МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДНІПРОПЕТРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ
ІМЕНІ АКАДЕМІКА В. ЛАЗАРЯНА

ЖОВТОНОГА М.М., БУЛГАКОВ Д.О., ПЕРЦЕВИЙ В.О.

ВИКОРИСТАННЯ ПРОГРАМНИХ ПРОДУКТІВ COMSOL MULTIPHYSICS® ТА COMSOL® SCRIPT™ ПРИ РОЗВ’ЯЗАННІ ЗАДАЧ ОПТИМІЗАЦІЇ ПРОМИСЛОВОЇ ТЕПЛОЕНЕРГЕТИКИ

ДНІПРО
2018

УДК 621.186.1:004.4
Ж 78

Жовтонога М. М.

Ж 78 Використання програмних продуктів COMSOL MULTIPHYSICS® ТА COMSOL® SCRIPT™ при розв'язанні задач оптимізації промислової теплоенергетики: навчальний посібник / М.М. Жовтонога, Д.О. Булгаков, В.О. Перцевий. – Д. : ЛІРА, 2018. – 116 с.
ISBN 978-966-383-976-9

Книга присвячена застосуванню математичних пакетів COMSOL MULTIPHYSICS®, COMSOL® SCRIPT™ та MATLAB® для розв'язання лінійних та нелінійних оптимізаційних задач промислової теплоенергетики. Докладно розглянуті питання коректного синтаксису задач безумовної оптимізації та задач оптимізації при наявності обмежень.

Приділено увагу розв'язанню задач оптимізації на прикладі теплотехнічних процесів, які описуються лінійними та нелінійними алгебраїчними рівняннями, а також диференціальними рівняннями у частинних похідних.

УДК 621.186.1:004.4

ISBN 978-966-383-976-9

ЗМІСТ

ВСТУП

Оптимізація - це процес отримання найкращих результатів в даних умовах. З математичної точки зору задача оптимізації теплотехнічних та теплотехнологічних апаратів, пристроїв, процесів полягає у відшукуванні екстремуму певного критерію ефективності функціонування теплотехнічних та теплотехнологічних апаратів, пристроїв, процесів при наявності низки обмежень на їхні технологічні та конструкційні параметри.

Зокрема, задачею оптимізації теплообмінних систем є пошук екстремуму критерію ефективності або оптимізації (технологічного, термодинамічного, економічного тощо) при заданій структурі технологічних зв'язків між теплообмінними апаратами з урахуванням обмежень, що стосуються вимог технологічного завдання на проектування теплообмінного апарату або технологічних умов його функціонування.

Зазвичай, змінними, що варіюються при оптимізації теплообмінних апаратів та систем, є масова витрата холодного теплоносія, кінцеві температури теплоносіїв, кількість теплоти, що передається в теплообмінному апараті.

Для оцінювання ефективності теплообмінного апарату можуть бути застосовані критерії ефективності різних типів:

1) технологічні:

а) припустима кінцева температура потоку одного з теплоносіїв;

б) швидкість руху потоків теплоносіїв в теплообмінному апараті тощо;

2) термодинамічні:

а) ККД тощо;

3) економічні:

а) зведені витрати на створення та функціонування теплообмінного апарату тощо.

1 ПОРІВНЯННЯ РЕЗУЛЬТАТІВ РОЗВ’ЯЗКІВ ЗАДАЧ ОПТИМІЗАЦІЇ В РІЗНИХ ВЕРСІЯХ MATLAB® OPTIMIZATION TOOLBOX™

В таблиці 1.1 та таблиці 1.2 представлено порівняння розв’язків задач оптимізації, наведених в роботі Ануфриева [1], та розв’язків цих задач, виконаних в усучасненій версії MATLAB® Optimization Toolbox™.

Текст програми для розв’язання задачі оптимізації, розв’язки яких наведено в таблиці 1.1, має наступний вигляд:

```
clc
clear
A = [4 5 15; 2 2 0; 5 3 4; 7 3 12];
A = -A;
b = [250; 60; 100; 220];
b = -b;
f = [44; 35; 100];
lb = [0; 0; 0];
[x, p] = linprog (f, A, b, [], [], lb);
```

Таблиця 1.1- Порівняння результатів розв’язку задачі в різних версіях MATLAB® Optimization Toolbox™

	x_1	x_2	x_3	y
За результатами розрахунку [1]	13,2143	16,7857	6,4286	1811,8
За результатами розрахунку в усучасненій версії MATLAB® Optimization Toolbox™	10,4167	19,5833	7,3611	1879,9

Текст програми для розв’язання задачі оптимізації, розв’язки яких наведено в таблиці 1.2, має наступний вигляд:

```
clc
clear
V = [102.0 27.1 -52.3 66.5;
      27.1 148.8 42.1 -66.4;
```

```
-52.3 42.1 246.5 56.9;
 66.5 -66.4 56.9 272.3];
Aeq = [11.3 13.2 16.1 17.4; 1 1 1 1];
beq = [15; 1];
lb = [0; 0; 0; 0];
x = quadprog (V, [], [], [], Aeq, beq, lb);
```

Таблиця 1.2- Порівняння результатів розв'язку задачі в різних версіях MATLAB® Optimization Toolbox™

	x_1	x_2	x_3	x_4
За результатами розрахунку [1]	0,0626	0,4359	0,1439	0,3575
За результатами розрахунку в усучасненій версії MATLAB® Optimization Toolbox™	0,0626	0,4359	0,1439	0,3575

В таблиці 1.3 представлено порівняння розв'язків задачі оптимізації, наведеної в роботі Дьяконова [2], та розв'язку цієї задач, виконаної в усучасненій версії MATLAB® Optimization Toolbox™.

Таблиця 1.3- Порівняння результатів розв'язку задачі в різних версіях MATLAB® Optimization Toolbox™

	x_1	x_2	x_3	x_4
За результатами розрахунку [2]	0,0626	0,4359	0,1439	0,3575
За результатами розрахунку в усучасненій версії MATLAB® Optimization Toolbox™	0,0626	0,4359	0,1439	0,3575

Текст програми для розв'язання задачі оптимізації, розв'язки яких наведено в таблиці 1.3, має наступний вигляд:

```
clc
clear
f = [-5; -4; -6];
A = [1 -1 1; 3 2 4; 3 2 0];
b = [20; 42; 30];
```

lb = [3; 1];

[x, fval, exitflag, output, lambda] = linprog (f, A, b, [], [], lb);

Як видно з таблиць 1.1-1.3, результати розрахунку задач оптимізації, виконаних в різних версіях програмного продукту MATLAB® Optimization Toolbox™ мають певні відмінності, що свідчить про недосконалість цього програмного продукту.

Серед відомих спеціалізованих програмних продуктів для розв'язанні задач оптимізації, найкраще себе зарекомендував модуль COMSOL® Optimization Lab програмного продукту .

Засади розв'язання задач оптимізації в цьому програмному продукті викладено в наступних розділах навчального посібника.

2 ПРОГРАМУВАННЯ ЗАДАЧ БЕЗУМОВНОЇ ОПТИМІЗАЦІЇ

2.1 Загальні теоретичні положення безумовної оптимізації

Метод Нелдера - Міда для знаходження абсолютного мінімуму функції є розвитком симплексного методу Спендлі, Хекста та Хімсворта. Цей метод є одним з найефективніших методів безумовної оптимізації, однак тільки для тих випадків, коли кількість незалежних аргументів функції не перевищує шести, тобто коли: $y = f(x_1, x_2, x_3, x_4, x_5, x_6)$.

Розглянемо сутність методу Нелдера - Міда на прикладі відшукування абсолютного мінімуму функції двох незалежних аргументів, тобто функції виду:

$$y = f(x_1, x_2) \tag{2.1}$$

Для функції двох незалежних аргументів симплексом є трикутник.

Увага ! Трикутник - це найпростіша двовимірна фігура, оскільки у нього три вершини. Фігурою з двома вершинами є пряма, однак пряма - це одновимірна фігура. З іншої сторони чотирикутник також представляє собою двовимірну фігуру, однак, вона більш ”складна” в порівнянні з трикутником, оскільки у чотирикутника чотири вершини.

Метод Нелдера - Міда - це схема пошуку, яка порівнює значення функції $y = f(x_1, x_2)$ в трьох вершинах трикутника. Найгірша вершина, тобто вершина, в якій функція $y = f(x_1, x_2)$ приймає найбільше значення (нагадаємо, що необхідно знайти абсолютний мінімум функції), відкидається та заміняється новою вершиною. Формується новий трикутник після чого пошук продовжується. При цьому будується послідовність трикутників, значення функції в вершинах якої зменшується. Зменшується розмір трикутника і таким чином знаходяться координати мінімуму функції, тобто значення $x_{1\min}$ та $x_{2\min}$, при яких функція **y**

приймає найменше значення y_{min} (див. рисунок 2.1).

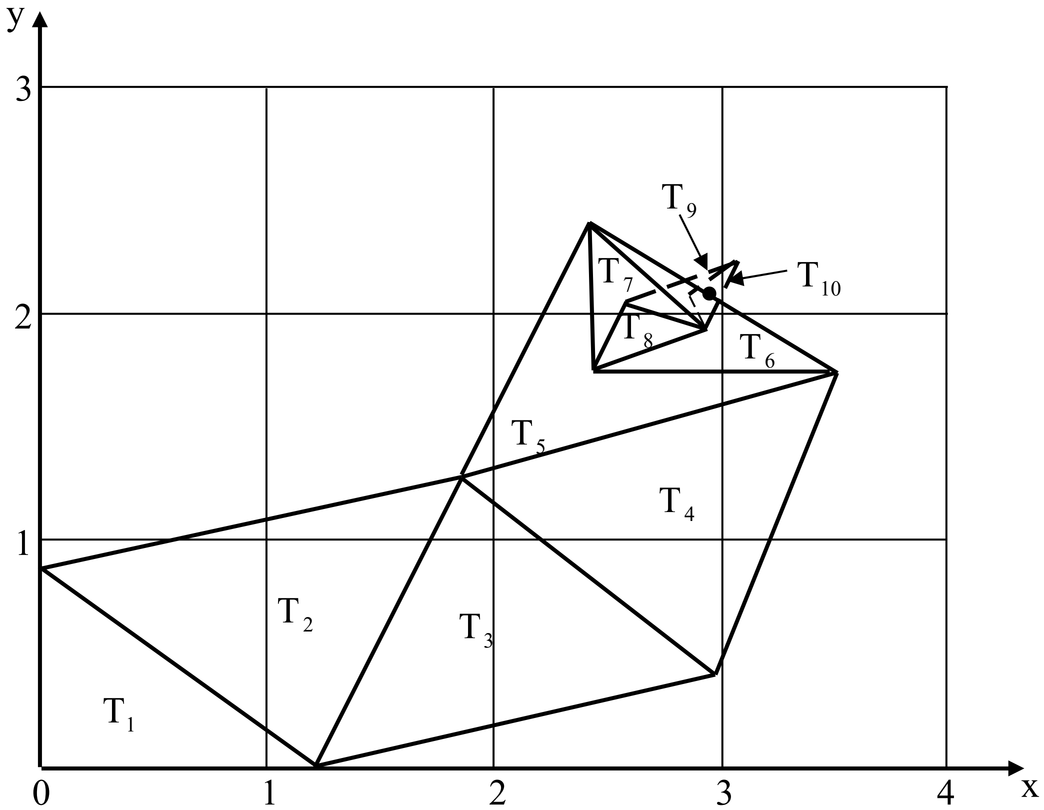

Рисунок 2.1 - Послідовність трикутників ($T_1, T_2, \dots T_{10}$), що сходиться до точки мінімуму функції для методу Нелдера - Міда

2.2 Приклади розв'язання задач безумовної оптимізації

Приклад 2.1.

Потрібно знайти абсолютний мінімум функції $y = -\sin(x)$.

Для розв'язання задачі в програмному продукті COMSOL MULTIPHYSICS® потрібно виконати наступні кроки:

1. Відкрити COMSOL® SCRIPT™.

2. Створити m-файл. Для цього необхідно зайти в пункт меню **File** > **New editor.** В полі вікна COMSOL Desktop® відкривається порожній m-файл.

Потрібно зберегти файл, попередньо присвоївши йому ім'я. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім'я файлу, наприклад, **optymizatsiya_function_1**.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

3. Написати першим рядком робочого поля m-файлу вбудовану команду **clear**, за допомогою якої при кожному запуску програми видаляються всі змінні з робочого поля (**Workspace)**, що призводить до звільнення ресурсів (оперативної пам'яті, навантаження на процесор тощо) ЕОМ.

4. Написати наступним рядком робочого поля m-файлу вбудовану команду **clc**, за допомогою якої видаляються всі команди з командного вікна (**Command Prompt)** при кожному запуску програми, що міститься в m-файлі.

5. Наступними двома рядками задати масиви значень незалежного аргументу **x** та залежної від нього функції **y** для побудови та відображення графіку функції, мінімізація якої розглядається в даному прикладі:

x_plot = [-4:0.01:4];

y_plot = -sin(x_plot);

Увага ! Можна відмовитись від побудови та відображення графіку функції, оскільки цей крок при написанні програми потрібний в подальшому лише для візуалізації точки на графіку, якій відповідає абсолютний мінімум функції. В такому випадку пунктами 5…8 потрібно знехтувати.

Для покращення наочного сприйняття рядків програми знак "=" можна відокремити пробілами від лівої та правої частини наведених вище виразів. В такий самий спосіб можна відокремити пробілами число "-4" від знаку "**:**", знак "**:**" від числа "0.01" тощо. Також можна відділити пробілом вираз функції "-sin" від імені масиву значень незалежного аргументу **(x_plot)**.

За синтаксисом першого з наведених вище двох рядків програми, після імені масиву **x_plot** та знаку "=" у квадратних дужках записуються параметри масиву, які розділяються двокрапкою. Порядок цих параметрів наступний:

1) Перше число в квадратних дужках – це початкове значення масиву;

2) Друге число в квадратних дужках – це крок значень масиву;

3) Третє число в квадратних дужках – це кінцеве значення масиву.

В даному прикладі в якості початкового значення масиву незалежного аргументу **x** обрано число “-4”, кінцевим значенням масиву **x** є число “4”. Крок значень масиву **x** в межах між початковим та кінцевим значенням дорівнює **0.01**.

Увага ! Вибір початкового та кінцевого значення масиву значень незалежного аргументу **x** є довільним, але варто мати на увазі, що період функції $y = -\sin(x)$ дорівнює $2 \cdot \pi$. Це означає, що абсциса абсолютного мінімуму функції $y = -\sin(x)$ буде знаходитись в межах $-\pi \le x \le \pi$, тобто в межах $-3{,}14 \le x \le 3{,}14$.

Завершити рядок програми рекомендується крапкою з комою, що дозволить запобігти відображенню цього рядка програми в командному вікні (**Command Prompt)** під час її виконання. Таким чином усувається проблема “захаращення” командного вікна (**Command Prompt)** в разі багаторазового виконання програми, наприклад, у випадках її налагодження або варіювання певними параметрами програми (відшукування абсолютного максимуму замість абсолютного мінімуму функції, внесення змін до виразу самої функції тощо).

За синтаксисом другого з наведених вище двох рядків програми, після імені масиву **y_plot** та знаку “**=**” записується вираз функції, абсолютний мінімум якої потрібно відшукати. Після виразу функції, тобто **-sin** в дужках записується ім’я масиву значень незалежного аргументу **x**, **(x_plot)**.

