжди вимагає комп'ютерної підтримки. Кожен фахівець, що професійно займається математичним моделюванням, робить все можливе для аналітичного дослідження моделі. Аналітичні рішення (т. Е. Представлені формулами, що виражають результати дослідження через вихідні дані) зазвичай зручніше і інформативніше чисельних. Однак, можливості аналітичних методів вирішення складних математичних задач дуже обмежені і, як правило, ці методи набагато складніше чисельних[9]. У нашому курсі домінують чисельні методи, які реалізуються на комп'ютерах. Відзначимо, що поняття «аналітичне рішення» і «комп'ютерне рішення» аж ніяк не протистоять один одному, так як:

а) все частіше комп'ютери при математичному моделюванні використовуються не тільки для чисельних розрахунків, але і для аналітичних перетворень;

б) результат аналітичного дослідження математичної моделі часто виражений настільки складною формулою, що при погляді на неї не складається наочного сприйняття описуваного нею процесу. Цю формулу (добре ще, якщо просто формулу!) Потрібно протабулювати, уявити графічно, проілюструвати в динаміці, іноді навіть озвучити, тобто виконати те, що називається «візуалізацією». Очевидно, можливості сучасних комп'ютерів найкращим чином відповідають цьому завданню[9].

Розглянемо процес комп'ютерного математичного моделювання, що включає чисельний експеримент з моделлю (рис. 1.3). Перший етап - визначення цілей моделювання. Основні з них такі:

1) Розуміння

Модель в цій ситуації потрібна для того, щоб зрозуміти, як влаштований конкретний об'єкт, яка його структура, основні властивості, закони розвитку і взаємодії з навколишнім світом.

2) Управління

Модель потрібна для того, щоб навчитися управляти об'єктом (або процесом) і визначити найкращі способи управління при заданих цілях і критеріях;

3) Прогнозування

Модель використовується для того, щоб прогнозувати прямі і непрямі наслідки впливу на об'єкт заданими способами[9].

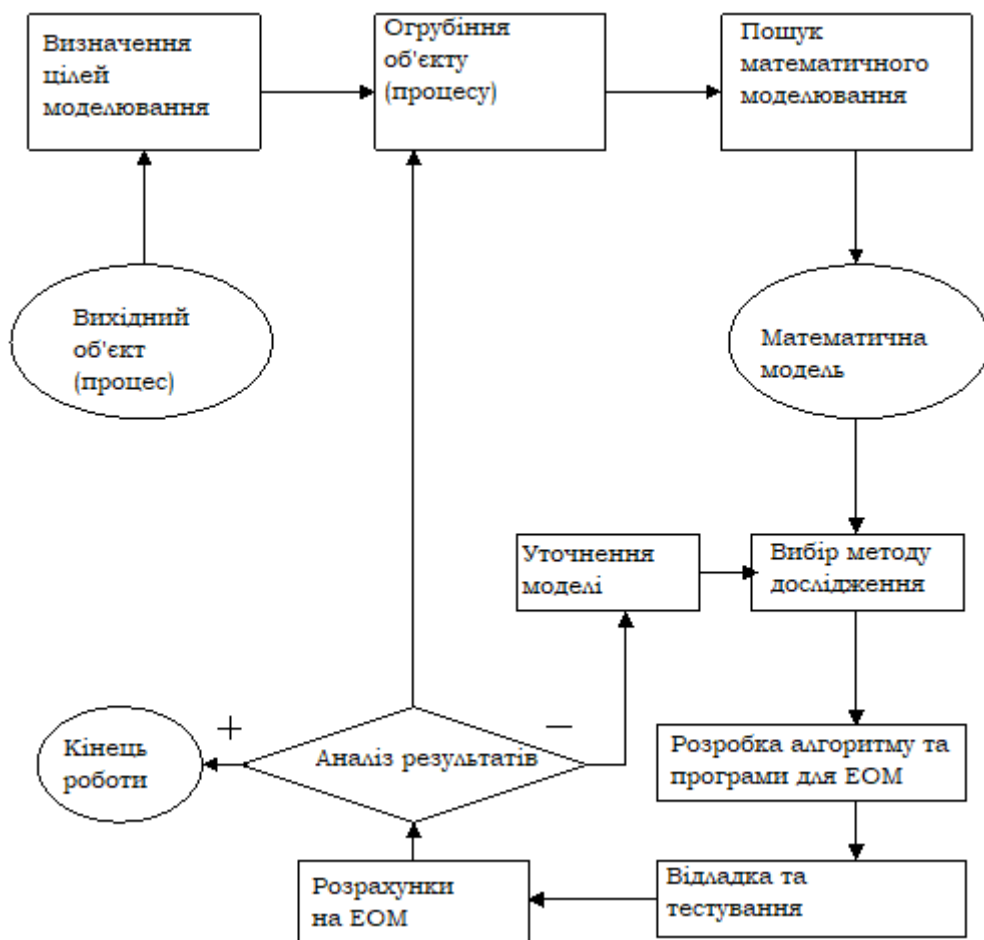


Рисунок 1.3 Загальна схема процесу комп'ютерного математичного моделювання

Пояснимо це на прикладах. Нехай об'єкт дослідження - взаємодія потоку рідини або газу з тілом, що є для цього потоку перешкодою. Досвід показує, що сила опору потоку з боку тіла зростає зі зростанням швидкості потоку, але при деякій досить високій швидкості ця сила стрибком зменшується з тим, щоб з подальшим збільшенням швидкості знову зрости. Що ж сталося, зумовивши зменшення сили опору[9]? Математичне моделювання дозволяє отримати чітку відповідь: в момент стрибкоподібного зменшення опору вихори, що утворюються в потоці рідини або газу позаду обтічного тіла, починають відриватися від нього і нестися потоком[9]. Приклад зовсім з іншої області: мирно співіснували зі стабільними численностями популяції двох видів особин, що мають загальну кормову базу, «раптом» починають різко міняти чисельність - і тут математичне моделювання дозволяє (з певною часткою достовірності) встановити причину явища, або, по крайній мере, спростувати певну гіпотезу про його причини. Вироблення концепції управління об'єктом - інша можлива мета моделювання. Який режим польоту літака вибрати для того, щоб політ був цілком безпечним і економічно найбільш вигідним? Як скласти графік виконання сотень видів робіт на будівництві великого об'єкта, щоб воно закінчилося в максимально короткий термін? Безліч таких проблем систематично виникає перед економістами, конструкторами, вченими[9].

Нарешті, прогнозування наслідків тих чи інших впливів на об'єкт може бути як простою справою (в нескладних фізичних системах), так і надзвичайно складною - на межі здійсненності - в системах біолого-економічних, соціальних. Якщо відносно легко відповісти на питання про зміну режиму поширення тепла в тонкому стрижні в залежності від змін в котрій становить його сплав, то незрівнянно важче простежити (передбачити) екологічні та кліматичні наслідки будівництва великої ГЕС або соціальні наслідки змін

податкового законодавства[9]. Можливо, і тут методи математичного моделювання будуть надавати в майбутньому більш значну допомогу[9].

Складемо список величин, від яких залежить поведінка об'єкта або хід процесу, а також тих величин, які бажано отримати в результаті моделювання. Позначимо перші з них (вхідні) через x_1, x_2, \dots, x_n „; другі (вихідні) через y_1, y_2, \dots, y_n . Символічно поведінку об'єкта або процесу можна представити у вигляді де F_j - ті дії, які слід зробити над вхідними параметрами, щоб отримати результати. Хоча запис нагадує позначення функції, ми тут використовуємо її в більш широкому сенсі[9]. Лише в найпростіших ситуаціях тут F є функція в звичайному сенсі; найчастіше вона висловлює лише наявність певної зв'язку між вхідними та вихідними параметрами моделі. Вхідні параметри x_i можуть бути відомі «точно», тобто піддаватися (по крайній мере, в принципі) вимірюванню однозначно і з будь-яким ступенем точності - тоді вони є детермінованими величинами[9]. Так, в класичній механіці, як складною не була б моделювана система, вхідні параметри детерміновані і, відповідно, детермінований процес еволюції такої системи. Однак в природі і суспільстві набагато частіше зустрічаються процеси іншого роду, коли значення вхідних параметрів відомі лише з певним ступенем імовірності, тобто ці параметри є імовірнісними (стохастичними), і, відповідно, випадковим є процес еволюції системи. Випадковий - не означає непередбачуваний[9]. Просто в цій ситуації характер дослідження і запитань різко змінюється - вони набувають вигляду «З якою ймовірністю ...?», «З яким математичним очікуванням ...?» і т.п. Прикладів випадкових процесів не злічити як в науці, так і в повсякденному житті (сили, що діють на літак, що летить у вітряну погоду; перехід вулиці при великому потоці транспорту і т.д.). Для стохастичної моделі вихідні параметри можуть бути як величинами імовірнісними, так і однозначно визначеними. Наприклад, на перехресті вулиць можна очікувати зеленого сигналу світлофора і півхвилини, і дві хвилини (з різною ймовірністю), але середній час очікування є величина цілком певна, і саме вона може бути об'єктом моделювання[9].