Після виконання цього рядка програми у відповідності до масиву значень незалежного аргументу **x**, який має ім’я **x_plot**, обчислюється масив залежної від нього функції **y**, якому присвоюється ім’я **y_plot**.

6. Написати команду відображення графіку функції по заданим масивам незалежного аргументу **x** та залежної від нього функції **y**:

plot(x_plot, y_plot);

Команда **plot**, за допомогою якої здійснюється відображення графіку функції має наступний синтаксис: після слова **plot** у дужках записуються імена масиву значень абсцис та масиву значень ординат функції (масиву

значень незалежного аргументу **x** та масиву значень залежної від нього функції **y**), за значеннями яких будується графік функції.

Увага ! В дужках після слова **plot** першим записується ім’я масиву, значення якого відповідають значенням осі абсцис, другим записується ім’я масиву, значення якого відповідають значенням осі ординат.

Після виконання наведеного вище рядка програми на панелі операційної системи ЕОМ в згорнутому вигляді з’явиться файл, який матиме назву **Figure 1**. В полі файлу **Figure 1** відображається графік функції $y = -\sin(x)$ в межах масиву значень незалежного аргументу **x** та масиву залежної від нього функції **y**.

7. Для відображення ліній сітки осей **x** та **y** на графіку наступним рядком програми потрібно записати:

grid on

Увага ! Можна відмовитись від відображення ліній сітки осей **x** та **y** на графіку за допомогою команди **grid on** (сітка увімкнута), оскільки цей крок при написанні програми потрібний в подальшому лише для покращення наочного сприйняття графіку. В такому випадку пунктом 7 потрібно знехтувати.

---

8. Для відображення в полі файлу, який має назву **Figure 1**, графіку функції та точки на графіку, якій відповідає абсолютний мінімум функції, наступним рядком програми потрібно записати:

hold on

Увага ! Наслідки нехтування командою **hold on** будуть розглянуті в пункті 16 даної задачі.

---

9. Записати в підструктуру з ім’ям **opt.obj.f** функцію, абсолютний мінімум якої потрібно визначити, за допомогою команди **inline** (визначення):

opt.obj.f = inline ('-sin(x)');

Увага ! Вираз функції після слова **inline** записується в дужках та виокремлюється з обох боків одинарними лапками.

В лівій частині виразу, наведеному в пункті 9, записується

підструктура **opt.obj.f**, яка використовується для запису до неї функції, мінімум або максимум якої потрібно знайти (цільової функції).

Як видно з форми запису підструктури **opt.obj.f**, її ім’я складається з наступних складових:

opt- **opt**imization (оптимізація). Ця складова імені записується першою та міститься в іменах всіх підструктур, які будуть зустрічатись в подальшому при розгляді алгоритму розв’язання поточної та наступних задач оптимізації;

obj.f- **obj**ect **f**unction (цільова функція). Друга (**obj**) та третя (**f**) складові імені цієї підструктури вказують на те, що підструктура з ім’ям **opt.obj.f** використовується для запису до неї цільової функції.

Увага ! Складові імені підструктури відокремлюються одне від одного за допомогою крапки. Присвоювати довільні імена складовим підструктур неможна. Тобто, наприклад, першій складовій всіх підструктур потрібно присвоювати тільки ім’я **opt**, оскільки COMSOL® Optimization Lab сприймає та працює лише з підструктурами, які мають певні, зрозумілі для програмного продукту, сталі імена. Те ж саме стосується складових імені підструктури **opt.obj.f** тощо.

---

10. Записати в підструктуру з ім’ям **opt.init.x** довільне значення незалежного аргументу **x**, починаючи з якого COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції та значення незалежного аргументу **x**, який відповідає абсолютному мінімуму функції:

opt.init.x = [0];

Як видно з форми запису підструктури **opt.init.x**, її ім’я складається з наступних складових:

opt- **opt**imization (оптимізація). Ця складова імені, як вже згадувалось раніше, розташовується першою та міститься в іменах всіх підструктур;

init.x- **init**ial **x** (початкове значення **x**). Друга (**init**) та третя (**x**) складові імені цієї підструктури вказують на те, що підструктура з ім’ям **opt.init.x** використовується для запису до неї початкового значення незалежного аргументу **x**.

Увага ! Початкове та довільне значення незалежного аргументу **x** впливає

лише на швидкість відшукання абсолютного мінімуму або максимуму цільової функції і не впливає на точність відшукання мінімуму або максимуму цільової функції.

Початкове значення незалежного аргументу **x** завжди записується в квадратних дужках. В даній задачі в якості початкового та довільного значення незалежного аргументу **x** задається нуль.

---

11. Задати команду, за допомогою якої буде виконуватись пошук абсолютного мінімуму цільової функції:

opt.sol = optnm(opt);

В якості такої команди виступає вбудований в програмний продукт розв'язувач **optnm**. В дужках після імені розв'язувача записується перша складова імені структури, в якій у вигляді підструктур **opt.obj.f** та **opt.init.x** містяться, відповідно, цільова функція та початкове значення незалежного аргументу **x**.

Увага ! Розв'язувач **optnm** для відшукування абсолютного максимуму або абсолютного мінімуму цільової функції використовує симплекс-метод Нелдера - Міда, який описано Банди [3]. Цей розв'язувач можна застосовувати лише для розв'язання задач безумовної оптимізації. У випадку існування обмежень, накладених на цільову функцію, розв'язувач **optnm** використовувати неможна.

В лівій частині виразу, наведеному в пункті 11, записується підструктура **opt.sol**, яка використовується для запису до неї результатів розв'язання задачі.

Як видно з форми запису підструктури **opt.sol**, її ім'я складається з наступних складових:

opt- **opt**imization (оптимізація);

sol- **sol**ution (розв'язання, розв'язок). Друга (**sol**) складова імені цієї підструктури вказує на те, що підструктура з ім'ям **opt.sol** використовується для запису до неї результатів розв'язання задачі.

12. Задати команду, за допомогою якої після завершення процесу розв'язання задачі в командному вікні (**Command Prompt)** можна переглянути налаштування розв'язувача, які встановлено за замовчуванням:

options = optprop

Як видно з форми запису команди **optprop**, її назва складається зі скорочень двох слів:

- **opt**- **opt**imization (оптимізація);

- **prop**- **prop**erties (властивості).

Увага ! Наприкінці наведеного вище рядка програми відсутня крапка з комою, що дозволить відобразити налаштування розв’язувача, які встановлено за замовчуванням, в командному вікні (**Command Prompt)** після завершення процесу розв’язання задачі.

За допомогою команди **optprop** в командному вікні можна відобразити налаштування будь-якого розв’язувача, а не тільки розв’язувача **optnm**, який застосовується в даній задачі.

В лівій частині виразу, наведеному в пункті 12, записується ім’я змінної **options**, яка використовується для запису до неї налаштувань розв’язувача.

Увага ! На відміну від імен підструктур, змінній, яка використовується для запису до неї налаштувань розв’язувача, можна присвоїти довільне ім’я. В даній задачі такій змінній присвоєно ім’я **options**.

---

13. Задати команду добування значення незалежного аргументу **x**, якому відповідає значення абсолютного мінімуму функції, з підструктури **opt.sol**, яка, нагадуємо, використовується для запису до неї результатів розв’язання задачі:

x_min = opt.sol.x;

Значення незалежного аргументу **x**, якому відповідає значення абсолютного мінімуму функції, знаходиться в підструктурі з ім’ям **opt.sol.x**.

Як видно з форми запису підструктури **opt.sol.x**, її ім’я складається з наступних складових:

opt- **opt**imization (оптимізація);

sol.x- **sol**ution **x** (розв’язок **x**). Друга (**sol**) та третя (**x**) складові імені цієї підструктури вказують на те, що підструктура з ім’ям **opt.sol.x**

використовується для запису до неї значення незалежного аргументу **x**, якому відповідає значення абсолютного мінімуму функції.

В лівій частині виразу, наведеному в пункті 13, записується ім'я змінної **x_min**, яка використовується для запису до неї значення незалежного аргументу **x**, якому відповідає значення абсолютного мінімуму функції, після його добування з підструктури **opt.sol.x**.

Увага ! Як вже згадувалось раніше, на відміну від імен підструктур, змінній, яка використовується для запису до неї значення незалежного аргументу **x**, якому відповідає значення абсолютного мінімуму функції, можна присвоїти довільне ім'я. В даній задачі такій змінній присвоєно ім'я **x_min**.

---

14. Аналогічним чином задається команда для добування значення абсолютного мінімуму функції з підструктури **opt.sol**:

f_min = opt.sol.eval.f;

Значення абсолютного мінімуму функції, знаходиться в підструктурі з ім'ям **opt.sol.eval.f**.

Як видно з форми запису підструктури **opt.sol.eval.f**, її ім'я складається з наступних складових:

opt- **opt**imization (оптимізація);

sol- **sol**ution (розв'язок);

eval.f- **eval**uation **f**unction (оцінювання функції). Третя (**eval**) та четверта (**f**) складові імені цієї підструктури вказують на те, що підструктура з ім'ям **opt.sol.eval.f** використовується для запису до неї значення абсолютного мінімуму функції.

В лівій частині виразу, наведеному в пункті 14, записується ім'я змінної **f_min**, яка використовується для запису до неї значення абсолютного мінімуму функції, після його добування з підструктури **opt.sol.eval.f**.

Увага ! Як вже згадувалось раніше, на відміну від імен підструктур, змінній, яка використовується для запису до неї значення абсолютного мінімуму функції, можна присвоїти довільне ім'я. В даній задачі такій змінній присвоєно ім'я **f_min**.

---

15. Для контролю за успішністю розв'язання задачі потрібно додати до програми два цикли **if…end**.

Перший цикл:

if opt.sol.exit == 1;

optymizatsiya = 'uspishna_optymizatsiya';

end

Другий цикл:

if opt.sol.exit ==0;

optymizatsiya = 'nevdala_optymizatsiya';

end

Відомості про успішність перебігу процесу розв'язання задачі COMSOL MULTIPHYSICS® записує до підструктури **opt.sol.exit**.

Як видно з форми запису підструктури **opt.sol.exit**, її ім'я складається з наступних складових:

opt- **opt**imization (оптимізація);

sol- **sol**ution (розв'язок);

exit - **exit** (вихід). Третя (**exit**) складова імені цієї підструктури вказує на те, що підструктура з ім'ям **opt.sol.exit** використовується для запису до неї вихідних, тобто прикінцевих параметрів, за якими можна визначити успішність розв'язання задачі.

Так, в разі успішного розв'язання задачі значення величини, яка записується до підструктури **opt.sol.exit**, дорівнює одиниці.

В першому рядку першого циклу **if…end**:

if opt.sol.exit == 1;

здійснюється порівняння значення величини, яка записана до підструктури **opt.sol.exit**, з одиницею.

Умова першого циклу **if…end** виконується, оскільки, значення величини, яка записується до підструктури **opt.sol.exit**, дорівнює одиниці.

Таким чином:

- перший цикл **if…end** завершується;

- змінній, яка використовується для запису до неї повідомлення про успішність розв’язання задачі присвоюється ім’я **optymizatsiya**;

- до цієї змінної записується повідомлення про успішність розв’язання задачі, наприклад: **'uspishna_optymizatsiya'**;

optymizatsiya = 'uspishna_optymizatsiya';

end

- другий цикл **if…end** не виконується;

- починається виконання наступного, 16-го пункту програми.

Увага ! Як вже згадувалось раніше, на відміну від імен підструктур, змінній, яка використовується для запису до неї повідомлення про успішність розв’язання задачі, можна присвоїти довільне ім’я. В даній задачі такій змінній присвоєно ім’я **optymizatsiya**.

Повідомлення про успішність розв’язання задачі **'uspishna_optymizatsiya'** виокремлюється з обох боків одинарними лапками.

У випадку, коли розв’язання задачі не є успішним значення величини, яка записується до підструктури **opt.sol.exit**, дорівнює нулю.

В першому рядку першого циклу **if…end**:

if opt.sol.exit == 1;

здійснюється порівняння значення величини, яка записана до підструктури **opt.sol.exit**, з одиницею.

Умова першого циклу **if…end** не виконується, оскільки, значення величини, яка записується до підструктури **opt.sol.exit**, дорівнює нулю.

Таким чином:

- перший цикл **if…end** завершується;

- виконується перевірка відповідності значення величини, яка записується до підструктури **opt.sol.exit**, умові другого циклу **if…end**

В першому рядку другого циклу **if…end**:

if opt.sol.exit == 0;

здійснюється порівняння значення величини, яка записана до підструктури **opt.sol.exit**, з нулем.

Умова другого циклу **if…end** виконується, оскільки, значення

величини, яка записується до підструктури **opt.sol.exit**, дорівнює нулю.

Таким чином:

- змінній, яка використовується для запису до неї повідомлення про успішність розв'язання задачі присвоюється ім'я **optymizatsiya**;

- до цієї змінної записується повідомлення про неуспішність розв'язання задачі, наприклад: **'nevdala_optymizatsiya'**;

optymizatsiya = 'nevdala_optymizatsiya';

end

- другий цикл **if…end** завершується;

- починається виконання наступного, 16-го пункту програми.

16. Для візуалізації знайденого в результаті розв'язання задачі абсолютного мінімуму функції, потрібно записати наступний рядок програми:

plot(x_min, f_min, 'ro');

Увага ! Синтаксис команди **plot**, детально описаний в пункті 6 алгоритму написання даної програми.

Варто зазначити, що кількість вхідних аргументів команди **plot** (вхідні аргументи записуються в дужках після слова **plot**) в пункті 16, дорівнює трьом, на відміну від кількості вхідних аргументів цієї команди в пункті 6, яке дорівнює двом.

За допомогою третього аргументу здійснюється зміна типу та кольору лінії, яка відображає графік функції в полі файлу. В даній задачі таким є файл, що має назву **Figure 1.**

У випадку завдання двох вхідних аргументів функції **plot** лінія графіка функції буде відображатись суцільною та матиме синій колір.

В разі завдання трьох вхідних аргументів функції **plot** (тобто додаткового аргументу 'ro') точка з координатами, незалежного аргументу **x**, якому відповідає значення абсолютного мінімуму функції, буде відображатись у вигляді кільця червоного кольору.

Такий вигляд точки, яка відображає абсолютний мінімум функції, обумовлений тим, що літерою **r** (red- червоний) визначається колір точки, а літерою **o** (ця літера має форму кільця) визначається її тип.

Як вже згадувалось в пункті 8, для одночасного відображення в полі одного

файлу, який має назву **Figure 1**, лінії графіку функції та точки на графіку, якій відповідає абсолютний мінімум функції, обов'язково потрібно записати наступний рядок програми:

hold on

При нехтуванні запису цього рядка до програми у потрібному місці, а саме після рядка програми, який містить команду для відображення графіку функції по заданим масивам незалежного аргументу **x** та залежної від нього функції **y**, в полі файлу, який має назву **Figure 1** буде відображатись лише точка, якій відповідає абсолютний мінімум функції. Лінія графіку функції буде відсутньою.

---

17. Останнім рядком програми можна задати команду, за допомогою якої після завершення процесу розв'язання задачі в командному вікні (**Command Prompt)** або робочому полі (**Workspace)** можна переглянути тип розв'язувача, за допомогою якого COMSOL® Optimization Lab здійснював пошук абсолютного мінімуму функції:

algorythm_optymizatsiyi = opt.sol.algorithm;

Тип розв'язувача знаходиться в підструктурі з ім'ям **opt.sol.algorithm**.

Як видно з форми запису підструктури **opt.sol.algorithm**, її ім'я складається з наступних складових:

opt- **opt**imization (оптимізація);

sol- **sol**ution (розв'язок);

algorithm - **algorithm** (алгоритм). Третя (**algorithm**) складова імені цієї підструктури вказуює на те, що підструктура з ім'ям **opt.sol.algorithm** використовується для запису до неї типу алгоритму оптимізації.

В лівій частині виразу, наведеному в пункті 17, записується ім'я змінної **algorythm_optymizatsiyi**, яка використовується для запису до неї типу алгоритму оптимізації після його добування з підструктури **opt.sol.algorithm**.

Увага ! Як вже згадувалось раніше, на відміну від імен підструктур, змінній, яка використовується для запису до неї типу алгоритму оптимізації, можна присвоїти довільне ім'я. В даній задачі такій змінній присвоєно ім'я **algorytm_optymizatsiyi**.

Пункти 5-8, 12, 16 та 17 даної програми записуються лише для поліпшення наочного сприйняття результатів розв'язання задачі і ними можна знехтувати при написанні програми.

18. Для розв'язання задачі в командному рядку програмного продукту COMSOL® SCRIPT™ потрібно записати наступне:

run optymizatsiya_function_1

Увага ! Після слова "run" ставиться пробіл та записується ім'я m-файлу. В даній задачі файл має ім'я optymizatsiya_function_1.

19. Натиснути **Enter**.

Результати розв'язання даної задачі наводяться в робочому полі (**Workspace)** у вигляді, наведеному в таблиці 2.1:

Таблиця 2.1- Результати розв'язання задачі

Name	Value
algorythm_optymizatsiyi	'optnm'
f_min	-1.0000
opt	$[1\times 1\,\text{struct}]$
optymizatsiya	'uspishna_optymizatsiya'
options	$[1\times 1\,\text{struct}]$
x_min	$[1.5708]$
x_plot	$[1\times 801\,\text{double}]$
y_plot	$[1\times 801\,\text{double}]$

Як видно з таблиці 2.1, абсолютний мінімум функції $y=-\sin(x)$ знаходиться в точці з координатами: $x=1.5708$; $y=-1.0000$.

При введенні задач безумовної та умовної, лінійної та нелінійної оптимізації можуть використовуватись підструктури з такими іменами:

opt.obj- використовується для задання цільової функції;

opt.lc- використовується для задання лінійних обмежень;

opt.nc- використовується для задання нелінійних обмежень;

opt.bc- використовується для задання обмежень верхньої та нижньої

границі незалежних аргументів;

opt.init- використовується для задання початкових значень незалежних аргументів;

opt.sol- використовується для запису результатів розв'язання задачі;

opt.obj.c- використовується для задання вектора **C**;

opt.obj.h- використовується для задання вектора **H**;

opt.obj.form- використовується для визначення типу (лінійна, квадратична, нелінійна) оптимізаційної задачі;

opt.obj.f- використовується для задання цільової функції;

opt.obj.g- використовується для задання градієнта цільової функції;

Увага ! Деякі з наведених вище підструктур вже детально розглянуто в межах навчального посібника, деякі з них буде розглянуто згодом. Розгляд певним підструктур виходить за межі навчального посібника.

Для закріплювання вивченого матеріалу розглянемо ще одну задачу.

Приклад 2.2.

Потрібно знайти абсолютний мінімум функції $z = x^4 + y^4 + 2 \cdot x^2 \cdot y^2 - 4 \cdot x + 3$.

Для розв'язання задачі необхідно виконати наступні кроки:

1. Відкрити COMSOL® SCRIPT™.

2. Виконати дії, які описано в пунктах 2…4 **прикладу 2.1**.

3. Далі наступними двома рядками потрібно задати масиви значень двох незалежних аргументів **x** та **y**. Для цього можна скористатись вбудованою функцією **linspace** (**linearly spaced** – лінійно розташоване), за допомогою якої створюється масив значень по заданим початковому та кінцевому значенню масиву зі сталим кроком між значеннями незалежних аргументів **x** та **y**, який визначається кількістю значень в масиві:

x1 = linspace(-6,6,500);

y1 = linspace(-6,6,500);

Увага ! У наведених вище двох рядках:

- **x1** та **y1**- імена масивів незалежних аргументів **x** та **y** відповідно;

- перше число в дужках відповідає початковим значенням масивів (в обох рядках початкові значення масивів незалежних аргументів **x** та **y** дорівнюють “-6”);

- друге число відповідає кінцевим значенням масивів (в обох рядках початкові значення масивів незалежних аргументів **x** та **y** дорівнюють “6”).

- третє число відповідає кількості значень незалежних аргументів **x** та **y** в масивах (в обох рядках кількість значень незалежних аргументів **x** та **y** дорівнює “500”).

Складовою імен масивів незалежних аргументів **x** та **y** є індекс “**1**” поруч з позначенням незалежних аргументів **x** та **y**, оскільки позначення “**x**” та “**y**” незалежних аргументів **x** та **y** будуть використані в пунктах 4 та 5 при відображенні та запису математичного виразу цільової функції.

Пункт 3 є початком низки команд для відображення тривимірного графіку функції, абсолютний мінімум якої потрібно знайти. Можна відмовитись від побудови та відображення графіку функції $z = x^4 + y^4 + 2 \cdot x^2 \cdot y^2 - 4 \cdot x + 3$, оскільки цей крок при написанні програми потрібний в подальшому лише для візуалізації точки на графіку, якій відповідає значення абсолютного мінімуму функції. В такому випадку пунктами 3…6 можна знехтувати.

---

4. Для відображення графіку функції $z = x^4 + y^4 + 2 \cdot x^2 \cdot y^2 - 4 \cdot x + 3$ необхідно сформувати двовимірний масив значень незалежних аргументів **x** та **y** по створеним у пункті 3 одновимірним масивам **x1** та **y1**. Таким чином, кожному значенню незалежного аргументу **x** відповідає певне значення незалежного аргументу **y**.

Для цього використовується команда **meshgrid** (**meshgrid** – сітка вічок):

[x, y] = meshgrid (x1, y1);

Синтаксис команди **meshgrid** наступний. Спочатку, тобто зліва від знаку “=” в якості вихідних параметрів команди **meshgrid** у квадратних дужках записуються імена незалежних аргументів **x** та **y**, до яких команда **meshgrid**, створивши попередньо, записує двовимірний масив значень.

Потім ставиться знак “=”, після якого, власне, записується команда **meshgrid**. Після команди **meshgrid** в дужках записуються імена одновимірних масивів незалежних аргументів **x1** та **y1**, з яких формується

двовимірний масив незалежних аргументів **x** та **y**.

5. Записується математичний вираз функції, графік якої потрібно відобразити, тобто функції $z = x^4 + y^4 + 2 \cdot x^2 \cdot y^2 - 4 \cdot x + 3$:

z = x.^4+y.^4+2.*x.^2*y.^2-4.*x+3;

Увага ! Необхідно зазначити, що за синтаксисом COMSOL® SCRIPT™ у виразах перед знаками **поелементного** множення “*”, ділення “/” та піднесення до степеня “^” завжди необхідно ставити крапку.

---

6. Будується тривимірний графік цільової функції по значеннях, якими є значення двовимірного масиву незалежних аргументів **x** та **y**, а також значення функції **z**, які розраховуються згідно з виразом, наведеним в пункті 5. Графік цільової функції $z = x^4 + y^4 + 2 \cdot x^2 \cdot y^2 - 4 \cdot x + 3$ будується за допомогою команди **surf** (**surface** – поверхня):

surf (x,y,z);

Синтаксис команди **surf** наступний. Після запису команди **surf** в дужках йде перелік осей, за якими будується графік. В даному випадку це осі **x**, **y** та **z**.

7. Для виконання наступних кроків, а саме:

- відображення в полі одного файлу, який має назву **Figure 1**, графіку функції та точки на графіку, якій відповідає абсолютний мінімум функції;

- запису в підструктуру з ім’ям **opt.obj.f** функції, абсолютний мінімум якої потрібно визначити;

- запису в підструктуру з ім’ям **opt.init.x** довільних значень незалежних аргументів **x** та **y**, починаючи з яких COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції;

- задання команди, за допомогою якої буде виконуватись пошук абсолютного мінімуму цільової функції;

- задання команди добування значень незалежних аргументів **x** та **y** з підструктури **opt.sol.x**;

- задання команди для добування значення абсолютного мінімуму функції з підструктури **opt.sol.eval.f**;

- контролю за успішністю розв’язання задачі, можна скористатись пунктами 8…15 **прикладу 2.1**.

Кроки, які перелічені в пункті 7, записуються у вигляді рядків програми наступним чином:

```
hold on
opt.obj.f = inline ('x(1)^4+x(2)^4+2.*x(1)^2*x(2)^2-4*x(1)+3');
opt.init.x = [10 10];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
if opt.sol.exit == 1;
optymizatsiya = 'uspishna optymizatsiya';
end
if opt.sol.exit == 0;
optymizatsiya = 'nevdala optymizatsiya';
end
```

Увага ! В підструктуру з ім’ям **opt.obj.f** функція, абсолютний мінімум якої потрібно визначити, записується у дещо іншому вигляді, ніж це наведено в пункті 5. Так, незалежний аргумент **x** при запису функції до структури **opt.obj.f** записується як **x(1)**, а незалежний аргумент **y** записується як **x(2)**.

Тоді запис функції до структури **opt.obj.f** матиме наступний вигляд:

opt.obj.f = inline ('x(1)^4+x(2)^4+2.*x(1)^2*x(2)^2-4*x(1)+3');

Необхідно зазначити, що ім’я всіх незалежних аргументів (**x**, **y** тощо) при запису функції до структури **opt.obj.f** може бути довільним (в даній задачі це ім’я **x**), проте однаковим для всіх незалежних аргументів. Відмінність одного незалежного аргументу від іншого полягає у додаванні до його імені індексу у дужках.

При запису в підструктуру з ім’ям **opt.init.x** довільних значень незалежних аргументів **x** та **y** в квадратних дужках, на відміну від **прикладу 2.1**, потрібно через

пробіл записати не одне, а два числа, що відповідає кількості незалежних аргументів:

opt.init.x = [10 10];

---

8. Для візуалізації знайденого в результаті розв'язання задачі абсолютного мінімуму функції потрібно записати наступний рядок:

plot3 (x_min(1,1), x_min(1,2), f_min, 'ro', 'linewidth', 3);

Команда **plot3** застосовується для відображення точки, яка відповідає абсолютному мінімуму функції, на тривимірному графіку функції.

Після назви команди за синтаксисом в дужках записуються параметри, за якими будується графік. Порядок запису параметрів наступний.

Першим вказується мінімальне значення незалежного аргументу **x**, яке записується у вигляді x_min(1,1).

Через кому другим вказується мінімальне значення незалежного аргументу **y**, яке записується у вигляді x_min(1,2).

Через кому третім вказується мінімальне значення функції **z**, яке записується у вигляді f_min.

Через кому четвертим в одинарних лапках вказується колір точки, яка відповідає абсолютному мінімуму функції.

Через кому п'ятим в одинарних лапках вказується параметр "товщина" (linewidth) лінії та значення товщини лінії (в даному прикладі вона дорівнює "3") при відображенні точки, яка відповідає абсолютному мінімуму функції.

9. Останнім рядком програми можна задати команду, за допомогою якої після завершення процесу розв'язання задачі в командному вікні (**Command Prompt)** або робочому полі (**Workspace)** можна переглянути тип розв'язувача, за допомогою якого COMSOL® Optimization Lab здійснював пошук абсолютного мінімуму функції:

algorythm_optymizatsiyi = opt.sol.algorithm;

Тип розв'язувача знаходиться в підструктурі з ім'ям

opt.sol.algorithm.

10. Для розв'язання задачі в командному рядку програмного продукту COMSOL® SCRIPT™ потрібно записати наступне:

run optymizatsiya_function_1

Увага ! Після слова "run" ставиться пробіл та записується ім'я m-файлу. В даній задачі файл має ім'я optymizatsiya_function_1.

---

11. Натиснути **Enter**.

Результати розв'язання даної задачі наводяться в робочому полі (**Workspace)** у вигляді, наведеному в таблиці 2.1:

Таблиця 2.1- Результати розв'язання задачі

Name	Value
algorythm_optymizatsiyi	'optnm'
f_min	0
opt	$[1\times 1\,\mathrm{struct}]$
optymizatsiya	'uspishna optymizatsiya'
x	[500 × 500 double]
x1	[1 × 500 double]
x_min	[1 0]
y	[500×500 double]
y1	[1×500 double]
z	[500×500 double]

Як видно з таблиці 2.1, абсолютний мінімум функції $z = x^4 + y^4 + 2x^2y^2 - 4x + 3$ знаходиться в точці з координатами: x = 1; y = 0; z = 0.

2.3 Приклади розв'язання теплоенергетичних задач безумовної оптимізації

Приклад 2.3.

Потужність, що споживається міксером-нагрівачем, визначається за

формулою:

$$P = 2 \cdot D + \mu^2 + 5 \cdot \gamma + n^2 - 10 \cdot n + 2000 \qquad (2.1)$$

де D - діаметр міксера-нагрівача, м;

n - швидкість обертання міксера-нагрівача, $\frac{об}{хв}$;

μ - динамічний коефіцієнт в'язкості рідини, що нагрівається, Па·с;

γ - питома вага рідини, $\frac{Па}{м}$.

Потрібно визначити мінімальну потужність, яка необхідна для обертання міксера-нагрівача, якщо питома вага рідини γ дорівнює $0{,}8\frac{Па}{м}$, динамічний коефіцієнт в'язкості рідини μ дорівнює 200 Па·с, а діаметр міксера-нагрівача дорівнює 2 м, тобто необхідно знайти P_{min} при мінімальній швидкості обертання міксера-нагрівача n_{min}.

Для розв'язання задачі необхідно виконати наступні кроки:

1. Відкрити COMSOL® SCRIPT™.

2. Виконати дії, які описано в пунктах 2…4 **прикладу 2.1**.

3.Задати константи, які задані умовою:

D = 2;

mu = 200;

gamma = 0.8;

Увага ! Літерою **D** позначено діаметр міксера-нагрівача, м, через **mu** позначено динамічний коефіцієнт в'язкості рідини, Па·с, через **gamma** – питому вагу рідини, $\frac{Па}{м}$.