Найважливішим етапом моделювання є розділення вхідних параметрів за ступенем важливості впливу їх змін на вихідні. Такий процес називається ранжируванням (поділом по рангах). Найчастіше неможливо, та й не потрібно враховувати всі фактори, які можуть вплинути на значення цікавлять нас величин y_j . Від того, наскільки вміло виділені найважливіші фактори, залежить успіх моделювання, швидкість і ефективність досягнення мети. Виділити найбільш значущі чинники і відсіяти менш важливі може лише фахівець в тій предметній області, до якої належить модель. Так, досвідчений вчитель знає, що на успіх контрольної роботи впливає ступінь знання предмета і психологічний настрій класу; однак, впливають і інші фактори - наприклад, яким уроком за рахунком йде контрольна, яка в цей момент погода і т.д. - фактично проведено ранжування. Відкидання менш значущих чинників огрубляє об'єкт моделювання і сприяє розумінню його головних властивостей і закономірностей. Зазвичай визначити, чи адекватна модель можна тільки в процесі експериментів з нею, аналізу результатів початкового моделювання. На рис. 2 проілюстровані дві крайні ситуації: а) певний параметр x_i , дуже сильно впливає на результуючу величину y_i ; б) майже не впливає на неї. Ясно, що якщо все що представляють інтерес величини y_i реагують на x_i , так, як зображено на рис. 2б, то x_i , є параметром, який при першому підході може бути з моделі виключений. Якщо ж хоча б одна з величин y_i реагує на зміну x_i , так, як зображено на рис. 2а, то x_i можна виключати з числа значущих параметрів[9].

Наступний етап - пошук математичного опису. На цьому етапі необхідно перейти від абстрактного формулювання моделі до формулювання, що має конкретне математичне наповнення. У цей момент модель постає перед нами у вигляді рівняння, системи рівнянь, системи нерівностей, диференціального рівняння або системи таких рівнянь і т.д.

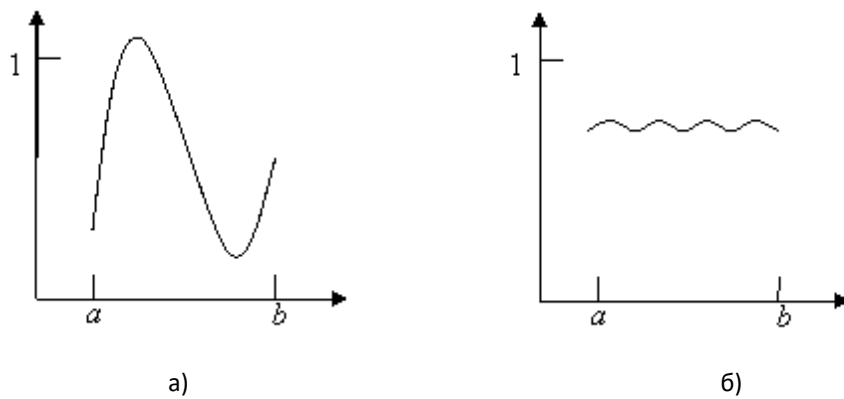


Рисунок 2. Варіанти ступені впливу величини x_i на результуючу величину y_j

Коли математична модель сформульована, потрібно вибрати метод її дослідження. Як правило, для вирішення однієї і тієї ж задачі є кілька конкретних методів, що розрізняються ефективністю, стійкістю і т.д. Від вірного вибору методу часто залежить успіх всього процесу[9].

Після розробки алгоритму та складання програми для ЕОМ необхідно вирішити з її допомогою найпростішу тестову задачу (бажано, з заздалегідь відомим відповіддю) з метою усунення грубих помилок. Це лише початок процедури тестування, яку важко описати формально вичерпним чином. По суті, тестування може тривати довго і закінчитися тоді, коли користувач за своїми професійними ознаками вважатиме програму вірною.

Потім слідує власне чисельний експеримент, і з'ясується, чи відповідає модель реального об'єкту (процесу). Модель адекватна реальному процесу, якщо деякі характеристики процесу, отримані на ЕОМ, збігаються з експериментальними із деяким ступенем точності. У разі невідповідності моделі реальному процесу повертаються до одного з попередніх етапів[9].

1.5 Різні класифікації математичних моделей

До класифікації математичних моделей можна підходити з різних позицій, поклавши в основу класифікації різні принципи. Можна класифікувати моделі за галузями наук (математичні моделі у фізиці, біології, соціології і т.д.) і за вживаним математичному апарату (моделі, засновані на використанні звичайних диференціальних рівнянь, диференціальних рівнянь в приватних

похідних, стохастичних методів, дискретних алгебраїчних перетворень і т. д.). Далі, якщо поцікавитися загальними закономірностями моделювання в різних науках (безвідносно до математичного апарату) і поставити на перше місце мети моделювання, то можна прийти до наступної класифікації:

- дескриптивні (описові) моделі;
- оптимізаційні моделі;
- багатокритеріальні моделі;
- ігрові моделі;
- імітаційні моделі.

Зупинимось на цій класифікації докладніше і пояснимо її на прикладах. Моделюючи рух комети, яка вторглася в Сонячну систему, ми описуємо ситуацію (передбачаємо траєкторію польоту комети, відстань, на якому вона пройде від Землі і т.д.), тобто ставимо чисто описові мети. У нас немає ніяких можливостей вплинути на рух комети, щось змінити в процесі моделювання.

В оптимізаційних моделях ми можемо впливати на процеси, намагаючись домогтися якоїсь мети. В цьому випадку в модель входить один або кілька параметрів, доступних нашому впливу. Наприклад, змінюючи тепловий режим в зерносховище, ми можемо прагнути підібрати такий, щоб досягти максимального збереження зерна, т. Е. Оптимізуємо процес[9].

Часто доводиться оптимізувати процес за кількома параметрами відразу, причому цілі можуть бути досить суперечливими. Наприклад, знаючи ціни на продукти і потреба людини в їжі, організувати харчування великих груп людей (в армії, літньому таборі і ін.) Як можна корисніше і якомога дешевше. Ясно, що ці цілі, взагалі кажучи, зовсім не збігаються, тобто при моделюванні буде кілька критеріїв, між якими треба шукати баланс. У цьому випадку говорять про багатокритеріальних моделях[9].

Ігрові моделі можуть мати відношення не тільки до дитячих ігор (в тому числі і комп'ютерним), але і до речей дуже серйозним. Наприклад, полководець перед битвою в умовах наявності неповної інформації про протистоять армії повинен розробити план, в якому порядку вводити в бій ті чи інші частини і т.п., з огляду на можливу реакцію супротивника. У сучасній математиці є спеціальний розділ - теорія ігор, що вивчає методи прийняття рішень в умовах неповної інформації.

Нарешті, буває, що модель великою мірою наслідує реальний процес, тобто імітує його. Наприклад, моделюючи динаміку чисельності мікроорганізмів в колонії, можна розглядати сукупність окремих об'єктів і стежити за долею кожного з них, ставлячи певні умови для його виживання, розмноження і т.д.[9]. При цьому іноді явне математичний опис процесу не використовується, замінюючись деякими словесними умовами (наприклад, після закінчення деякого відрізка часу мікроорганізм ділиться на дві частини, а іншого відрізка - гине). Інший приклад - моделювання руху молекул в газі, коли кожна молекула представляється у вигляді кульки, і задаються умови поведінки цих кульок при зіткненні один з одним і зі стінками (наприклад, абсолютно пружний удар); при цьому не потрібно використовувати ніяких рівнянь руху.

Можна сказати, що найчастіше імітаційне моделювання застосовується в спробі описати властивості великої системи за умови, що поведінка складових її об'єктів дуже просто і чітко сформульовано. Математичний опис тоді проводиться на рівні статистичної обробки результатів моделювання при знаходженні макроскопічних характеристик системи[9]. Такий комп'ютерний експеримент фактично претендує на відтворення натурального експерименту. На питання ж «навіщо це робити?» можна дати таку відповідь: імітаційне моделювання дозволяє виділити «в чистому вигляді» сліdstва гіпотез, закладених в наші уявлення про мікрособиттях, очистивши їх від неминучого в натурному експерименті впливу інших факторів, про які ми можемо навіть

не підозрювати. Якщо ж таке моделювання включає і елементи математичного опису подій на мікрорівні, і якщо дослідник при цьому не ставить завдання пошуку стратегії регулювання результатів (наприклад, управління чисельністю колонії мікроорганізмів), то відміну імітаційної моделі від дескриптивної досить умовно; це, швидше, питання термінології[9].

Ще один підхід до класифікації математичних моделей поділяє їх на детерміновані і стохастичні (ймовірні). У детермінованих моделях вхідні параметри піддаються вимірюванню однозначно і з будь-яким ступенем точності, тобто є детермінованими величинами. Відповідно, процес еволюції такої системи детермінований. В стохастичних моделях значення вхідних параметрів відомі лише з певним ступенем імовірності, тобто ці параметри є стохастичними; відповідно, випадковим буде і процес еволюції системи. При цьому, вихідні параметри стохастичною моделі можуть бути як величинами імовірнісними, так і однозначно визначаються[9].

Нарешті, якщо обмежитися безперервними детерміністськими моделями, то їх часто поділяють на системи з зосередженими параметрами і системи з розподіленими параметрами. Системи із зосередженими параметрами описуються за допомогою кінцевого числа звичайних диференціальних рівнянь для залежних від часу змінних. Простір станів має тут кінцеву розмірність (число ступенів свободи системи звичайно). На противагу цьому під системами з розподіленими параметрами розуміють системи, що описуються кінцевим числом диференціальних рівнянь в приватних похідних. Тут змінні стану в кожен момент часу є функції однієї або кількох просторових змінних. Простір станів має в цьому випадку нескінченну розмірність, тобто система має нескінченним числом ступенів свободи[9].

1.6 Тестування

Тестування програмного забезпечення – це процес, що використовується для виміру якості розроблюваного програмного

забезпечення. Зазвичай, поняття якості обмежується такими поняттями, як коректність, повнота, безпечність, але може містити більше технічних вимог, які описані в стандарті ISO 9126. Тестування - це процес технічного дослідження, який виконується на вимогу замовників, і призначений для вияву інформації про якість продукту відносно контексту, в якому він має використовуватись. До цього процесу входить виконання програми з метою знайдення помилок[9].

Якість не є абсолютною, це суб'єктивне поняття. Тому тестування не може повністю забезпечити коректність програмного забезпечення. Воно тільки порівнює стан і поведінку продукту зі специфікацією. При цьому треба розрізняти тестування програмного забезпечення і забезпечення якості програмного забезпечення, до якого належать усі складові ділового процесу, а не тільки тестування[9].

Існує багато підходів до тестування програмного забезпечення, але ефективно тестування складних продуктів – це по суті дослідницький процес, а не тільки створення і виконання рутинної процедури.

Тестування пронизує весь життєвий цикл ПЗ, починаючи від проектування і закінчуючи невизначено довгим етапом експлуатації. Ці роботи безпосередньо пов'язані із завданнями управління вимогами та змінами, адже метою тестування є якраз можливість переконатися у відповідності програм заявленим вимогам[9].

Тестування – процес також ітераційний. Після виявлення та виправлення кожної помилки обов'язково слід повторити тести, щоб переконатися у працездатності програми. Більше того, для ідентифікації причини виявленої проблеми може знадобитися проведення спеціального додаткового тестування. При цьому потрібно завжди пам'ятати про фундаментальний висновок, зроблений професором Едджером Дейкстри у 1972 році: "Тестування програм може служити доказом наявності помилок, але ніколи не доведе їхню відсутність!"[9].

Існує багато підходів до вирішення завдання тестування та верифікації ПЗ, але ефективне тестування складних програмних продуктів - це процес у великій мірі творчий, не зводиться до прямування строгими і чіткими процедурами або до створення таких.

З точки зору ISO 9126 якість (програмних засобів) можна визначити як сукупну характеристику досліджуваного ПЗ з урахуванням наступних складових: надійність, супроводжуваність, практичність, ефективність, мобільність, функціональність [10].

Більший список атрибутів і критеріїв можна знайти в стандарті ISO 9126 Міжнародної організації зі стандартизації. Склад і зміст документації, супутньої процесу тестування, визначається стандартом IEEE 829-1998 Standard for Software Test Documentation [11].

1.7 C++, як інструмент для комп'ютерного моделювання

Основні переваги мови для моделювання систем:

- висока швидкодія;
- використання на мультипроцесорних і гетерогенних (різнопроцесорних) системах;
- можливість прямого доступу до апаратних ресурсів ;
- наявність розвинених бібліотек спрощують створення модельованих систем;
- широке розповсюдження і переносимість.

Універсальність мови дозволяє використовувати її в різних задачах моделювання. Більш того, при розробці мови її ключові концепції класів і об'єктів були запозичені з мови Simula, спеціально призначеної, як впливає з назви, для комп'ютерного "симулювання"(моделювання).

Крім розробки самостійних програмних продуктів C ++ може використовуватися для розширення функціональності мов програмування Python, R, matlab, scilab і інших, які можуть бути використані в комп'ютерному моделюванні.

Мова C ++ є компільованою мовою (на відміну від різних мов і мов з компіляцією часу виконання). Оригінальний текст C ++ перетвориться компілятором мови в машинний код, виконуваний безпосередньо процесором. Ця властивість дозволяє писати програми для платформ, що не мають виконавчих систем часу виконання або виконуваних бібліотек, наприклад, для мікропроцесорів, сигнальних або графічних процесорів в тих випадках, коли ці платформи дозволяють максимально ефективно вирішити задачу.

Для C ++ існує безліч середовищ розробки, що включають в себе компілятори мови, які працюють під управлінням різних операційних систем: Windows, Linux, OSX (Mac) та ін., і створюють виконуваний код як для процесорів, на яких працюють самі, так само і для зовнішніх цільових процесорів. Мова є статично типізованою (static typing), тобто типи змінних визначаються на етапі компіляції. Допустимі неявні перетворення типів (слабка типізація або weak typing). Типи змінних задаються явно.

1.8 Висновки

В даному розділі проаналізовано сучасний стан розвитку комп'ютерного-математичного моделювання фізичних процесів та об'єктів. Було обрано мову програмування.

Розділ 2.ПРОЕКТНА ЧАСТИНА

2.1 Постановка задачі

Розробити алгоритм і комп'ютерну програму для моделювання роботи двопозиційного регулятора температури в теплообмінному апараті. Процес дослідити для різних співвідношень між сумарними теплоємностями середовищ в кожусі, через який підводиться тепло, і в кожусі, де відбувається реакція.

В залежності від конструктивних особливостей теплообмінника мають місце різні величини відтоку тепла. Дослідити, як впливають ці особливості на процес регулювання температури в середовищі, де знаходяться реагуючі речовини.

2.2 Математична постановка задачі

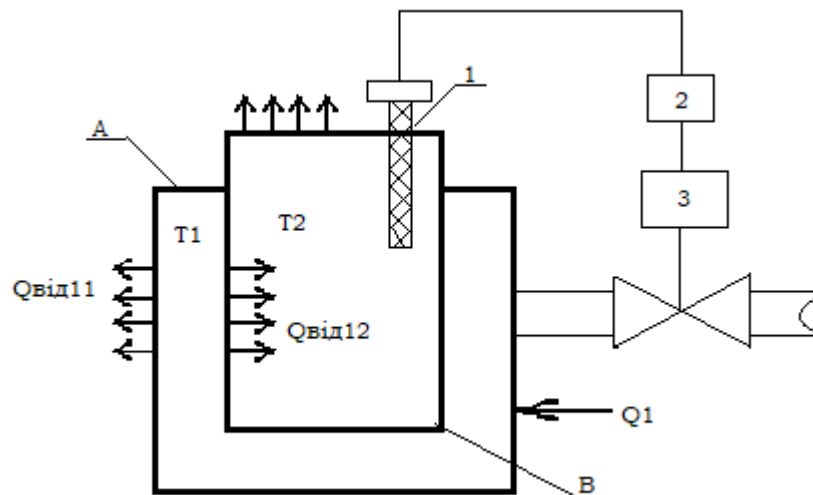


Рисунок 2.1 – Схема системи регулювання температури в теплообмінному апараті

На рисунку 2.1 приведена схема системи регулювання температури в теплообмінному апараті.

Умовні позначення на схемі:

A – кожух для підведення тепла;

B – посудина із реагуючими речовинами;

1 – датчик температури;

2 – регулятор температури;

3 – соленоїдний клапан для подачі теплоносія;

Q_1 – приток тепла;

$Q_{від11}$ – тепловідведення через кожух назовні;

$Q_{від12}$ – відведення тепла в посудину;

$Q_{від21}$ – відведення тепла назовні через кришку апарата;

T_1 – температура теплоносія;

T_2 – температура реагуючих речовин.

При відведенні тепла має місце так зване самовирівнювання, коли із зменшенням різниці температур між джерелом тепла і його споживачем, потік тепла зменшується і стає нульовим при різниці температур також нуль.

P_1, P_{12}, P_{21} – коефіцієнти самовирівнювання відповідно для $Q_{від11}, Q_{від12}, Q_{від21}$. Вони залежать від конструктивних особливостей апарату. Також від цих особливостей залежить сумарна теплоємність C_1 ємності А і C_2 – ємності В.

Об'єкт описується наступною системою диференціальних рівнянь

$$\frac{dT_1}{dt} = \frac{1}{C_1} [Q_1 - P_{11}T_1 - P_{12}(T_1 - T_2)] \quad (3)$$

$$\frac{dT_2}{dt} = \frac{1}{C_2} [P_{12}(T_1 - T_2) - P_{21}T_2] \quad (4),$$

де t - час

$$Q_1 = \text{const при } \{T_2 < (T_n - \Delta T) \text{ і } \frac{dT_2}{dt} < 0 \text{ } T_2 < (T_n + \Delta T) \text{ і } \frac{dT_2}{dt} > 0 \quad (5)$$

$$Q = 0 \text{ при } \{T_2 < (T_n + \Delta T) \text{ і } \frac{dT_2}{dt} > 0 \text{ } T_2 < (T_n - \Delta T) \text{ і } \frac{dT_2}{dt} < 0 \quad (6),$$

Де T_n – номінальна(задана) температура всередині посудини із реагентами,

ΔT – допустиме відхилення температури від номінального значення T_n .

Таким чином, ставиться задача отримати залежності $T_1(t), T_2(t)$ для заданих параметрів теплоносія при різних співвідношеннях P_{12}/P_{11} та C_1/C_2 .

Задача зводиться до обчислення $T_1(t), T_2(t)$ при заданих початкових умовах:

$t = t_0, T_1(t_0), T_2(t_0)$. Тобто треба розв'язати задачу Коші.

2.3 Вибір методу розв'язання задачі

В систему автоматичного регулювання температури (САР) входить двохпозиційний регулятор, який є нелінійним елементом. Це ускладнює аналітичне дослідження моделі. Тому для розв'язання задачі вибирається чисельний метод розв'язання задачі Коші, метод Рунге-Кутта четвертого порядку.