4. Наступними двома рядками задати масиви значень обертів **n** та потужності **P**. Цей крок необхідний для побудови та відображення графіку функції, мінімізація якої розглядається в даному прикладі:

```
n_plot = [-100:1:100];
P_plot = 2.*D + mu.^2 + 5.*gamma + n_plot.^2 - 10.*n_plot + 2000;
```

5. Написати команду відображення графіку функції по заданим масивам незалежного аргументу **n** та залежної від нього функції **P**:

plot (n_plot, P_plot);

6. Для відображення ліній сітки осей **x** та **y** на графіку наступним рядком програми потрібно записати:

grid on

7. Для відображення в полі файлу, який має назву **Figure 1**, графіку функції та точки на графіку, якій відповідає абсолютний мінімум функції, наступним рядком програми потрібно записати:

hold on

8. Для виконання наступних кроків, а саме:

- відображення в полі файлу, який має назву **Figure 1**, графіку функції та точки на графіку, якій відповідає абсолютний мінімум функції;

- запису в підструктуру з ім'ям **opt.obj.f** функції, абсолютний мінімум якої потрібно визначити;

- запису в підструктуру з ім'ям **opt.init.x** довільного значення незалежного аргументу **n**, починаючи з якого COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції;

- задання команди, за допомогою якої буде виконуватись пошук абсолютного мінімуму цільової функції;

- задання команди добування значення незалежного аргументу **n** з підструктури **opt.sol.x**;

- задання команди для добування значення абсолютного мінімуму функції з підструктури **opt.sol.eval.f**;

- контролю за успішністю розв'язання задачі, можна скористатись пунктами 8…15 **прикладу 2.1.**

Кроки, які перелічені в пункті 8 даної задачі, записуються у вигляді рядків програми наступним чином:

opt.obj.f = inline ('2.*2+200.^2+5.*0.8+ x.^2-10.*x+2000');

opt.init.x = [0];

```
opt.sol = optnm (opt);

n_min = opt.sol.x;

P_min = opt.sol.eval.f;

if opt.sol.exit == 1;

optymizatsiya = 'uspishna optymizatsiya';

end

if opt.sol.exit == 0;

optymizatsiya = 'nevdala optymizatsiya';

end
```

Увага ! В підструктуру з ім'ям **opt.obj.f** функція, абсолютний мінімум якої потрібно визначити, записується у дещо іншому вигляді у порівнянні з формулою (2.1). Так, при записі виразу цільової функції в підструктуру **opt.obj.f** в ньому мають бути лише незалежні аргументи, а константи, які присутні в формулі (2.1), необхідно замінити на їх чисельні значення. В даному прикладі незалежним аргументом є швидкість обертання міксера-нагрівача **n**.

Тоді запис цільової функції в підструктуру **opt.obj.f** матиме наступний вигляд:

opt.obj.f = inline ('2.*2 + 200.^2 + 5.*0.8 + x.^2 - 10.*x + 2000');

Існує можливість запису цільової функції в підструктуру **opt.obj.f** у тому вигляді, в якому вона наведена у формулі (2.1). Для цього необхідно створити окремий файл-функцію і записати до нього вираз функції, мінімум якої потрібно знайти, а також константи, які входять до виразу цільової функції.

---

9. Для візуалізації знайденого в результаті розв'язання задачі абсолютного мінімуму функції можна скористатись пунктом 16 **прикладу 2.1** і записати наступний рядок програми:

plot(x_min, f_min, 'ro');

10. Останнім рядком програми можна задати команду, за допомогою якої після завершення процесу розв'язання задачі в командному вікні (**Command Prompt)** або робочому полі (**Workspace)** можна переглянути

тип розв’язувача, за допомогою якого COMSOL® Optimization Lab здійснював пошук абсолютного мінімуму функції (аналогічно **прикладу 2.1**):

algorythm_optymizatsiyi = opt.sol.algorithm;

11. Для розв’язання задачі в командному рядку програмного продукту COMSOL® SCRIPT™ потрібно записати наступне:

run optymizatsiya_function_1

Увага ! Після слова “run” ставиться пробіл та записується ім’я m-файлу. В даній задачі файл має ім’я optymizatsiya_function_1.

---

12. Натиснути **Enter**.

Результати розв’язання даної задачі наводяться в робочому полі (**Workspace)** у вигляді, наведеному в таблиці 2.2:

Таблиця 2.2- Результати розв’язання задачі

Name	Value
algorythm_optymizatsiyi	‘optnm’
D	2
n_min	[5]
n_plot	[1×101 double]
opt	[1×1 struct]
optymizatsiya	‘uspishna_optymizatsiya’
options	[1×1 struct]
P_min	41983
P_plot	[1×101 double]
mu	200
gamma	0.8000

Як видно з таблиці 2.2, мінімальна потужність, яка необхідна для функціонування міксера, складає 41983Вт, що відповідає швидкості обертання $5\,\frac{\text{об}}{\text{хв}}$.

Існує можливість запису цільової функції в підструктуру **opt.obj.f** у

тому вигляді, в якому вона наведена у формулі (2.1). Для цього необхідно створити окремий файл-функцію і записати до нього вираз функції, мінімум якої потрібно знайти, а також константи, які входять до виразу цільової функції. Розглянемо цей варіант розв’язання задачі в **прикладі 2.4**.

Приклад 2.4.

Необхідно розв’язати задачу, наведену в **прикладі 2.3**, за допомогою запису цільової функції до файлу-функції.

Для розв’язання задачі необхідно виконати дії, які описано в пунктах 2…7 **прикладу 2.2**.

8. Для запису в підструктуру з ім’ям **opt.obj.f** функції, абсолютний мінімум якої потрібно визначити, необхідно створити окремий файл-функцію.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

Для створення файлу-функції необхідно створити m-файл. Детальний опис послідовності створення m-файлу наведений в **прикладі 2.1**.

Ім’я файлу-функції повинно співпадати з ім’ям цільової функції, яка записується до файлу-функції. Наприклад, ім’ям цільової функції, а разом і файлу функції, буде ім’я **function_1**.

---

9. Потрібно зберегти файл-функцію. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім’я файлу-функції, тобто **function_1**.

10. Підструктуру **opt.obj.f** записують у наступному вигляді:

opt.obj.f = 'function_1';

Увага ! Після запису імені підструктури, тобто **opt.obj.f**, та знаку “=” записується ім’я файлу-функції, тобто, **function_1**. Ім’я файлу-функції виокремлюється з обох боків одинарними лапками.

В порожньому файлі-функції перший рядок починається зі слова **function**. У випадку, коли m-файл розпочинається зі слова **function**, такий файл сприймається програмними продуктами COMSOL MULTIPHYSICS®

та MATLAB® в якості файлу-функції.

Після слова **function** в першому рядку файлу-функції записується позначення цільової функції (в даному прикладі ця функція позначена через літеру **P**), а після знаку “=” вказується ім’я функції (в даному прикладі це **function_1**) після якого в дужках записується ім’я незалежного аргументу (в даному прикладі згідно з рівнянням (2.1) незалежним аргументом є швидкість обертання міксера-нагрівача **n**).

Таким чином, перший рядок файлу-функції має наступний вигляд:

```
function P = function_1(n)
```

Наступними рядками файлу-функції записуються константи, що містяться в цільовій функції, тобто у рівнянні (2.1):

```
D = 2;
mu = 200;
gamma = 0.8;
```

Наступним і останнім рядком до файлу-функції записується власне вираз самої цільової функції, мінімум якої потрібно знайти, тобто рівняння (2.1).

Останній рядок файлу-функції виглядатиме так:

```
P = 2.*D+mu.^2+5.*gamma+ n.^2-10.*n+2000;
```

В цілому, файл-функція має наступний вигляд:

```
function P = function_1(n)
D = 2;
mu = 200;
gamma = 0.8;
P = 2.*D+mu.^2+5.*gamma+ n.^2-10.*n+2000;
```

Увага ! Запис констант, що містяться в цільовій функції, завжди здійснюється перед рядком запису цільової функції. В іншому випадку COMSOL MULTIPHYSICS® видаватиме повідомлення про помилку і зупинятиме обчислення.

Редагування файлу-функції завершено. Його можна закрити, або залишити відкритим.

11. Наступним рядком в підструктуру з ім’ям **opt.obj.form** записується тип цільової функції (лінійна, квадратична, нелінійна), у відповідності до якого обирається тип розв’язання оптимізаційної задачі.

Цільова функція, тобто рівняння (2.1) є квадратичною, оскільки максимальним степенем піднесення незалежного аргументу **n** є квадрат.

Увага ! Незважаючи на те, що цільова функція є квадратичною, бажано для усих типів цільової функції обирати нелінійний тип розв’язання оптимізаційної задачі і записувати це після знаку “=” в підструктуру з ім’ям **opt.obj.form** наступним чином:

opt.obj.form = 'nlin';

Слово **nlin**, яке виокремлюється з обох боків одинарними лапками, є скороченням від слова nonlinear (нелінійний).

Послідовність виконання інших дій при розв’язанні задачі, а саме:

- запис в підструктуру з ім’ям **opt.init.x** довільного значення незалежного аргументу **n**, починаючи з якого COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції;

- задання команди, за допомогою якої буде виконуватись пошук абсолютного мінімуму цільової функції;

- задання команди добування значення незалежного аргументу **n** з підструктури **opt.sol.x**;

- задання команди для добування значення абсолютного мінімуму функції з підструктури **opt.sol.eval.f**;

- контроль за успішністю розв’язання задачі, описано в пункті 8 **прикладу 2.2**.

В цілому, програма для відшукування мінімуму цільової функції матиме наступний вигляд:

```
clear
clc
D = 2;
mu = 200;
gamma = 0.8;
n_plot = [0:1:100];
```

```
P_plot =2.*D+mu.^2+5.*gamma+ n_plot.^2-10.*n_plot+2000;
plot(n_plot,P_plot);
grid on
hold on
opt.obj.f = 'function_1';
opt.obj.form = 'nlin';
opt.init.x = [0];
opt.sol = optnm(opt);
options = optprop;
n_min = opt.sol.x;
P_min = opt.sol.eval.f;
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
if opt.sol.exit ==0;
optymizatsiya = 'nevdala_optymizatsiya';
end
plot(n_min,P_min,'ro') % изображение минимума на графике функции
algorythm_optymizatsiyi = opt.sol.algorithm;
```

При цьому результати розв’язання задачі (табл. 2.2) не змінюються, що вказує на коректний записцільової функції до файлу-функції.

Приклади задач безумовної оптимізації для самостійного розв’язання наведені в додатку А.

3 ПРОГРАМУВАННЯ ЗАДАЧ УМОВНОЇ ОПТИМІЗАЦІЇ

3.1 Загальні теоретичні положення оптимізації при наявності обмежень у вигляді рівностей

Розглянемо теоретичні основи одного зі способів мінімізації функції двох незалежних аргументів:

$$z = f(x, y) \tag{3.1}$$

де на **x** та **y** накладено обмеження, що задається у вигляді рівняння:

$$g(x, y) = 0 \tag{3.2}$$

Увага! Для спрощення викладання методики знаходження мінімуму функції при наявності обмежень у вигляді рівностей обрано функцію двох незалежних аргументів $z = f(x, y)$, однак кількість незалежних аргументів може бути значно більшою.

Рівняння $g(x, y) = 0$ можна розв’язати відносно **y** як функції **x**, тобто:

$$y = h(x) \tag{3.3}$$

На практиці може виявитися важким або навіть неможливим знайти явний вигляд функції $h(x)$.

При умові, що існує похідна функції $h(x)$:

$$\frac{dy}{dx} = \frac{d(h(x))}{dx} = -\frac{\frac{\partial g}{\partial x}}{\frac{\partial g}{\partial y}} \tag{3.4}$$

Тоді функцію $z = f(x, y) = f(x, h(x))$ можна записати як функцію однієї незалежної змінної **x**.

Необхідною умовою мінімуму функції **z** буде співвідношення:

$$\frac{dz}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dx} = 0 \tag{3.5}$$

Таким чином, для отримання значень **x** та **y** в точці мінімуму функції **z** необхідно знайти розв'язок системи рівнянь, що складається з рівняння (3.5) та рівняння (3.2).

Покажемо тепер на прикладі як користуючись формулами (3.1)-(3.5), можна відшукати мінімум функції.

Приклад 3.1.

Знайти мінімум функції $f(x, y) = x^2 + y^2$ при обмеженні $x + y = 4$.

В даній задачі згідно з формулою (3.1):

$$f(x, y) = x^2 + y^2 \tag{3.6}$$

У рівнянні, що описує обмеження, що накладається на функцію, мінімальне значення якої потрібно знайти, виразимо **y** через **x**:

$$y = 4 - x \tag{3.7}$$

У відповідності із формулою (3.5) знайдемо похідні $\frac{\partial f}{\partial x}$ та $\frac{\partial f}{\partial y}$ функції $f(x, y) = x^2 + y^2$:

$$\frac{\partial f}{\partial x} = 2 \cdot x \tag{3.8}$$

При диференціюванні функції по змінній **x**, функція **y** є константою (тобто звичайним числом).

Аналогічним чином:

$$\frac{\partial f}{\partial y} = 2 \cdot y \tag{3.9}$$

У відповідності з формулою (3.5) знайдемо похідну обмеження $\frac{dy}{dx}$, яке накладається на функцію, що мінімізується:

$$\frac{dy}{dx} = -1 \tag{3.10}$$

Підставимо рівняння (3.8)-(3.10) у формулу (3.5):

$$\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot \frac{dy}{dx} = 2 \cdot x + 2 \cdot y \cdot (-1) = 0 \tag{3.11}$$

Рішення системи, що складається з рівняння (3.11) та обмеження (3.7), дозволяє визначити значення x_{min} та y_{min} при яких значення функції **f** є мінімальним:

$$\begin{cases} 2 \cdot x - 2 \cdot y = 0 \\ x + y = 4 \end{cases} \qquad (3.12)$$

Розв'язання системи рівнянь (3.12) дає наступні результати:

$$x_{min} = 2;$$

$$y_{min} = 2.$$

Мінімальне значення функції **f** при $x_{min} = 2$ та $y_{min} = 2$ дорівнює 8.

Приклад 3.2.

Відшукаємо розв'язок попередньої задачі за допомогою модуля оптимізації OPTIMIZATION LAB програми COMSOL MULTIPHYSICS.

Увага ! Дана задача відноситься до задач нелінійної оптимізації, оскільки цільова функція $f(x, y) = x^2 + y^2$ є нелінійною, незважаючи на лінійне обмеження $x + y = 4$.

Для розв'язання задачі необхідно виконати наступні кроки:

1. Відкрити COMSOL® SCRIPT™.

2. Для запису в підструктуру з ім'ям **opt.obj.f** функції, абсолютний мінімум якої потрібно визначити, необхідно створити окремий файл-функцію.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

Для створення файлу-функції необхідно створити m-файл. Детальний опис послідовності створення m-файлу наведений в **прикладі 2.1**.

Ім'я файлу-функції повинно співпадати з ім'ям цільової функції, яка записується до файлу-функції. Наприклад, ім'ям цільової функції, а разом і файлу функції, буде ім'я **function_1**.

---

3. Потрібно зберегти файл-функцію. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім'я файлу-функції, тобто **function_1**.

4. В порожньому файлі-функції перший рядок починається зі слова

function. У випадку, коли m-файл розпочинається зі слова **function**, такий файл сприймається програмними продуктами COMSOL MULTIPHYSICS® та MATLAB® в якості файлу-функції.

Після слова **function** в першому рядку файлу-функції записується позначення цільової функції (в даному прикладі ця функція позначена через літеру **f**), а після знаку “=” вказується ім’я функції (в даному прикладі це **function_1**) після якого в дужках записується ім’я незалежного аргументу, або декількох незалежних аргументів, від яких залежить функція (в даному прикладі, згідно з рівнянням (3.6), незалежними аргументами є **x** та **y**).

Таким чином, перший рядок файлу-функції має наступний вигляд:

function f = function_1(x)

Увага ! Необхідно зазначити, що ім’я всіх незалежних аргументів (**x**, **y** тощо) при запису математичного виразу функції у файлі-функції може бути довільним (в даній задачі це ім’я **x**), проте однаковим для всіх незалежних аргументів. Відмінність одного незалежного аргументу від іншого полягає у додаванні до його імені індексу у дужках (1, 2 тощо).