2.4 Хід розв'язання задачі

Система рівнянь, які описують теплообмінний апарат, уже виражені відносно похідних. Але в систему (5), (6) входить похідна $\frac{dT^2}{dt}$, що потребує додаткових операцій по чисельному диференціюванню. При цьому слід враховувати, що при нормальній роботі регулятора температури її значення буде змінюватися на інтервалі $T_n \pm \Delta T$. Як правило допуск ΔT на відхилення температури від номінального значення незначний порівняно із значенням T_n . Тому замість обчисленої чисельним методом похідної застосовується різниця між поточним і попереднім значенням температури в посудині.

2.5 Алгоритм розв'язання задачі

Під час описання алгоритму застосовуються ідентифікатори, які описані в таблиці 1.

Таблиця 1

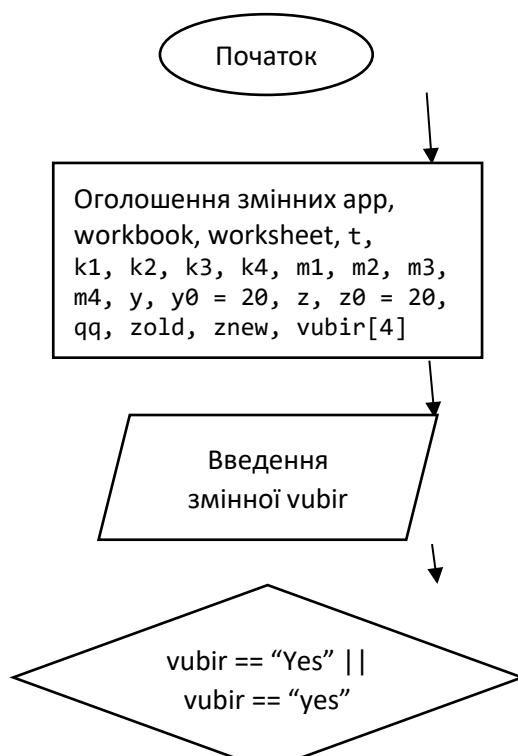
Змінна	Ідентифікатор	Пояснення
t, t0, tk, h	t, t0, tk, h	Час, його початкове і кінцеве значення, крок зміни часу
Q1	Q1	Приток тепла, ккал/хв*град.
P11, P12, P21	P11, P12, P21	Коефіцієнт самовирівнювання ккал/хв*град
C1, C2	C1, C2	Сумарні теплоємності для ємностей А і В відповідно

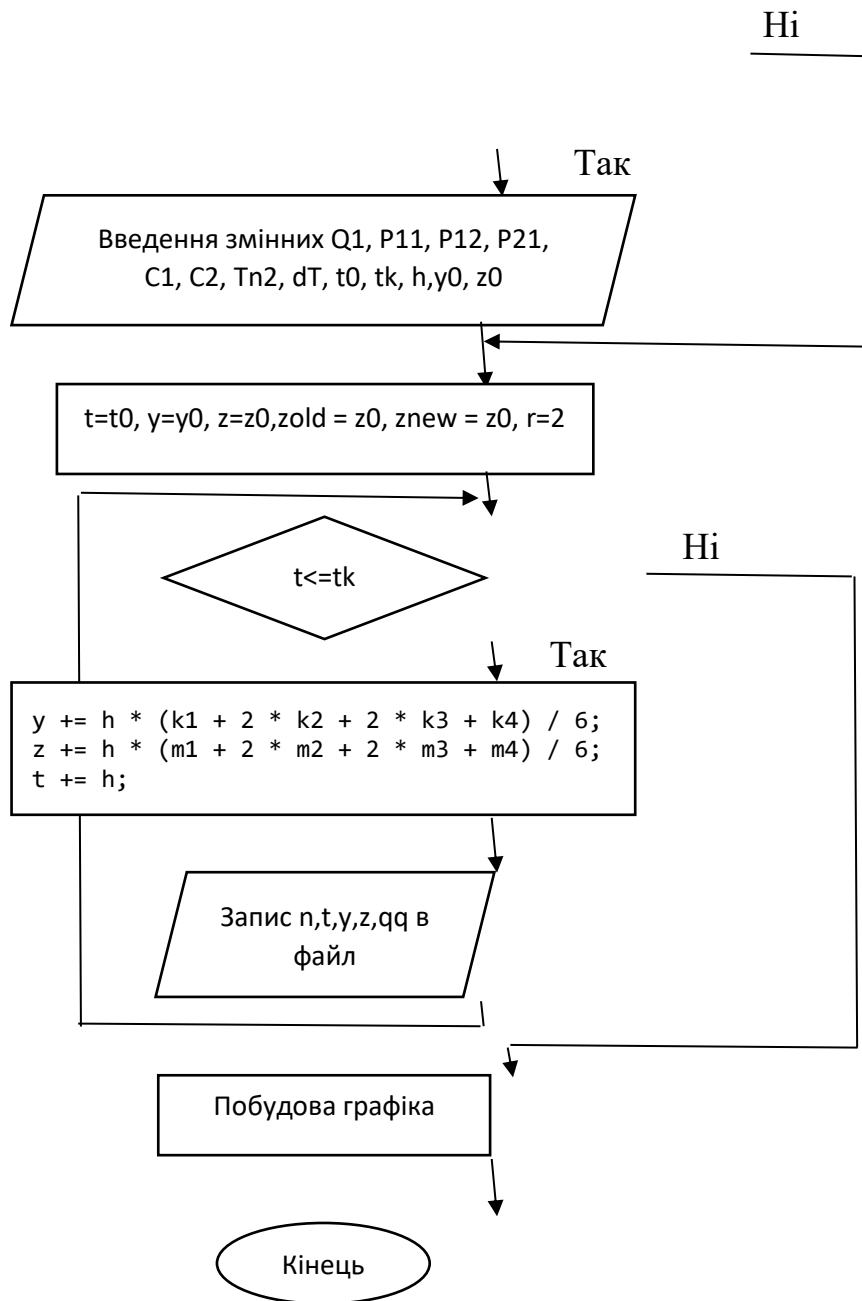
$T_H, \Delta T$	T_{n2}, dt	Номінальна температура допустиме відхилення температури i
$T1, T2$	y, z	Температури в кожусі i в посудині v
$T1(t_0), T2(t_0)$	y_0, z_0	Початкові температури
	$k1, k2, k3, k4$ $m1, m2, m3, m4$	Коефіцієнти для обчислення методом Рунге-Кутта
	$zold, znew$	Попереднє i поточне значення температури в посудині v
	qq	Поточне значення підведеного тепла

2.6 Алгоритми роботи програми

Блок-схема функції `main()` представлена на схемі 2.

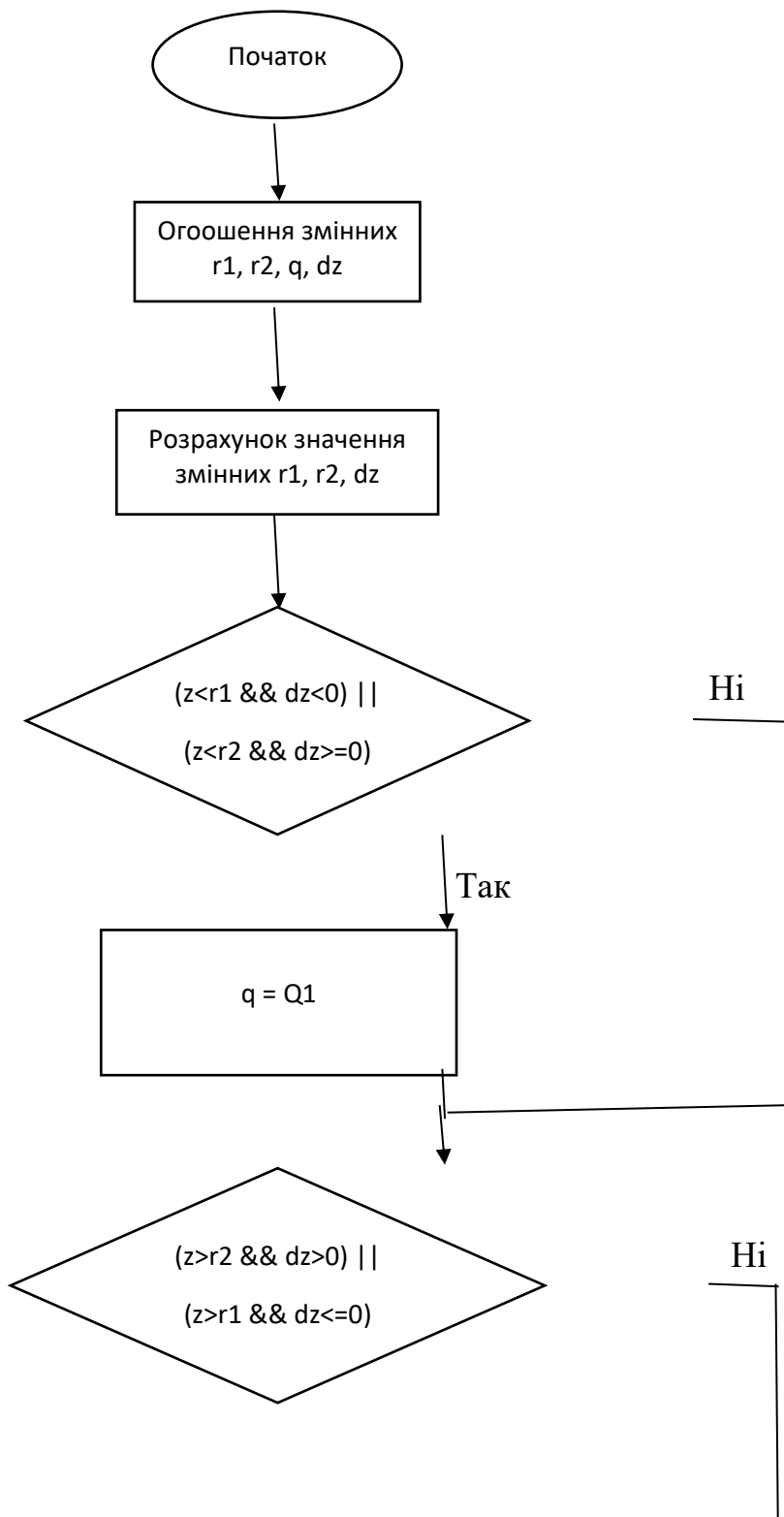
Схема 2

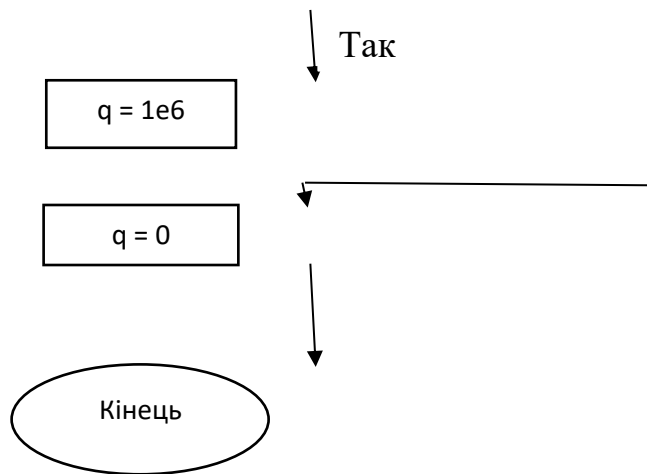




Блок-схема функції QQ() представлена на схемі 3.

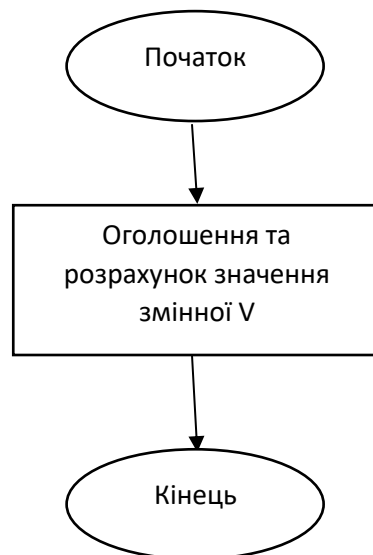
Схема 3





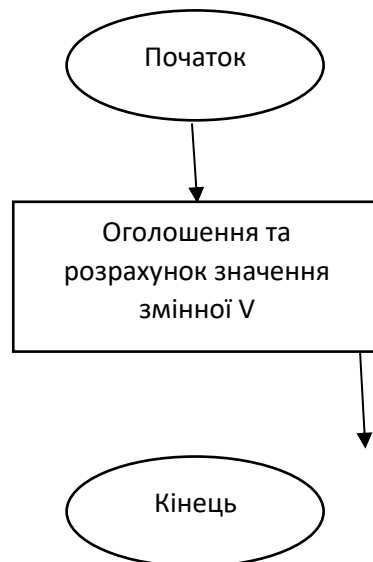
Блок-схема функції f1() представлена на схемі 4.

Схема 4



Блок-схема функції f2() представлена на схемі 5.

Схема 5



2.7 Налаштування властивостей проекту

Для початку потрібно увімкнути підтримку загальноомовного середовища виконання (CLR), для цього потрібно перейти по шляху «Проект»-«Свойства:sirobaba»-«C/C++»-«Общие», у відповідному полі вибрати «Поддержка общезыковой среды выполнения (CLR) (/clr)» та натиснути кнопку «Применить» та кнопку «ОК». Підключення CLR зображено на рис.3.1.

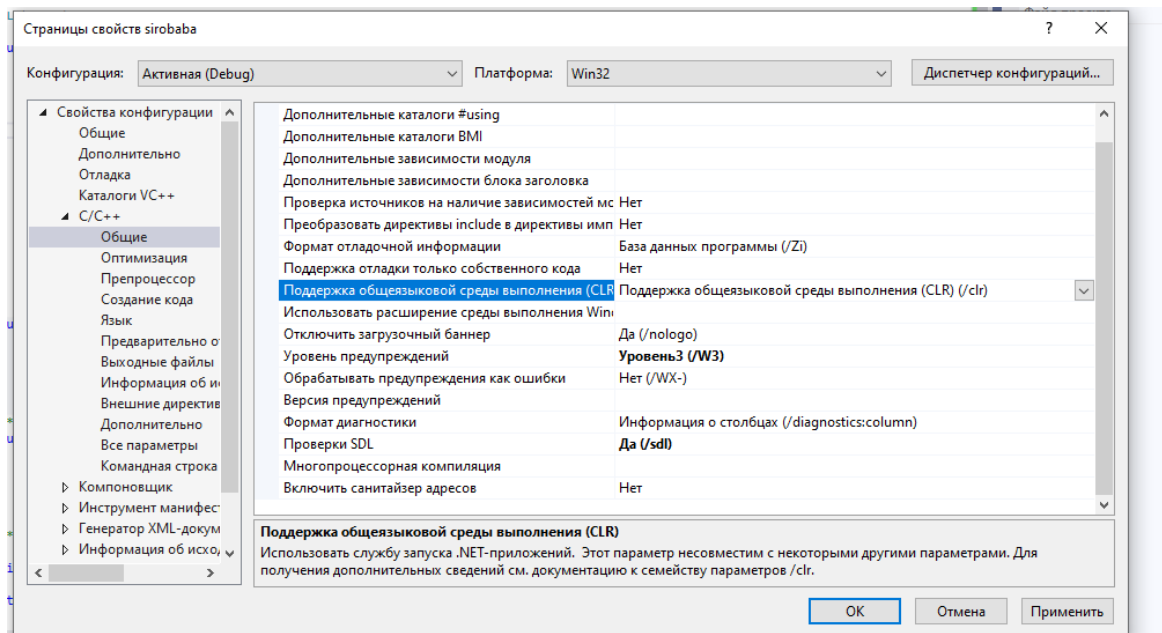


Рисунок 2.2 – Увімкнення CLR

Після цього потрібно підключити додаткові параметри для командного рядку, для цього потрібно перейти по шляху «Проект»-«Свойства:sirobaba»-«C/C++»-«Командная строка», у полі «Дополнительные параметры» ввести «/Zc:twoPhase-» та натиснути кнопку «Применить» та кнопку «ОК». Підключення додаткових параметрів для командного рядка зображено на рис.3.2.

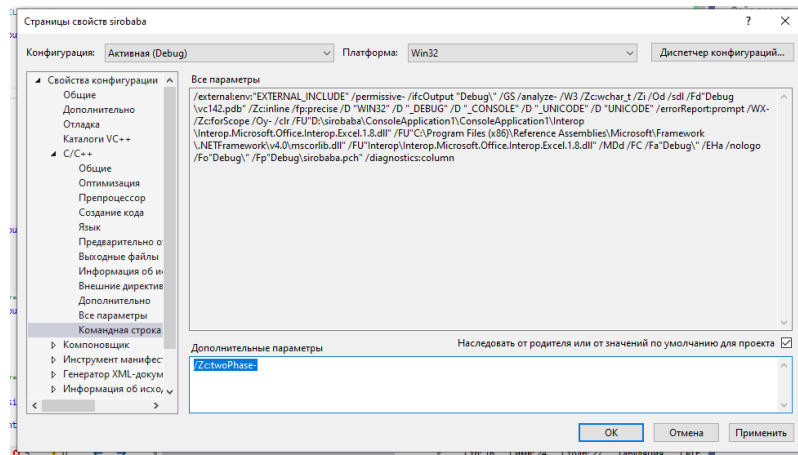


Рисунок 2.3— Підключення додаткових параметрів для командного рядка

Для того, щоб робити з програмою Microsoft Excel потрібно додати необхідні посилання в проект, для цього потрібно натиснути правою кнопкою миші на рядок «Ссылки» та з списку вибрати «Добавить ссылку», після чого у розділі «COM» знайти бібліотеку «Microsoft Excel 15.0 Object Library» та додати її до проекту, натиснувши кнопку «ОК». Підключення бібліотеки «Microsoft Excel 15.0 Object Library» зображено на рис.3.3.

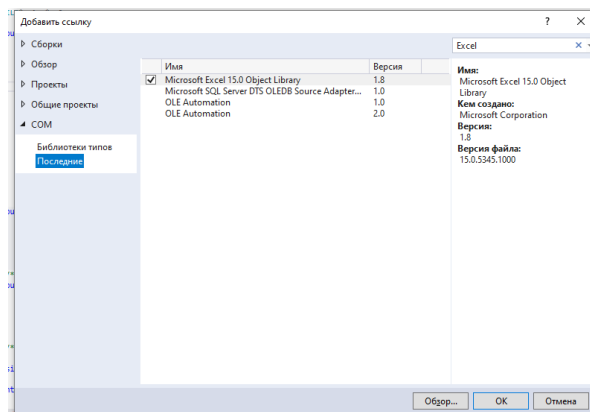


Рисунок 2.4 – Підключення бібліотеки «Microsoft Excel 15.0 Object Library»

2.8 Інструкція для користувача

Програма sirobaba.cpp написана мовою C++. Передбачено робочий та тестовий режими роботи програми. Для вибору робочого режиму треба на запитання комп'ютера відповісти «yes» або «Yes». В цьому випадку дані водяться в режимі діалогу.