Математичний вираз функції записується у дещо іншому вигляді, ніж це наведено у рівнянні (3.6). Так, незалежний аргумент **x** записується як **x(1)**, а незалежний аргумент **y** записується як **x(2)**.

Тоді математичний вираз функції, який записується другим і останнім рядком до файлу-функції, матиме наступний вигляд:

f = x(1)^2 + x(2)^2

В цілому, файл-функція має наступний вигляд:

function f = function_1(x)

f = x(1)^2 + x(2)^2

Редагування файлу-функції завершено. Його можна закрити, або залишити відкритим.

5. Створити m-файл. Для цього необхідно зайти в пункт меню **File > New editor.** В полі вікна COMSOL Desktop® відкривається порожній m-файл.

Потрібно зберегти файл, попередньо присвоївши йому ім’я. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім’я файлу, наприклад, **optymizatsiya_function_1**.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

---

6. Написати першим рядком робочого поля m- файлу вбудовану команду **clear**.

7. Написати наступним рядком робочого поля m-файлу вбудовану команду **clc**.

8. Записати цільову функцію в підструктуру **opt.obj.f**.

Для цього підструктуру **opt.obj.f** записують у наступному вигляді:

opt.obj.f = 'function_1';

Увага! Після запису імені підструктури та знаку “=” записується назва файлу-функції. В даному випадку файл-функція має назву function_1. Назва файлу-функції виділяється одинарними лапками.

---

9. Наступним рядком в підструктуру з ім’ям **opt.obj.form** записується тип цільової функції (лінійна, квадратична, нелінійна), у відповідності до якого обирається тип розв’язання оптимізаційної задачі.

Цільова функція, тобто рівняння (3.6), є квадратичною, оскільки максимальним степенем піднесення незалежних аргументів **x** та **y** є квадрат.

Увага ! Незважаючи на те, що цільова функція є квадратичною бажано для усих типів цільової функції обирати нелінійний тип розв’язання оптимізаційної задачі і записувати це після знаку “=” в підструктуру з ім’ям **opt.obj.form** наступним чином:

opt.obj.form = 'nlin';

Слово **nlin**, яке виокремлюється з обох боків одинарними лапками, є скороченням від слова nonlinear (нелінійний).

---

10. Наступним рядком до m-файлу потрібно записати обмеження (рівняння (3.12)), яке накладено на цільову функцію

$$x + y = 4 \qquad (3.12)$$

Виконаємо аналіз математичного виразу обмеження:

1) Обмеження задано у вигляді рівності.

2) Обмеження є лінійним, оскільки максимальним степенем піднесення незалежних аргументів **x** та **y** є одиниця.

Схема підструктур для задання лінійних обмежень наведена на рисунку 3.1.

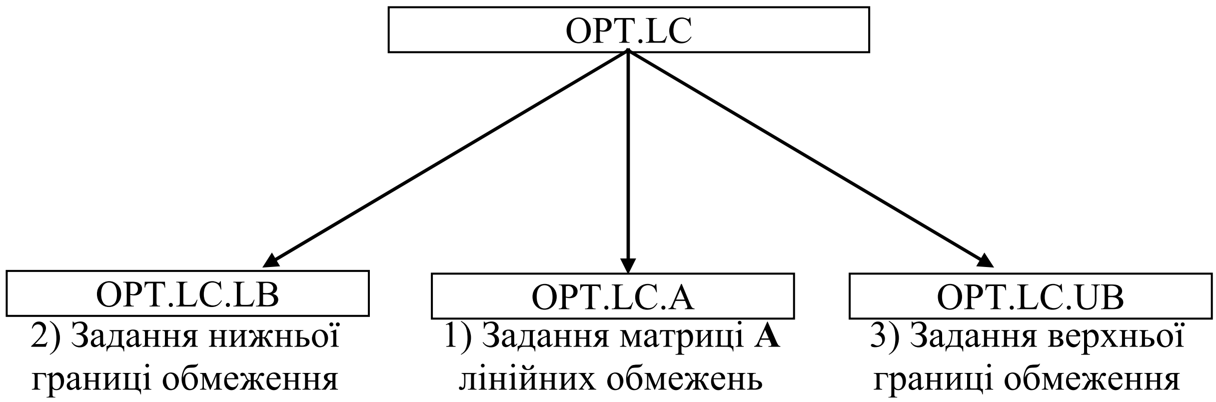

Рисунок 3.1- Схема підструктур для задання лінійних обмежень

Увага ! Схема підструктур для задання лінійних обмежень є універсальною, тобто вона використовується для задання лінійних обмежень як у вигляді рівностей, так і у вигляді нерівностей.

Номерами 1, 2 та 3 на **рисунку 3.1** позначено послідовність запису підструктур в m-файл.

Як видно з форми запису підструктури **opt.lc**, її ім'я складається з наступних складових:

opt- **opt**imization (оптимізація). Ця складова імені записується першою та міститься в іменах всіх підструктур, які будуть зустрічатись в подальшому при розгляді алгоритму розв'язання поточної та наступних задач оптимізації;

lc- **l**inear **c**onstraints (лінійні обмеження). Друга (**lc**) складова імені цієї підструктури вказує на те, що підструктура з ім'ям **opt.lc** використовується для запису до неї лінійних обмежень.

Увага ! Складові імені підструктури відокремлюються одне від одного за допомогою крапки. Присвоювати довільні імена складовим підструктур неможна. Тобто, наприклад, першій складовій всіх підструктур потрібно присвоювати тільки ім'я **opt**, оскільки COMSOL® Optimization Lab сприймає та працює лише з підструктурами, які мають певні, зрозумілі для програмного продукту, сталі імена. Те ж саме стосується

складових імені підструктури **opt.lc** тощо.

Записати в підструктуру з ім’ям **opt.lc.A** ліву частину лінійного обмеження, тобто ліву частину рівняння (3.12), у вигляді:

opt.lc.A = [1 1];

В лівій частині виразу записується підструктура з ім’ям **opt.lc.A**, яка використовується для запису до неї матриці **A**, що представляє собою вектор-рядок, що містить з числові коефіцієнти при незалежних аргументах **x** та **y** лівої частини рівняння (3.12).

У докладному вигляді рівняння (3.12) можна записати наступним чином:

$$+1 \cdot x + 1 \cdot y = +4 \qquad (3.13)$$

Як видно з рівняння (3.13), числовий коефіцієнт при незалежному аргументі **x** дорівнює “+1”, числовий коефіцієнт при незалежному аргументі **y** також дорівнює “+1”.

Увага ! Знак “плюс”, на відміну від знаку “мінус” при запису до матриці **A** числових коефіцієнтів при незалежних аргументах, не ставиться.

Числові коефіцієнти при запису їх до матриці **A** розділяються пробілом.

Матриця **A** представляє собою вектор-рядок лише для даної задачі.

Сформулюємо наступне правило: кількість рядків матриці **A** дорівнює кількості лінійних обмежень, які накладено на цільову функцію.

Оскільки в даній задачі на цільову функцію накладено лише одне лінійне обмеження, яке складається з двох незалежних аргументів **x** та **y**, матриця **A** представляє собою матрицю, яка містить один рядок та два стовпчика.

Числові значення у векторах-рядках відокремлюються пробілом.

---

10. Наступними двома рядками до певних підструктур m-файлу потрібно записати праву від знаку “=”частину обмеження $x + y = 4$, тобто цифру “+4”.

Для запису правої частини рівняння (3.12) потрібно скористатись підструктурами **opt.lc.lb** та **opt.lc.ub**.

Як видно з форми запису підструктури **opt.lc**, її ім'я складається зі складових **opt** та **lc**, які були розглянуті раніше, а також складових:

lb- **l**ower **b**oundary (нижня границя). Ця складова імені підструктури **opt.lc**.**lb** вказує на те, що вона використовується для запису нижньої границі лінійного обмеження.

ub- **u**pper **b**oundary (верхня границя). Ця складова імені підструктури **opt.lc**.**ub** вказує на те, що вона використовується для запису верхньої границі лінійного обмеження.

Увага ! Складові імені підструктури відокремлюються одне від одного за допомогою крапки. Присвоювати довільні імена складовим підструктур неможна. Тобто, наприклад, першій складовій всіх підструктур потрібно присвоювати тільки ім'я **opt**, оскільки COMSOL® Optimization Lab сприймає та працює лише з підструктурами, які мають певні, зрозумілі для програмного продукту, сталі імена.

Таким чином, підструктури **opt.lc**.**lb** та **opt.lc**.**ub** матимуть наступний вигляд:

opt.lc.lb = 4;

opt.lc.ub = 4;

Увага ! Оскільки в даній задачі на цільову функцію накладено обмеження у вигляді рівності, нижня границя обмеження дорівнює верхній частині обмеження.

---

11. Для виконання наступних кроків, а саме:

- запису в підструктуру з ім'ям **opt.init.x** довільних значень незалежних аргументів **x** та **y**, починаючи з яких COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції;

- задання команди, за допомогою якої буде виконуватись пошук абсолютного мінімуму цільової функції;

- задання команди добування значень незалежних аргументів **x** та **y** з підструктури **opt.sol.x**;

- задання команди для добування значення абсолютного мінімуму функції з підструктури **opt.sol.eval.f**;

- контролю за успішністю розв'язання задачі, можна скористатись

пунктами 10…15 **прикладу 2.1.**

Увага ! На відміну від **прикладу 2.1** при запису в підструктуру з ім'ям **opt.init.x** довільних значень незалежних аргументів **x** та **y** в квадратних дужках потрібно через пробіл записати не одне, а два числа, що відповідає кількості незалежних аргументів:

opt.init.x = [0 0];

Кроки, які перелічені в пункті 11 даної задачі, записуються у вигляді рядків програми наступним чином:

```
opt.init.x = [0 0];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
x_min_1 = x_min(1,1);
x_min_2 = x_min(1,2);
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
if opt.sol.exit == 0;
optymizatsiya = 'nevdala_optymizatsiya';
end
```

12. Для розв'язання задачі в командному рядку програмного продукту COMSOL® SCRIPT™ потрібно записати наступне:

run optymizatsiya_function_1

Увага ! Після слова "run" ставиться пробіл та записується ім'я m-файлу. В даній задачі файл має ім'я optymizatsiya_function_1.

---

13. Натиснути **Enter**.

В цілому, програма для відшукування мінімуму цільової функції матиме наступний вигляд:

```
clc
clear
```

```
opt.obj.f = 'function_1';
opt.obj.form = 'nlin';
opt.lc.A = [1 1];
opt.lc.lb = 4;
opt.lc.ub = 4;
opt.init.x = [0 0];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
x_min_1 = x_min(1,1);
x_min_2 = x_min(1,2);
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
if opt.sol.exit == 0;
optymizatsiya = 'nevdala_optymizatsiya';
end
```

Результати розв’язання даної задачі наводяться в робочому полі (**Workspace)** у вигляді, наведеному в таблиці 3.1.

Таблиця 3.1- Результати розв’язання задачі

Name	Value
algorythm_optymizatsiyi	‘optnm’
f_min	8
opt	$[1 \times 1\,\mathrm{struct}]$
optymizatsiya	‘uspishna optymizatsiya’
x_min	[2 2]
x_min_1	2
x_min_2	2

Як видно з табл. 3.1 результати розв’язання задачі (табл. 2.2) не змінюються в порівнянні з **прикладом 3.1**, що вказує на правильне

використання файлу-функції.

3.2 Загальні теоретичні положення оптимізації при наявності обмежень у вигляді рівностей та/або нерівностей

Розглянемо наступну задачу.

Приклад 3.3.

На придбання насосів для нової котельні виділено 20000 умовних грошових одиниць (у.г.о.). Площина дільниці, на якій мають розташовуватись насоси не повинна перевищувати 38м^2.

Існує можливість придбати насоси типу “А” та типу “Б”. При цьому насоси типу “А” коштують 5000у.г.о., займають площину 8м^2 кожний та забезпечують продуктивність 7м^3/хв. Насоси типу “Б” коштують 2000у.г.о., займають площину 4м^2 кожний та забезпечують продуктивність 3м^3/хв.

Необхідно розрахувати оптимальний варіант придбання насосів, який забезпечує при певних обмеженнях максимальну продуктивність дільниці насосів.

Позначимо через **x** кількість насосів типу “А”, через **y**– кількість насосів типу “Б”. Таким чином, необхідно обрати певну кількість насосів типу “А” та насосів типу “Б”, щоб забезпечити максимальну продуктивність **P** дільниці насосів за одну хвилину.

В такому випадку вираз для цільової функції, максимальне значення якої потрібно знайти, матиме вигляд:

$$P = 7 \cdot x + 3 \cdot y \qquad (3.14)$$

При цьому необхідно дотриматись наступних обмежень:

- обмеження на вартість обладнання (в у.г.о.):

$$5000 \cdot x + 2000 \cdot y \leq 20000 \tag{3.15}$$

- обмеження на площину дільниці, на якій мають розташовуватись насоси (в м^2):

$$8 \cdot x + 4 \cdot y \leq 38 \tag{3.16}$$

- обмеження на невід’ємність кількості насосів типу “А” та типу “Б”:

$$x \geq 0 \tag{3.17}$$

$$y \geq 0 \tag{3.18}$$

Наведену вище задачу можна розв’язати, зокрема, методом перебору.

Обмеження на вартість обладнання та обмеження на площину дільниці вказують на те, що $x \leq 4$. Таким чином, **x** може приймати лише одне з п’яти значень: 0, 1, 2, 3 або 4.

Якщо $x = 4$, то з обмеження на вартість обладнання випливає, що $y = 0$, тому:

$$P = 7 \cdot x + 3 \cdot y = 7 \cdot 4 + 3 \cdot 0 = 28$$

Якщо $x = 3$, то з обмеження на вартість обладнання випливає, що $y \leq 2$, а з обмеження на площину дільниці випливає, що $y \leq 3$. Отже, максимальна продуктивність при умові дотримання обмежень досягається при $y = 2$, а саме:

$$P = 7 \cdot x + 3 \cdot y = 7 \cdot 3 + 3 \cdot 2 = 27$$

Якщо $x = 2$, то з обмеження на вартість обладнання випливає, що $y \le 5$, з обмеження на площину дільниці також випливає, що $y \le 5$. Це означає, що максимальна продуктивність при умові дотримання обмежень досягається при $y = 5$, а саме:

$$P = 7 \cdot x + 3 \cdot y = 7 \cdot 2 + 3 \cdot 5 = 29$$

Якщо $x = 1$, то з обмеження на вартість обладнання випливає, що $y \le 7$, з обмеження на площину дільниці також випливає, що $y \le 7$. Це означає, що максимальна продуктивність при умові дотримання обмежень досягається при $y = 7$, а саме:

$$P = 7 \cdot x + 3 \cdot y = 7 \cdot 1 + 3 \cdot 7 = 28$$

Якщо $x = 0$, то з обмеження на вартість обладнання випливає, що $y \le 10$, з обмеження на площину дільниці також випливає, що $y \le 9$. Це означає, що максимальна продуктивність при умові дотримання обмежень досягається при $y = 9$, а саме:

$$P = 7 \cdot x + 3 \cdot y = 7 \cdot 0 + 3 \cdot 9 = 27$$

Всі можливі варіанти розглянуто. Максимальна продуктивність $P = 29 \text{м}^3/\text{хв}$ досягається при $x = 2$ та $y = 5$. Таким чином, необхідно купити два насоси тупу “А” та п’ять насосів типу “Б”.

Приклад 3.4.