Результати виводяться в файли «siro.txt» та «siroexl.txt», також дані виводяться в файл «Test.xlsx», по яким в Microsoft Excel виконується побудова графіка залежності.

2.9 Контрольні приклади

Дослідження провести для кількох варіантів, які враховують конструктивні особливості теплообмінника:

Варіант 1. $P12 = 58$ ккал/хв*град, $P11 = 0.188$, $C1 = 12$ ккал/град, $C2 = 221$ ккал/град.

Варіант 2. $P12 = 52.7$ ккал/хв*град, $P11 = 0.183$, $C1 = 221$ ккал/град, $C2 = 221$ ккал/град.

Варіант 3. $P12 = 52.7$ ккал/хв*град, $P11 = 0.183$, $C1 = 221$ ккал/град, $C2 = 12$ ккал/град.

Протестуємо повний цикл роботи програми: з даними за замовчуванням та з даними, які запропоновані у другому та третьому варіантах.

Для тестування системи запускаємо проект на виконання в середовищі розробки. Після запуску відкривається програма з пропозицією ввести свої дані або використати дані за замовчуванням. Також відкривається програма Microsoft Excel, оскільки розрахунки заносяться в тому числі в таблиці Microsoft Excel і в подальшому за цими даними програма побудує графік залежності. Вікно програми після запуску зображено на рис.2.5.

```
If you do not want to use a test variant, enter Yes or yes
Else enter any another symbol
5
Test Complete. Press any key to exit
Finish!
```

Рисунок 2.5 – Вікно програми після запуску

На рисунку 2.6 зображено результат тестування для першого варіанту даних. На рисунку 2.7 зображена частина даних, які з'являються внаслідок проведення розрахунків для даних з першого варіанту.

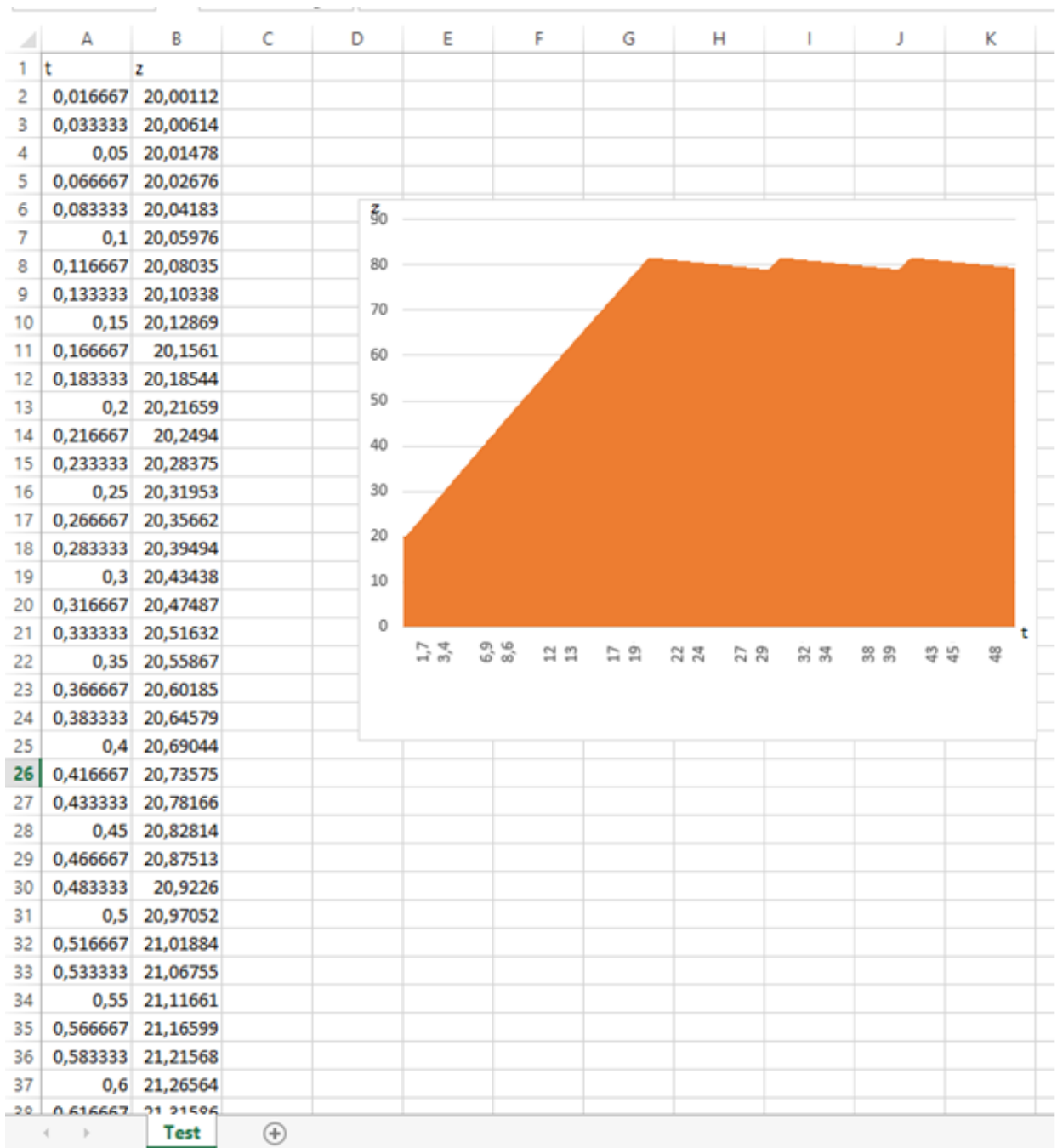


Рисунок 2.6 – Графік залежності температуриреагуючих речовин в реакторі від часу для першого варіанту даних

t	y	z
0.0	20	20
0.0	21.02	20
0.0	21.97	20.01
0.1	22.85	20.01
0.1	23.68	20.03
0.1	24.44	20.04
0.1	25.15	20.06
0.1	25.81	20.08
0.1	26.43	20.1
0.1	27	20.13
0.2	27.54	20.16
0.2	28.04	20.19
0.2	28.51	20.22
0.2	28.94	20.25
0.2	29.35	20.28
0.2	29.73	20.32
0.3	30.09	20.36
0.3	30.42	20.39
0.3	30.73	20.43
0.3	31.03	20.47
0.3	31.3	20.52
0.3	31.56	20.56
0.4	31.81	20.6
0.4	32.04	20.65
0.4	32.25	20.69
0.4	32.46	20.74
0.4	32.65	20.78
0.4	32.83	20.83
0.5	33.01	20.88
0.5	33.17	20.92
0.5	33.33	20.97
0.5	33.47	21.02
0.5	33.62	21.07
0.6	33.75	21.12
0.6	33.88	21.17
0.6	34	21.22
0.6	34.12	21.27
0.6	34.23	21.32
0.6	34.34	21.37
0.7	34.45	21.42
0.7	34.55	21.47
0.7	34.64	21.52
0.7	34.74	21.57
0.7	34.83	21.62

Рисунок 2.7 - Результат розрахунків для першого варіанту даних

На рисунку 2.8 зображено введення даних для другого варіанту тестування. На рисунку 2.9 зображено графік залежності для другого варіанту даних. На рисунку 2.10 зображена частина даних, які з'являються внаслідок проведення розрахунків для даних з другого варіанту.

```

If you do not want to use a test variant, enter Yes or yes
Else enter any another symbol
yes
Enter Q1 kkal/minute*grad
766.7
Enter P11 kkal/minute*grad
0.183
Enter P12 kkal/minute*grad
52.7
Enter P21 kkal/minute*grad
0.62
Enter C1 kkal/grad
221
Enter C2 kkal/grad
221
Enter Tn2 grad
80
Enter dT grad
1
Enter t0 minute
0
Enter tk minute
50
Enter h minute
0.0166666
Enter y0 grad
20
Enter z0 grad
20
Test Complete. Press any key to exit
Finish!