Відшукаємо розв’язок попередньої задачі за допомогою модуля оптимізації COMSOL® Optimization Lab програмного продукту COMSOL MULTIPHYSICS®.

Увага ! Дана задача відноситься до задач лінійної оптимізації, оскільки

цільова функція (3.14) та обмеження (3.15)-(3.18) є лінійними.

Для розв'язання задачі необхідно виконати наступні кроки:

1. Відкрити COMSOL® SCRIPT™.

2. Для запису в підструктуру з ім'ям **opt.obj.f** функції, абсолютний мінімум якої потрібно визначити, необхідно створити окремий файл-функцію.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

Для створення файлу-функції необхідно створити m-файл. Детальний опис послідовності створення m-файлу наведений в **прикладі 2.1**.

Ім'я файлу-функції повинно співпадати з ім'ям цільової функції, яка записується до файлу-функції. Наприклад, ім'ям цільової функції, а разом і файлу функції, буде ім'я **function_1**.

3. Потрібно зберегти файл-функцію. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім'я файлу-функції, тобто **function_1**.

4. В порожньому файлі-функції перший рядок починається зі слова **function**. У випадку, коли m-файл розпочинається зі слова **function**, такий файл сприймається програмними продуктами COMSOL MULTIPHYSICS® та MATLAB® в якості файлу-функції.

Після слова **function** в першому рядку файлу-функції записується позначення цільової функції (в даному прикладі ця функція позначена через літеру **f**), а після знаку "=" вказується ім'я функції (в даному прикладі це **function_1**) після якого в дужках записується ім'я незалежного аргументу, або декількох незалежних аргументів, від яких залежить функція (в даному прикладі згідно з рівнянням (3.14) незалежними аргументами є **x** та **y**).

Таким чином, перший рядок файлу-функції має наступний вигляд:

function f = function_1(x)

Увага ! Необхідно зазначити, що ім'я всіх незалежних аргументів (**x**, **y** тощо) при запису математичного виразу функції у файлі-функції може бути довільним (в

даній задачі це ім’я **x**), проте однаковим для всіх незалежних аргументів. Відмінність одного незалежного аргументу від іншого полягає у додаванні до його імені індексу у дужках (1, 2 тощо).

Математичний вираз функції записується у дещо іншому вигляді, ніж це наведено у рівнянні (3.14). Так, незалежний аргумент **x** записується як **x(1)**, а незалежний аргумент **y** записується як **x(2)**.

Тоді математичний вираз функції, який записується другим і останнім рядком до файлу-функції, матиме наступний вигляд:

f = 7*x(1) + 3*x(2)

В цілому, файл-функція має наступний вигляд:

```
function P = function_1(x)
P = 7*x(1) + 3*x(2)
```

Редагування файлу-функції завершено. Його можна закрити, або залишити відкритим.

5. Створити m-файл. Для цього необхідно зайти в пункт меню **File** > **New editor.** В полі вікна COMSOL Desktop® відкривається порожній m-файл.

Потрібно зберегти файл, попередньо присвоївши йому ім’я. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім’я файлу, наприклад, **optymizatsiya_function_1**.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

---

6. Написати першим рядком робочого поля m-файлу вбудовану команду **clear**.

7. Написати наступним рядком робочого поля m-файлу вбудовану команду **clc**.

8. Записати цільову функцію в підструктуру **opt.obj.f**.

Для цього підструктуру **opt.obj.f** записують у наступному вигляді:

```
opt.obj.f = 'function_1';
```

Увага! Після запису імені підструктури та знаку “=” записується назва файлу-функції. В даному випадку файл-функція має назву function_1. Назва файлу-функції

виділяється одинарними лапками.

---

9. Наступним рядком в підструктуру з ім'ям **opt.obj.form** записується тип цільової функції (лінійна, квадратична, нелінійна), у відповідності до якого обирається тип розв'язання оптимізаційної задачі.

Цільова функція, тобто рівняння (3.14), є лінійною, оскільки максимальним степенем піднесення незалежних аргументів **x** та **y** є одиниця.

Увага ! Незважаючи на те, що цільова функція є лінійною бажано для усих типів цільової функції обирати нелінійний тип розв'язання оптимізаційної задачі і записувати це після знаку “=” або знаків “≤” та “≥” в підструктуру з ім'ям **opt.obj.form** наступним чином:

```
opt.obj.form = 'nlin';
```

Слово **nlin**, яке виокремлюється з обох боків одинарними лапками, є скороченням від слова nonlinear (нелінійний).

---

10. Наступними рядками до m-файлу потрібно записати обмеження (рівняння (3.15)-(3.18)), які накладено на цільову функцію.

Виконаємо аналіз математичних виразів обмежень:

1) Обмеження задано у вигляді нерівностей.

2) Обмеження є лінійними, оскільки максимальним степенем піднесення незалежних аргументів **x** та **y** є одиниця.

Схема підструктур для задання лінійних обмежень, наведених в рівнянні (3.15) та рівнянні (3.16), представлена на рисунку 3.1.

Схема підструктур для задання лінійних обмежень, наведених в рівнянні (3.17) та рівнянні (3.18), представлена на рисунку 3.2.

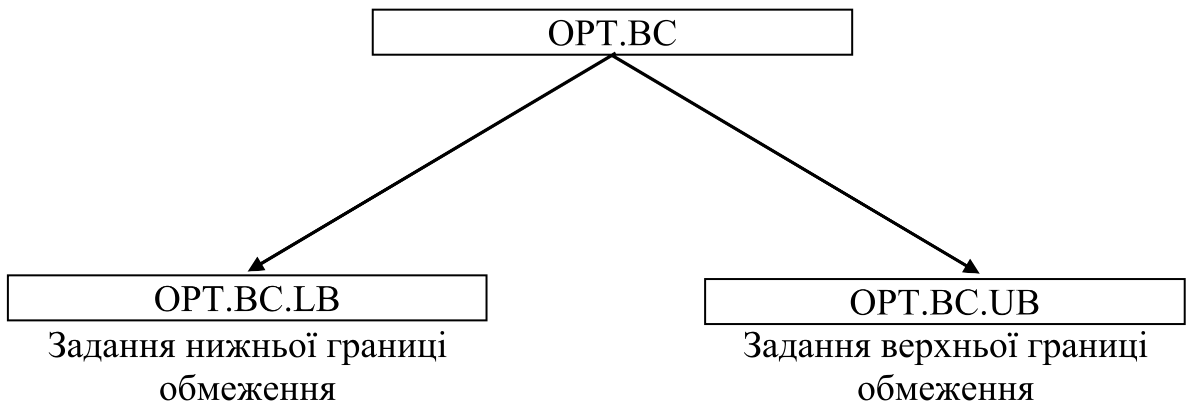

Рисунок 3.2- Схема підструктур для задання лінійних обмежень

Увага ! Для задання лінійних обмежень, наведених в рівнянні (3.17) та рівнянні (3.18) використовуються інші підструктури (див. рис. 3.2), в порівнянні з лінійними обмеженнями, наведеними в рівнянні (3.15) та рівнянні (3.16).

Записати в підструктуру з ім'ям **opt.lc.A** ліву частину лінійного обмеження, тобто ліву частину рівняння (3.12), у вигляді:

$$\mathrm{opt.lc.A} = \begin{bmatrix} 5000 & 2000 \\ 8 & 4 \end{bmatrix};$$

В лівій частині виразу записується підструктура з ім'ям **opt.lc.A**, яка використовується для запису до неї матриці **A**, що представляє собою матрицю, яка містить числові коефіцієнти при незалежних аргументах **x** та **y** лівої частини рівняння (3.15) та рівняння (3.16).

У докладному вигляді рівняння (3.15) та рівняння (3.16) можна записати наступним чином:

$$+5000 \cdot \mathrm{x} + 2000 \cdot \mathrm{y} \le +20000 \qquad (3.19)$$

$$+8 \cdot \mathrm{x} + 4 \cdot \mathrm{y} \le +38 \qquad (3.20)$$

Як видно з рівняння (3.19), числовий коефіцієнт при незалежному аргументі **x** дорівнює “+5000”, числовий коефіцієнт при незалежному аргументі **y** дорівнює “+2000”.

Як видно з рівняння (3.20), числовий коефіцієнт при незалежному аргументі **x** дорівнює “+8”, числовий коефіцієнт при незалежному аргументі **y** дорівнює “+4”.

Увага ! Знак “плюс”, на відміну від знаку “мінус” при запису до матриці **A** числових коефіцієнтів при незалежних аргументах, не ставиться.

Числові коефіцієнти при запису їх до матриці **A** розділяються пробілом.

Оскільки в даній задачі на цільову функцію накладено два лінійних обмеження, кожне з яких містить два незалежні аргументи(**x** та **y**), матриця **A** містить два рядка та два стовпчика.

---

11. Двома рядками до певних підструктур m-файлу потрібно

записати праві від знаку “≤” частини обмеження (3.19) та обмеження (3.20), тобто цифри “+20000” та “+38”.

Для запису правої частини обмеження (3.19) та обмеження (3.20) потрібно скористатись підструктурами **opt.lc.lb** та **opt.lc.ub**.

Підструктури **opt.lc.lb** та **opt.lc.ub** матимуть наступний вигляд:

opt.lc.lb = [-inf -inf];

opt.lc.ub = [20000 38];

Увага ! Нижньою границею обмеження (3.15) та обмеження (3.16) є $-\infty$. В програмному продукті COMSOL MULTIPHYSICS® існує вбудована функція **inf** (від англійського слова “infinity” - нескінченність).

Підструктури **opt.lc.lb** та **opt.lc.ub** представляють собою вектори-рядки, що містять праві частини обмежень (в даній задачі це праві частини обмеження (3.15) та обмеження (3.16).

Сформулюємо наступне правило: кількість стовпчиків вектора-рядка підструктур **opt.lc.lb** та **opt.lc.ub** дорівнює кількості лінійних обмежень, які накладено на цільову функцію.

Числові значення у векторах-рядках відокремлюються пробілом.

---

12. Двома рядками до підструктури **opt.bc** потрібно записати обмеження (3.17) та обмеження (3.18).

Як видно з форми запису підструктури **opt.bc**, її ім’я складається з наступних складових:

opt- **opt**imization (оптимізація). Ця складова імені записується першою та міститься в іменах всіх підструктур, які будуть зустрічатись в подальшому при розгляді алгоритму розв’язання поточної та наступних задач оптимізації;

bc- **b**oundary **c**onstraints (граничні обмеження). Друга (**bc**) складова імені цієї підструктури вказує на те, що підструктура з ім’ям **opt.bc** використовується для запису до неї обмежень, пов’язаних з незалежним аргументом (або декількома незалежними аргументами) цільової функції.

Увага ! Складові імені підструктури відокремлюються одне від одного за допомогою крапки. Присвоювати довільні імена складовим підструктур неможна.

Тобто, наприклад, першій складовій всіх підструктур потрібно присвоювати тільки ім'я **opt**, оскільки COMSOL® Optimization Lab сприймає та працює лише з підструктурами, які мають певні, зрозумілі для програмного продукту, сталі імена. Те ж саме стосується складових імені підструктури **opt.lc** тощо.

Оскільки в лівій від знаку "≥" частині обмеження (3.17) та обмеження (3.18) знаходяться лише незалежні аргументи **x** та **y** немає потреби записувати їх до жодної з підструктур.

Натомість, для запису правої частини обмеження (3.17) та обмеження (3.18) потрібно скористатись підструктурами **opt.bc.lb** та **opt.bc.ub**.

Як видно з форми запису підструктури **opt.bc**, її ім'я складається зі складових **lb** та **ub**:

lb- **l**ower **b**oundary (нижня границя). Ця складова імені підструктури **opt.bc.lb** вказує на те, що вона використовується для запису нижньої границі незалежного аргументу.

ub- **u**pper **b**oundary (верхня границя). Ця складова імені підструктури **opt.lc.ub** вказує на те, що вона використовується для запису верхньої границі незалежного аргументу (або декількох незалежних аргументів) цільової функції.

Увага ! Складові імені підструктури відокремлюються одне від одного за допомогою крапки. Присвоювати довільні імена складовим підструктур неможна. Тобто, наприклад, першій складовій всіх підструктур потрібно присвоювати тільки ім'я **opt**, оскільки COMSOL® Optimization Lab сприймає та працює лише з підструктурами, які мають певні, зрозумілі для програмного продукту, сталі імена.

Таким чином, підструктури **opt.bc.lb** та **opt.bc.ub** матимуть наступний вигляд:

opt.bc.lb = [0 0];

opt.bc.ub = [inf inf];

Увага ! Підструктури **opt.bc.lb** та **opt.bc.ub** являють собою вектори-рядки, що містять праві частини обмежень (в даній задачі це праві частини обмеження (3.17) та обмеження (3.18).

Сформулюємо наступне правило: кількість стовпчиків вектора-рядка

підструктур **opt.bc.lb** та **opt.bc.ub** дорівнює кількості обмежень, які накладено на незалежні аргументи цільової функції.

Числові значення у векторах-рядках відокремлюються пробілом.

---

13. Для виконання наступних кроків, а саме:

- запису в підструктуру з ім’ям **opt.init.x** довільних значень незалежних аргументів **x** та **y**, починаючи з яких COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції;

- задання команди, за допомогою якої буде виконуватись пошук абсолютного мінімуму цільової функції;

- задання команди добування значень незалежних аргументів **x** та **y** з підструктури **opt.sol.x**;

- задання команди для добування значення абсолютного мінімуму функції з підструктури **opt.sol.eval.f**;

- контролю за успішністю розв’язання задачі, можна скористатись пунктами 10…15 **прикладу 2.1.**

Увага ! На відміну від **прикладу 2.1** при запису в підструктуру з ім’ям **opt.init.x** довільних значень незалежних аргументів **x** та **y** в квадратних дужках потрібно через пробіл записати не одне, а два числа, що відповідає кількості незалежних аргументів:

opt.init.x = [0 0];

Кроки, які перелічені в пункті 13 даної задачі, записуються у вигляді рядків програми наступним чином:

```
opt.init.x = [0 0];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
x_min_1 = x_min(1,1);
x_min_2 = x_min(1,2);
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
```

```
if opt.sol.exit == 0;
optymizatsiya = 'nevdala_optymizatsiya';
end
```

14. Для розв'язання задачі в командному рядку програмного продукту COMSOL® SCRIPT™ потрібно записати наступне:

run optymizatsiya_function_1

Увага ! Після слова "run" ставиться пробіл та записується ім'я m-файлу. В даній задачі файл має ім'я optymizatsiya_function_1.

---

15. Натиснути **Enter**.

В цілому, програма для відшукування мінімуму цільової функції матиме наступний вигляд:

```
clear
clc
opt.obj.f = 'function_1';
opt.obj.form = 'nlin';
```

$$\text{opt.lc.A} = \begin{bmatrix} 5000 & 2000 \\ 8 & 4 \end{bmatrix};$$

```
opt.lc.lb = [-inf -inf];
opt.lc.ub = [20000 38];
opt.bc.lb = [0 0];
opt.bc.ub = [inf inf];
opt.init.x = [0 0];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
x_min_1 = x_min(1,1);
x_min_2 = x_min(1,2);
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
```

```
if opt.sol.exit == 0;
optymizatsiya = 'nevdala_optymizatsiya';
end
```

Результати розв’язання даної задачі наводяться в робочому полі (**Workspace)** у вигляді, наведеному в таблиці 3.2.

Таблиця 3.2- Результати розв’язання задачі

Name	Value
algorythm_optymizatsiyi	‘optnm’
f_min	29
opt	$\left[1\times1\,\text{struct}\right]$
optymizatsiya	‘uspishna optymizatsiya’
x_min	[2 5]
x_min_1	2
x_min_2	5

Як видно з табл. 3.2 результати розв’язання задачі не змінюються в порівнянні з **прикладом 3.3**, що вказує на правильне використання файлу-функції та m-файлу.

3.3 Приклади розв’язання теплоенергетичних задач умовної оптимізації

Приклад 3.5.

Прокатний стан обслуговує дві однакові методичні печі. Максимальна продуктивність кожної із них складає $80\,\text{т/ч}$, а мінімальна продуктивність кожної із них складає $10\,\text{т/ч}$. При роботі печей з максимальною продуктивністю собівартість нагріву металу складає $4\,\text{грн/т}$, а при роботі печей с мінімальною продуктивністю собівартість нагріву металу складає $14\,\text{грн/т}$.

Необхідно так організувати роботу двох методичних печей, щоб

одночасно забезпечити продуктивність стану $120\,т/ч$ при мінімальній собівартості нагріву металу.

Позначимо продуктивність першої печі через **x**, а продуктивність другої печі через **y**. Критерієм оптимізації буде собівартість нагріву металу.

Математичний запис цільової функції, мінімальне значення якої потрібно відшукати, має вигляд:

$$f = (15{,}43 - 0{,}143 \cdot x) \cdot \frac{x}{120} + (15{,}43 - 0{,}143 \cdot y) \cdot \frac{y}{120} \qquad (3.21)$$

Обмеження, що накладаються на функцію оптимізації, наступні:

$$x + y = 120 \qquad (3.22)$$

$$x \geq 10 \qquad (3.23)$$

$$y \geq 10 \qquad (3.24)$$

$$x \leq 80 \qquad (3.25)$$

$$y \leq 80 \qquad (3.26)$$

Відшукаємо розв'язок попередньої задачі за допомогою модуля оптимізації COMSOL® Optimization Lab програмного продукту COMSOL MULTIPHYSICS®.

Увага ! Дана задача відноситься до задач нелінійної оптимізації, оскільки цільова функція (3.21) є квадратичною, незважаючи на те, що обмеження (3.22)-(3.26) є лінійними.

Для розв'язання задачі потрібно виконати наступні кроки:

1. Відкрити COMSOL® SCRIPT™.

2. Для запису в підструктуру з ім'ям **opt.obj.f** функції, абсолютний мінімум якої потрібно визначити, необхідно створити окремий файл-функцію.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

Для створення файлу-функції необхідно створити m-файл. Детальний опис послідовності створення m-файлу наведений в **прикладі 2.1**.

Ім'я файлу-функції повинно співпадати з ім'ям цільової функції, яка записується до файлу-функції. Наприклад, ім'ям цільової функції, а разом і файлу функції, буде ім'я **function_1**.

3. Потрібно зберегти файл-функцію. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім'я файлу-функції, тобто **function_1**.

4. В порожньому файлі-функції перший рядок починається зі слова **function**. У випадку, коли m-файл розпочинається зі слова **function**, такий файл сприймається програмними продуктами COMSOL MULTIPHYSICS® та MATLAB® в якості файлу-функції.

Після слова **function** в першому рядку файлу-функції записується позначення цільової функції (в даному прикладі ця функція позначена через літеру **f**), а після знаку "=" вказується ім'я функції (в даному прикладі це **function_1**) після якого в дужках записується ім'я незалежного аргументу, або декількох незалежних аргументів, від яких залежить функція (в даному прикладі згідно з рівнянням (3.21) незалежними аргументами є **x** та **y**).

Таким чином, перший рядок файлу-функції має наступний вигляд:

function f = function_1(x)

Увага ! Необхідно зазначити, що ім'я всіх незалежних аргументів (**x**, **y** тощо) при запису математичного виразу функції у файлі-функції може бути довільним (в даній задачі це ім'я **x**), проте однаковим для всіх незалежних аргументів. Відмінність одного незалежного аргументу від іншого полягає у додаванні до його імені індексу у дужках (1, 2 тощо).

Математичний вираз функції записується у дещо іншому вигляді, ніж це наведено у рівнянні (3.21). Так, незалежний аргумент **x** записується як **x(1)**, а незалежний аргумент **y** записується як **x(2)**.

Тоді математичний вираз функції, який записується другим і

останнім рядком до файлу-функції, матиме наступний вигляд:

f = (15.43-0.143*x(1))*(x(1)/120)+(15.43-0.143*x(2))*(x(2)/120)

В цілому, файл-функція має наступний вигляд:

function f = function_1(x)

f = (15.43-0.143*x(1))*(x(1)/120)+(15.43-0.143*x(2))*(x(2)/120)

Редагування файлу-функції завершено. Його можна закрити, або залишити відкритим.

5. Створити m-файл. Для цього необхідно зайти в пункт меню **File** > **New editor.** В полі вікна COMSOL Desktop® відкривається порожній m-файл.

Потрібно зберегти файл, попередньо присвоївши йому ім'я. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім'я файлу, наприклад, **optymizatsiya_function_1**.

Увага ! Адресом збереження файлу повинна бути тека, в яку було встановлено (інстальовано) програмний продукт COMSOL MULTIPHYSICS®.

---

6. Написати першим рядком робочого поля m-файлу вбудовану команду **clear**.

7. Написати наступним рядком робочого поля m-файлу вбудовану команду **clc**.

8. Записати цільову функцію в підструктуру **opt.obj.f**.

Для цього підструктуру **opt.obj.f** записують у наступному вигляді:

opt.obj.f = 'function_1';

Увага! Після запису імені підструктури та знаку "=" записується назва файлу-функції. В даному випадку файл-функція має назву function_1. Назва файлу-функції виділяється одинарними лапками.

---

9. Наступним рядком в підструктуру з ім'ям **opt.obj.form** записується тип цільової функції (лінійна, квадратична, нелінійна), у відповідності до якого обирається тип розв'язання оптимізаційної задачі.

Цільова функція, тобто рівняння (3.21), є квадратичною, оскільки максимальним степенем піднесення незалежних аргументів **x** та **y** є квадрат.

Увага ! Незважаючи на те, що цільова функція є лінійною бажано для усих типів цільової функції обирати нелінійний тип розв'язання оптимізаційної задачі і записувати це після знаку "=" або знаків "≤" та "≥" в підструктуру з ім'ям **opt.obj.form** наступним чином:

opt.obj.form = 'nlin';

Слово **nlin**, яке виокремлюється з обох боків одинарними лапками, є скороченням від слова nonlinear (нелінійний).

---

10. Наступними рядками до m-файлу потрібно записати обмеження (рівняння (3.22)-(3.26)), які накладено на цільову функцію.

Виконаємо аналіз математичних виразів обмежень:

1) Обмеження задано у вигляді нерівностей та рівностей.

2) Обмеження є лінійними, оскільки максимальним степенем піднесення незалежних аргументів **x** та **y** є одиниця.

Схема підструктур для задання лінійного обмеження, наведеного в рівнянні (3.22), представлена на рисунку 3.1.

Схема підструктур для задання лінійних обмежень, наведених в рівняннях (3.23)-(3.26), представлена на рисунку 3.2.

Записати в підструктуру з ім'ям **opt.lc.A** ліву частину лінійного обмеження, тобто ліву частину рівняння (3.22), у вигляді:

opt.lc.A = [1 1];

В лівій частині виразу записується підструктура з ім'ям **opt.lc.A**, яка використовується для запису до неї матриці **A**, що представляє собою вектор-рядок, що містить з числові коефіцієнти при незалежних аргументах **x** та **y** лівої частини рівняння (3.22).

У докладному вигляді рівняння (3.22) можна записати наступним чином:

$$+1 \cdot x + 1 \cdot y = +120 \qquad (3.27)$$

Як видно з рівняння (3.27), числовий коефіцієнт при незалежному аргументі **x** дорівнює "+1", числовий коефіцієнт при незалежному

аргументі **y** також дорівнює “+1”.

11. Наступними двома рядками до певних підструктур m-файлу потрібно записати праву від знаку “=”частину обмеження $x + y = 120$, тобто цифру “+120”.

Для запису правої частини рівняння (3.22) потрібно скористатись підструктурами **opt.lc.lb** та **opt.lc.ub**.

Підструктури **opt.lc.lb** та **opt.lc.ub** матимуть наступний вигляд:

opt.lc.lb = 120;

opt.lc.ub = 120;

Увага ! Оскільки в даній задачі на цільову функцію накладено обмеження у вигляді рівності, нижня границя обмеження дорівнює верхній частині обмеження.

---

12. Двома рядками до підструктури **opt.bc** потрібно записати обмеження (3.23)-(3.26).

Увага ! Оскільки в лівій від знаків “$\geq$” та “$\leq$” частині обмежень (3.23)-(3.26) знаходяться лише незалежні аргументи **x** та **y** немає потреби записувати їх до жодної з підструктур.

Натомість, для запису правої частини обмежень (3.23)-(3.26) потрібно скористатись підструктурами **opt.bc.lb** та **opt.bc.ub**.

підструктури **opt.bc.lb** та **opt.bc.ub** матимуть наступний вигляд:

opt.bc.lb = [10 10];

opt.bc.ub = [80 80];

Увага ! Підструктури **opt.bc.lb** та **opt.bc.ub** представляють собою вектори-рядки, що містять праві частини обмежень (в даній задачі це праві частини обмежень (3.23)-(3.26).

Числові значення у векторах-рядках відокремлюються пробілом.

---

13. Для виконання наступних кроків, а саме:

- запису в підструктуру з ім’ям **opt.init.x** довільних значень незалежних аргументів **x** та **y**, починаючи з яких COMSOL® Optimization Lab розпочинає пошук значення абсолютного мінімуму функції;

- задання команди, за допомогою якої буде виконуватись пошук

абсолютного мінімуму цільової функції;

- задання команди добування значень незалежних аргументів **x** та **y** з підструктури **opt.sol.x**;

- задання команди для добування значення абсолютного мінімуму функції з підструктури **opt.sol.eval.f**;

- контролю за успішністю розв'язання задачі, можна скористатись пунктами 10…15 **прикладу 2.1.**

Увага ! На відміну від **прикладу 2.1** при запису в підструктуру з ім'ям **opt.init.x** довільних значень незалежних аргументів **x** та **y** в квадратних дужках потрібно через пробіл записати не одне, а два числа, що відповідає кількості незалежних аргументів:

opt.init.x = [0 0];

Кроки, які перелічені в пункті 13 даної задачі, записуються у вигляді рядків програми наступним чином:

```
opt.init.x = [0 0];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
x_min_1 = x_min(1,1);
x_min_2 = x_min(1,2);
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
if opt.sol.exit == 0;
optymizatsiya = 'nevdala_optymizatsiya';
end
```

14. Для розв'язання задачі в командному рядку програмного продукту COMSOL® SCRIPT™ потрібно записати наступне:

run optymizatsiya_function_1

Увага ! Після слова "run" ставиться пробіл та записується ім'я m-файлу. В

даній задачі файл має ім’я optymizatsiya_function_1.

15. Натиснути **Enter**.

В цілому, програма для відшукування мінімуму цільової функції матиме наступний вигляд:

```
clear
clc
opt.obj.f = 'function_1';
opt.obj.form = 'nlin';
opt.lc.A = [1 1];
opt.lc.lb = 120;
opt.lc.ub = 120;
opt.bc.lb = [10 10];
opt.bc.ub = [80 80];
opt.init.x = [0 0];
opt.sol = optnm(opt);
x_min = opt.sol.x;
f_min = opt.sol.eval.f;
x_min_1 = x_min(1,1);
x_min_2 = x_min(1,2);
if opt.sol.exit == 1;
optymizatsiya = 'uspishna_optymizatsiya';
end
if opt.sol.exit == 0;
optymizatsiya = 'nevdala_optymizatsiya';
end
```

Результати розв’язання даної задачі наводяться в робочому полі (**Workspace)** у вигляді, наведеному в таблиці 3.3.

Приклади задач умовної оптимізації для самостійного розв’язання наведені в додатку Б.

Таблиця 3.3- Результати розв’язання задачі

Name	Value
algorythm_optymizatsiyi	‘optnm’
f_min	5.67
opt	$[1 \times 1\,\text{struct}]$
optymizatsiya	‘uspishna optymizatsiya’

4 ПРОГРАМУВАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ, СФОРМУЛЬОВАНИХ У ВИГЛЯДІ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

4.1 Програмування оптимізаційних задач промислової теплоенергетики, сформульованих у вигляді диференціальних рівнянь

Приклад 4.1.

Розглянемо трубчатий реактор, в якому проходить реакція:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

Нехай обидві реакції від “A” до “B” та від “B” до “C” є реакціями першого порядку. Позначимо концентрації реагентів “A” та “B” через c_1 та c_2. Тоді для них справедливі наступні рівняння рівноваги:

$$\frac{dc_1}{d\tau} = -k_1 \cdot c_1 \tag{4.1}$$

$$\frac{dc_2}{d\tau} = k_1 \cdot c_1 - k_2 \cdot c_2 \tag{4.2}$$

Для більш наочного відображення розв’язку задачі в середовищі COMSOL MULTIPHYSICS® зробимо заміну похідних за часом в рівнянні (4.1) та рівнянні (4.2) похідними за просторовою координатою, наприклад, абсцисою **x**. Рівняння (4.1) та рівняння (4.2) матимуть наступний вигляд:

$$\frac{dc_1}{dx} = -k_1 \cdot c_1 \tag{4.3}$$

$$\frac{dc_2}{dx} = k_1 \cdot c_1 - k_2 \cdot c_2 \tag{4.4}$$

де x - продольна координата реактора, віднесена до швидкості

потоку реагентів v, м;

k_1, k_2 - константи швидкості реакції, $\frac{1}{c}$.

Константи швидкості реакції k_1 та k_2 залежать від температури T у реакторі наступним чином:

$$k_1 = k_{10} \cdot e^{\frac{-E_1}{R \cdot T}} \tag{4.5}$$

$$k_2 = k_{20} \cdot e^{\frac{-E_2}{R \cdot T}} \tag{4.6}$$

де k_{10}, k_{20}, E_1, E_2 та R - константи.

Задача оптимізації полягає у тому, щоб при заданій довжині **L** реактора знайти такий розподіл температур по довжині реактора $T = f(x)$, при якому на виході з реактора продукт “В” матиме максимальну концентрацію, тобто c_2 набуває максимального значення.

Граничні умови для даної задачі представлені в таблиці 4.1.

Таблиця 4.1- Граничні умови до задачі

На вході в реактор	$c_1 = c_{1_0}$ $c_2 = c_{2_0}$
На виході з реактора	$\frac{dc_1}{dx} = 0$ $\frac{dc_2}{dx} = 0$

В таблиці 4.1 c_{10} и c_{20} - відповідно концентрації продуктів ”А” та ”В” на вході в реактор.

Схема граничних умов наведена на рисунку 4.1.

Числові значення констант для даної задачі представлені в таблиці 4.2.