```

Рисунок 2.8 – Введення даних для другого варіанту

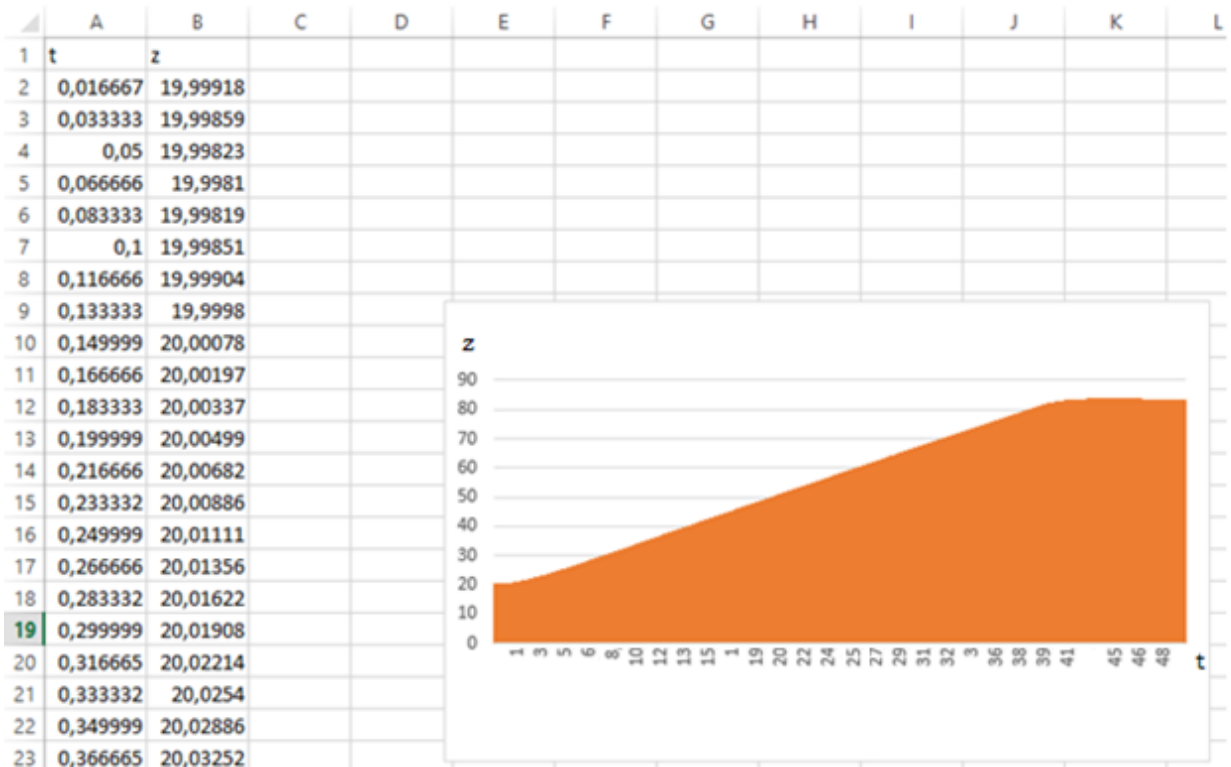


Рисунок 2.9 – – Графік залежності температуриреагуючих речовин в реакторі від часу для другого варіанту даних

t	y	z
0.0	20	20
0.0	20.06	20
0.0	20.11	20
0.0	20.16	20
0.1	20.22	20
0.1	20.27	20
0.1	20.33	20
0.1	20.38	20
0.1	20.44	20
0.1	20.49	20
0.2	20.54	20
0.2	20.6	20
0.2	20.65	20
0.2	20.7	20.01
0.2	20.75	20.01
0.2	20.81	20.01
0.3	20.86	20.01
0.3	20.91	20.01
0.3	20.96	20.02
0.3	21.01	20.02
0.3	21.06	20.02
0.3	21.12	20.03
0.4	21.17	20.03
0.4	21.22	20.03
0.4	21.27	20.04
0.4	21.32	20.04
0.4	21.37	20.04
0.4	21.42	20.05
0.4	21.47	20.05
0.5	21.52	20.06
0.5	21.57	20.06
0.5	21.62	20.07
0.5	21.67	20.07
0.5	21.72	20.08
0.5	21.76	20.08
0.6	21.81	20.09
0.6	21.86	20.09

Рисунок 2.10 - Результат розрахунків для другого варіанту даних

На рисунку 2.11 зображено введення даних для другого варіанту тестування. На рисунку 2.12 зображено графік залежності для другого варіанту даних. На рисунку 2.13 зображена частина даних, які з'являються внаслідок проведення розрахунків для даних з другого варіанту.

```

If you do not want to use a test variant, enter Yes or yes
Else enter any another symbol
yes
Enter Q1 kkal/minute*grad
766.7
Enter P11 kkal/minute*grad
0.183
Enter P12 kkal/minute*grad
52.7
Enter P21 kkal/minute*grad
0.62
Enter C1 kkal/grad
221
Enter C2 kkal/grad
12
Enter Tn2 grad
80
Enter dT grad
1
Enter t0 minute
0
Enter tk minute
50
Enter h minute
0.01666
Enter y0 grad
20
Enter z0 grad
20
Test Complete. Press any key to exit
Finish!

```

Рисунок 2.11 – Введення даних для третього варіанту

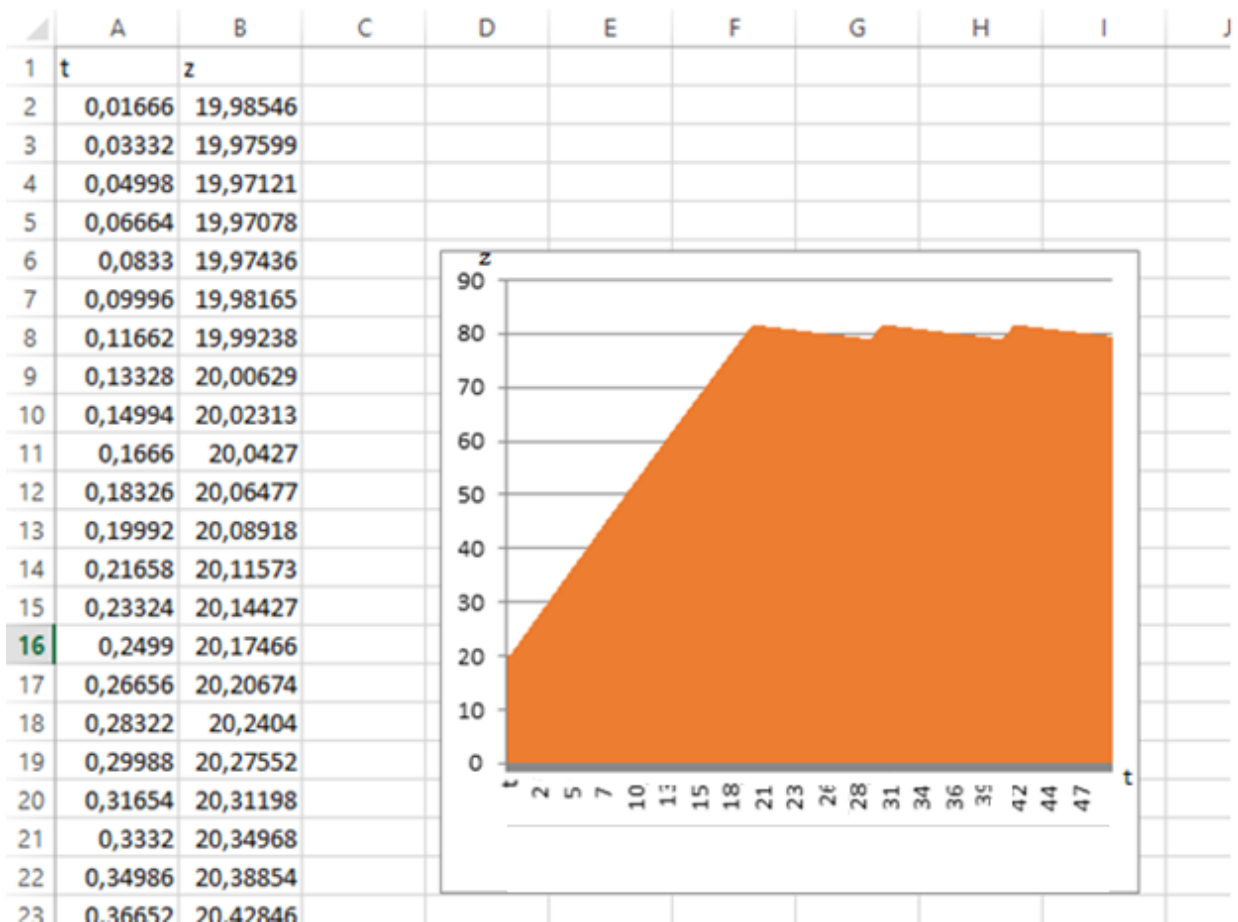


Рисунок 2.12 – Графік залежності температуриреагуючих речовин в раекторі від часу для третього варіанту даних

t	y	z
0.0	20	20
0.0	20.06	19.99
0.0	20.11	19.98
0.0	20.16	19.97
0.1	20.22	19.97
0.1	20.27	19.97
0.1	20.33	19.98
0.1	20.38	19.99
0.1	20.43	20
0.1	20.49	20.02
0.2	20.54	20.03
0.2	20.59	20.05
0.2	20.65	20.08
0.2	20.7	20.1
0.2	20.75	20.13
0.2	20.81	20.16
0.3	20.86	20.19
0.3	20.91	20.22
0.3	20.96	20.25
0.3	21.02	20.28
0.3	21.07	20.32
0.3	21.12	20.36
0.4	21.17	20.39
0.4	21.23	20.43
0.4	21.28	20.47
0.4	21.33	20.51
0.4	21.38	20.55
0.4	21.43	20.59
0.4	21.49	20.64
0.5	21.54	20.68
0.5	21.59	20.72
0.5	21.64	20.77
0.5	21.69	20.81
0.5	21.75	20.86
0.5	21.8	20.9
0.6	21.85	20.95
0.6	21.9	20.99
0.6	21.95	21.04
0.6	22	21.09
0.6	22.06	21.13
0.6	22.11	21.18
0.7	22.16	21.23
0.7	22.21	21.28
0.7	22.26	21.33
0.7	22.31	21.37
0.7	22.37	21.42
0.7	22.42	21.47
0.8	22.47	21.52
0.8	22.52	21.57
0.8	22.57	21.62
0.8	22.62	21.67
--	--	--

Рисунок 2.13 - Результат розрахунків для третього варіанту даних

Аналіз отриманих результатів свідчить, що алгоритм і комп'ютерна програма працюють правильно.

Провівши порівняння між трьома заданими варіантами можна дійти висновку, що найкраще підтримує температуру перший варіант, оскільки в даному випадку спостерігається найменше відхилення температури від заданої норми.

2.10 Висновки

В ході розробки другого розділу було сформульовано ряд вимог, яким повинна відповідати програма.

Було розроблено структуру інтерфейсу програми. Інтерфейс розроблено в мінімалістичному стилі, щоб не перевантажувати користувача надлишковою інформацією. В даному розділі також було визначено основні вікна програми, елементи керування, що розташовуються на них. Також було розроблено блок-схему основних алгоритмів програми.

Модульна структура програми дозволяє додавати нові функціональні можливості, не вносячи глобальних змін в поточну структуру програми.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено програму для моделювання роботи двопозиційного регулятора температури в теплообмінному апараті. Розроблене програмне забезпечення має практичне значення в моделюванні роботи двопозиційного регулятора температури в теплообмінному апараті.

В ході розробки було досліджено сучасний стан програмного забезпечення для моделювання фізичних об'єктів та процесів. До основних переваг програми відноситься зручний інтерфейс та простота роботи.

Проведено постановку задач дослідження та аналіз вимог до програмного забезпечення. Розроблено блок-схеми алгоритмів роботи програми.

Безпосередній розробці програми передувало визначення засобів її створення. Було обрано мову програмування C++ та середовище програмування Microsoft Visual Studio 2019.

Тестування програми показало повну її працездатність та відповідність поставленому технічному завданню. Розроблено інструкцію користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бирюков Б. В., Гастеев Ю. А., Геллер Е. С. Моделирование. М.: БСЭ, 1994
2. Базыкин А.Д. – Математическая биофизика взаимодействующих популяций. - М.: Наука, 2005
3. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. – Практическое моделирование динамических систем – Спб, БХВ-Петербург, 2002.
4. Курс - Математическое моделирование Информатика и информационные технологии в педагогическом образовании. № 2. - Омск: РЦ НИ-ТО, 2006. - С29-34.
5. В. С. Анищенко - Динамические системы – Образовательный журнал СГУ им. Н.Г. Чернышевского, 2003
6. Асанов А. З. - Введение в математическое моделирование динамических систем - Казань: издательство Казанского государственного университета, 2007 – 205 с.
7. Концепция информатизации образования - Информатика и образование. - 2010. - № 1.
8. Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре – (Классика Computer Science). 2004.
9. Тестирование программногo обеспечения. [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Тестирование_программного_обеспечения. – Назва з екрану.
10. Искусство тестирования программ, 3-е издание [Электронный ресурс] – Режим доступа: <http://www.dialektika.com/books/978-5-8459-1796-6.html>. – Назва з екрану.
11. Калбертсон Роберт Быстрое тестирование. / Роберт Калбертсон, Крис Браун, Гэри Кобб. – М.:«Вильямс», 2002. – 374 с..

ДОДАТКИ

Додаток А

Код файлу sirobaba.cpp

```
#include <iostream>
#include<stdio.h>
#include<math.h>
#include <string>

double P11 = 0.183, P12 = 52.7, P21 = 0.62, C1 = 12, C2 = 221, Q1 = 766.7, Tn2 = 80, dT =
1;
double t0 = 0, h = 1. / 60, tk = 50;
FILE* p1, * p2;

double QQ(double t, double z, double zpaper, double zpotoch)
{
    double r1, r2, q, dz;
    r1 = Tn2 - dT;
    r2 = Tn2 + dT;
    dz = zpotoch - zpaper;

    if ((z < r1 && dz < 0) || (z < r2 && dz >= 0))
        q = Q1;
    else
        if ((z > r2 && dz > 0) || (z > r1 && dz <= 0))
            q = 0;
        else
            q = 1e6;
    return q;
}
double f1(double t, double y, double z, double q)
{
    double v;
    v = (q - P11 * y - P12 * (y - z)) / C1;
    return v;
}

double f2(double t, double y, double z)
{
    double v;
    v = (P12 * (y - z) - P21 * z) / C2;
    return v;
}

using namespace Microsoft::Office::Interop::Excel;

int main()
{
    Microsoft::Office::Interop::Excel::Application^ app
        = gcnew Microsoft::Office::Interop::Excel::ApplicationClass();
    app->Visible = true;
    Microsoft::Office::Interop::Excel::Workbook^ workbook = app->Workbooks-
>Add(System::Type::Missing);
    Microsoft::Office::Interop::Excel::Worksheet^ worksheet
        = safe_cast<Microsoft::Office::Interop::Excel::Worksheet^>(app-
>ActiveSheet);
    worksheet->Name = "Test";

    double t;
    double k1, k2, k3, k4, m1, m2, m3, m4;
    double y, y0 = 20, z, z0 = 20;
```

```

double qq, zold, znew;
char vubir[4];
fopen_s(&p1, "siro.txt", "w");
fopen_s(&p2, "siroexl.txt", "w");
puts("If you do not want to use a test variant, enter Yes or yes");
puts("Else enter any another symbol");
fflush(stdin);
scanf_s("%s", &vubir, 4);
if (strcmp(vubir, "Yes") == 0 || strcmp(vubir, "yes") == 0)
{
    puts(" Enter Q1 kkal/minute*grad ");
    scanf_s("%lf", &Q1);
    puts(" Enter P11 kkal/minute*grad ");
    scanf_s("%lf", &P11);
    puts(" Enter P12 kkal/minute*grad ");
    scanf_s("%lf", &P12);
    puts(" Enter P21 kkal/minute*grad ");
    scanf_s("%lf", &P21);
    puts(" Enter C1 kkal/grad ");
    scanf_s("%lf", &C1);
    puts(" Enter C2 kkal/grad ");
    scanf_s("%lf", &C2);

    puts(" Enter Tn2 grad ");
    scanf_s("%lf", &Tn2);
    puts(" Enter dT grad ");
    scanf_s("%lf", &dT);
    puts(" Enter t0 minute ");
    scanf_s("%lf", &t0);
    puts(" Enter tk minute ");
    scanf_s("%lf", &tk);
    puts(" Enter h minute ");
    scanf_s("%lf", &h);
    puts(" Enter y0 grad ");
    scanf_s("%lf", &y0);
    puts(" Enter z0 grad ");
    scanf_s("%lf", &z0);
}

t = t0;
y = y0;
z = z0;
zold = z0;
znew = z0;
fprintf(p1, "t=%0.11f y=%0.4lg z=%0.4lg \n", t, y, z);
fprintf(p2, "%0.11f %0.4lg %0.4lg \n", t, y, z);

worksheet->Cells[1, "A"] = "t";
worksheet->Cells[1, "B"] = "z";

int r = 2;
while (t <= tk)
{
    qq = QQ(t, z, zold, znew);
    k1 = f1(t, y, z, qq);
    m1 = f2(t, y, z);
    k2 = f1(t + h / 2, y + h * k1 / 2, z + h * m1 / 2, qq);
    m2 = f2(t + h / 2, y + h * k1 / 2, z + h * m1 / 2);
    k3 = f1(t + h / 2, y + h * k2 / 2, z + h * m2 / 2, qq);
    m3 = f2(t + h / 2, y + h * k2 / 2, z + h * m2 / 2);
    k4 = f1(t + h, y + h * k3, z + h * m3, qq);
    m4 = f2(t + h, y + h * k3, z + h * m3);

    y += h * (k1 + 2 * k2 + 2 * k3 + k4) / 6;
    z += h * (m1 + 2 * m2 + 2 * m3 + m4) / 6;
}

```

```

        t += h;

        worksheet->Cells[r, "A"] = t;
        worksheet->Cells[r, "B"] = z;

        r++;

        fprintf(p1, "t=%0.11f y=%0.4lg z=%0.4lg qq=%0.4lg\n", t, y, z, qq);
        fprintf(p2, "%0.11f      %0.4lg      %0.4lg      %0.4lg\n", t, y, z, qq);

        zold = znew;
        znew = z;
    }

    Microsoft::Office::Interop::Excel::Range^ chartRange;
    Microsoft::Office::Interop::Excel::ChartObjects^ xlCharts =
safe_cast<ChartObjects^>(worksheet->ChartObjects(System::Type::Missing));
    Microsoft::Office::Interop::Excel::ChartObject^ myChart = xlCharts->Add(150, 80,
300, 250);

    Microsoft::Office::Interop::Excel::Chart^ chartPage = myChart->Chart;
    chartRange = worksheet->Range["A1", "B3001"];

    chartPage->SetSourceData(chartRange, System::Reflection::Missing::Value);
    chartPage->ShowDataLabelsOverMaximum = true;
    safe_cast<Series^>(chartPage->SeriesCollection(1))->Border->Weight = \
        XlBorderWeight::xlThick;
    safe_cast<Series^>(chartPage->SeriesCollection(2))->Border->Weight = \
        XlBorderWeight::xlThick;

    try
    {
        workbook->SaveAs(System::Environment::CurrentDirectory + "\\Test.xlsx",
            System::Type::Missing, System::Type::Missing, System::Type::Missing,
            false, false,
            Microsoft::Office::Interop::Excel::XlSaveAsAccessMode::xlShared,
            false, false,
            System::Type::Missing, System::Type::Missing, System::Type::Missing
        );

        workbook->Close(false, System::Type::Missing, System::Type::Missing);

        System::Runtime::InteropServices::Marshal::ReleaseComObject(workbook);

        app->Quit();

        System::Runtime::InteropServices::Marshal::ReleaseComObject(app);
    }
    catch(...)
    {}

    System::Console::WriteLine("Test Complete. Press any key to exit");
    puts("Finish!");
    System::Console::ReadKey();
}

```