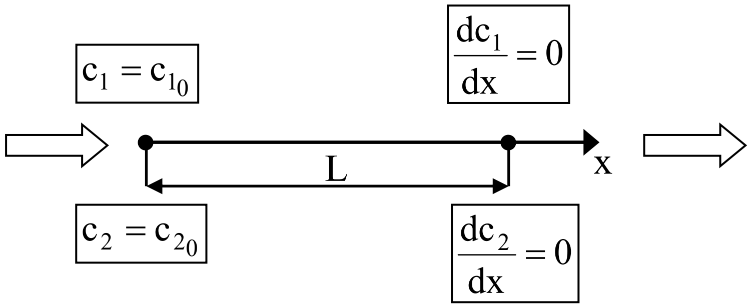

Рисунок 4.1- Схема граничних умов до задачі

Таблиця 4.2- Числові значення констант до задачі

Позначення величини	Значення
Концентрація продукту ”A” на вході в реактор c_{10}, $моль/м^3$	0,00053
Концентрація продукту ”B” на вході у реактор c_{20}, $моль/м^3$	0,00043
Константа k_{10}, 1/с	$0{,}892 \cdot 10^9$
Константа k_{20}, 1/с	$0{,}768 \cdot 10^{16}$
Енергія активації хімічної реакції E_1, Дж/моль	75420
Енергія активації хімічної реакції E_2, Дж/моль	125700
Константа R, Дж/(моль · К)	8,38
Довжина реактора L, м	480
Температура у реакторі (первинне наближення) T_0, К	250

Увага ! Час перебування продукту “A” та продукту “B” в реакторі становить 480с., але, оскільки нестаціонарна задача зведена до стаціонарної внаслідок заміни похідних за часом в рівнянні (4.1) та рівнянні (4.2) похідними за просторовою координатою, довжина реактора чисельно дорівнює часу перебування продукту “A” та продукту “B” в реакторі.

Формулювання задачі, таким чином, завершено і можна розпочати її розв’язання в середовищі COMSOL MULTIPHYSICS®. Для цього потрібно виконати наступні кроки.

1. Відкрити **Model Navigator**.

2. Вибрати в списку **Space Dimension 1D**.

Увага ! Як видно з рівняння (4.3) та рівняння (4.4) задача є одновимірною,

тому в списку, що розгортається потрібно обрати **1D**.

---

3. Натиснути на довгасту кнопку **Multiphysics**, яка знаходиться у правому нижньому кутку меню **Model Navigator**.

4. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **PDE Modes** > **PDE, General form** > **Stationary analysis**.

5. В області **Dependent variables** замість розрахункової функції **u** (яка є такою поза вибором) потрібно написати **c_1** (концентрація продукту ”A”).

6. Натиснути на кнопку **Add** в полі **Multiphysics**.

7. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **PDE Modes** > **PDE, General form** > **Stationary analysis**.

8. В області **Dependent variables** замість розрахункової функції **u** (яка є такою поза вибором) потрібно написати **c_2** (концентрація продукту ”B”).

9. Натиснути на кнопку **Add** в полі **Multiphysics**.

10. Натиснути **OK.**

Виконання перелічених вище дій призводить до формування системи, що складається з двох диференціальних рівнянь відповідно до кількості невідомих величин в рівнянні (4.3) та рівнянні (4.4). В даній задачі невідомими величинами є концентрація продукту ”A” **c_1** та концентрація продукту ”B” **c_2**.

Відкриється графічний інтерфейс користувача (GUI).

Тепер потрібно зберегти файл, попередньо присвоївши йому ім’я. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім’я файлу, наприклад, **reactor_1**.

Для відображення геометричної області, в якій буде відшукуватись розв’язок задачі потрібно:

11. Зайти в пункт меню **Draw** > **Specify Objects** > **Line**.

У вікні в полі **Coordinates** потрібно ввести значення довжини реактора:

x: 0   480 (через пробіл).

12. Натиснути **ОК**.

Задамо константи, які входять до даної задачі.

Для цього потрібно виконати наступні кроки:

13. Зайти в пункт меню **Options** > **Constants**.

Значення коефіцієнтів в закладку **Constants** потрібно ввести так, як показано у вікні таблиці 4.3 за умовами даної програми.

Таблиця 4.3- Вікно присвоєння значень коефіцієнтів

Name	**Expression**	**Value**	**Description**
c_10	0.53[mol/m^3]	$0.53\left[mol/m^3\right]$	
c_20	0.43[mol/m^3]	$0.43\left[mol/m^3\right]$	
k_10	0.892e9[1/s]	8.92e8[1/s]	
k_20	0.768e16[1/s]	7.68e15[1/s]	
E_1	18000[J/mol]	18000[J/mol]	
E_2	30000[J/mol]	30000[J/mol]	
R	2[J/(mol*K)]	2[J/(mol*K)]	

14. Зайти в пункт меню **Multiphysics**. Переконатися, що перемикач встановлено навпроти першого диференціального рівняння (диференціального рівняння для розрахункової функції **c_1**), тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
● 1 PDE, General Form (g) 2 PDE, General Form (g2)

Для задання диференціального рівняння (4.3) потрібно виконати наступні кроки.

15. Зайти в пункт меню **Physics** > **Subdomain Settings**.

16. На закладці **Subdomains** в полі **Subdomain selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

17. На закладці **Coefficients** в полі **PDE coefficients** поза вибором коефіцієнтам **Г**, **F**, $\mathbf{e_a}$ и $\mathbf{d_a}$, як видно з таблиці 4.4, присвоєні наступні значення.

Таблиця 4.4- Вікно присвоєння значень коефіцієнтів поза вибором

Coefficient	Value / Expression
Г	$-c\_1x$
F	1
e_a	0
d_a	1

Значення коефіцієнтів в закладку **Coefficients** для рівняння (4.3) потрібно ввести так, як показано в таблиці 4.5:

Таблиця 4.5- Вікно присвоєння значень коефіцієнтів

Coefficient	Value / Expression
Г	0
F	$-k\_10*c\_1*\exp(-E\_1/(R*T))-c\_1x$
e_a	0
d_a	0

Увага ! Синтаксис задання звичайних диференціальних рівнянь, якими є рівняння (4.3) та рівняння (4.4) детально описано Жовтоногою в літературі [4].

18. Натиснути **ОК**.

Для задання граничних умов рівняння (4.3) потрібно виконати наступні кроки:

19. Зайти в пункт меню **Physics** > **Boundary Settings**.

20. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **1**. На закладці **Coefficients** в полі **Boundary conditions** встановити перемикач на граничну умову Діріхле (**Dirichlet boundary condition)**. Значення коефіцієнтів **G** та **R** ввести у відповідності до таблиці 4.6.

Таблиця 4.6- Вікно присвоєння значень коефіцієнтів

Coefficient	Value / Expression
G	0
R	c_10-c_1

Увага ! Нагадаємо, що **c_1** становить собою розрахункову величину, тобто розрахункову концентрацію продукту “А” по довжині реактора, а **c_10** становить собою концентрацію продукту ”А” на вході в реактор. Значення цієї величини вже задано в меню **Options** > **Constants**.

21. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. На закладці **Coefficients** в полі **Boundary conditions** встановити перемикач на граничну умову Неймана (**Neumann boundary condition)**.

Коефіцієнт **G** залишити без зміни.

22. Натиснути **ОК**.

Виконання цього кроку завершує задання рівняння (4.3), а також граничних умов до нього в COMSOL MULTIPHYSICS®. Тепер можна перейти до введення в COMSOL MULTIPHYSICS® рівняння (4.4).

Для цього потрібно зайти в пункт меню **Multiphysics**. Встановити перемикач навпроти другого диференціального рівняння (диференціального рівняння для розрахункової функції **c_2**), тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
1 PDE, General Form (g) ● 2 PDE, General Form (g2)

Для задання диференціального рівняння (4.4) потрібно виконати наступні кроки.

23. Зайти в пункт меню **Physics** > **Subdomain Settings**.

24. На закладці **Subdomains** в полі **Subdomain selection** натиснути на цифру **1.** Після цього поле з цифрою **1** зафарбується в синій колір.

25. На закладці **Coefficients** в полі **PDE coefficients** поза вибором коефіцієнтам **Г**, **F**, $\mathbf{e_a}$ и $\mathbf{d_a}$ (табл. 4.7) присвоєні наступні значення.

Таблиця 4.7- Вікно присвоєння значень коефіцієнтів поза вибором

Coefficient	**Value / Expression**
Г	$-c\_1x$
F	1
e_a	0
d_a	1

Значення коефіцієнтів в закладку **Coefficients** для рівняння (4.4) потрібно ввести так, як показано в таблиці 4.8.

26. Натиснути **ОК**.

Для задання граничних умов рівняння (4.4) потрібно виконати наступні кроки:

27. Зайти в пункт меню **Physics** > **Boundary Settings**.

Таблиця 4.8- Вікно присвоєння значень коефіцієнтів

Coefficient	Value / Expression
Г	0
F	$-k\_10 * c\_1 * \exp(-E\_1/(R * T)) -$ $-k\_20 * c\_2 * \exp(-E\_1/(R * T)) - c\_2x$
e_a	0
d_a	0

28. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **1**. На закладці **Coefficients** в полі **Boundary conditions** встановити перемикач на граничну умову Діріхле (**Dirichlet boundary condition)**. Значення коефіцієнтів **G** та **R** ввести у відповідності до таблиці 4.9.

Таблиця 4.9- Вікно присвоєння значень коефіцієнтів

Coefficient	Value / Expression
G	0
R	$c\_20 - c\_2$

Увага ! Нагадаємо, що **c_2** становить собою розрахункову величину, тобто розрахункову концентрацію продукту “B” по довжині реактора, а **c_20** становить собою концентрацію продукту ”B” на вході в реактор. Значення цієї величини вже задано в меню **Options** > **Constants**.

29. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. На закладці **Coefficients** в полі **Boundary conditions** встановити перемикач на граничну умову Неймана (**Neumann boundary condition)**. Коефіцієнт **G** залишити без зміни.

30. Натиснути **OK**.

Виконанням цієї дії завершується задання рівняння (4.4), а також граничних умов до нього в COMSOL MULTIPHYSICS®.

Наступним кроком є задання цільової функції та умов, тобто обмежень, які накладено на параметри, що описують дану задачу.

Для цього потрібно виконати наступні кроки.

31. Зайти в меню **Multiphysics** та відкрити пункт меню **Model Navigator**.

32. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **Optimization and Sensitivity** > **Optimization**.

Увага ! В області **Dependent variables** поле з ім’ям розрахункової функції поза

вибором є неактивним. Для модуля **Optimization** спосіб задання самої розрахункової функції, тобто цільової функції та її імені дещо відрізняються від способу задання розрахункової функції диференціальних рівнянь та їх імен. Спосіб задання самої цільової функції та її імені для модуля **Optimization** будуть розглянуті в рамках цієї задачі згодом.

---

33. Натиснути на кнопку **Add** в полі **Multiphysics**.

34. Натиснути **OK.**

Виконання перелічених вище дій призводить до формування системи, що складається з двох диференціальних рівнянь, відповідно до кількості невідомих величин, та цільової функції.

Для задання цільової функції та інших параметрів задачі оптимізації необхідно виконати наступні кроки.

35. Зайти в пункт меню **Multiphysics**. Переконатися, що перемикач встановлено навпроти третього диференціального рівняння, тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
1 PDE, General Form (g) 2 PDE, General Form (g2) ● 3 Optimization (opt)

Задання цільової функції та інших параметрів оптимізації здійснюється на основі таких міркувань.

Оскільки задача оптимізації полягає у тому, щоб при заданій довжині **L** реактора знайти такий розподіл температур по довжині реактора $T = f(x)$, при якому на виході з реактора продукт "B" матиме максимальну концентрацію, тобто розрахункова функція c_2 набуватиме максимального значення, необхідно певним чином задати цільову функцію саме на границі **2** розрахункової області.

Увага ! Сформулюємо наступне правило. Задавати цільову функцію завжди потрібно в тій частині розрахункової області, в якій необхідно відшукати максимум або мінімум цільової функції.

Для задання цільової функції на границі **2** розрахункової області необхідно виконати наступні кроки.

36. Зайти в пункт меню **Physics** > **Boundary Settings**.

37. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. Після цього поле з цифрою **2** зафарбується в синій колір.

На закладці **Objective** (цільова функція) в полі **Objective contribution** (вираз для цільової функції) поза вибором значення параметру $\mathbf{q_0}$ у вічці **Point contribution** (внесок для точки) є відсутнім.

У вічко **Point contribution** потрібно записати вираз або розрахункову величину, мінімальне значення якого (якої) потрібно відшукати.

Оскільки в рамках розв'язання даної задачі необхідно забезпечити максимальне значення концентрації продукту "B", тобто необхідно забезпечити максимальне значення розрахункової функції c_2 на границі **2**, вічко **Point contribution** необхідно оформити так, як показано в таблиці 4.10.

Таблиця 4.10- Вікно присвоєння значень коефіцієнта

q_0	$-c\_2$	Point contribution

Увага ! Задавати цільову функцію завжди потрібно в тій частині розрахункової області, в якій необхідно відшукати максимум або мінімум цільової функції.

Перед позначенням розрахункової функції c_2 стоїть знак "мінус", оскільки за умовами даної задачі необхідно відшукати максимальне значення цільової функції, в той час, як програмний продукт завжди відшукуватиме мінімальне значення цільової функції. Таким чином, для відшукування максимального значення цільової функції перед її виразом необхідно поставити знак "мінус".

В разі, якщо необхідно відшукати мінімальне значення цільової знак "мінус" ставити не потрібно.

38. Натиснути **ОК.**

Далі необхідно задати:

- параметр або декілька параметрів, варіювання якими дозволить досягти мінімального або максимального значення цільової функції;

- граничні значення варіювання значень параметра або декількох

параметрів;

- межі варіювання (вся розрахункова область, границя області, точка області) параметра або декількох параметрів.

Відповідно до умов даної задачі, параметром, варіювання яким дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, є температура **T** в реакторі.

Для задання цього параметру, а також його граничних значень та меж варіювання необхідно виконати наступні кроки.

39. Зайти в пункт меню **Physics** > **Scalar Settings**.

На закладці **Variables** (змінні) в полі **Variable settings** (налаштування змінної) поза вибором значення параметрів **Variable** (змінна) та **Init** є відсутніми.

У вічко **Variable** потрібно записати позначення параметра, варіювання якими дозволить досягти мінімального (в даній задачі максимального) значення цільової функції. Відповідно до умов даної задачі, параметром, варіювання яким дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, є температура **T** в реакторі.

У вічко **Init** потрібно записати початкове значення температури **T** в реакторі.

Увага! Початкове та, водночас, довільне значення температури **T** в реакторі впливає лише на швидкість відшукання абсолютного мінімуму або максимуму цільової функції і не впливає на точність відшукання мінімуму або максимуму цільової функції.

В даній задачі в якості початкового та довільного значення температури **T** в реакторі задається 350К.

Таким чином, поле **Variable settings** закладки **Variables** має виглядати так, як показано в таблиці 4.11.

Таблиця 4.11- Вікно присвоєння значень коефіцієнтів

Variable	**Init**
T	350

Місце задання меж варіювання температури **T** в реакторі

обумовлено тим, що цей параметр не є розрахунковою функцією, до яких в даній задачі належать концентрації продуктів “A” та “B”.

Увага! Сформулюємо наступне правило. В разі, якщо параметр або декілька параметрів, варіювання якими дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, не є розрахунковими функціями, задання параметра або декількох параметрів, а також його (їх) граничних значень та меж варіювання здійснюється пункті меню **Physics** > **Scalar Settings**.

40. Натиснути **OK**.

Задання всіх умов, що описують задачу, завершено.

41. Для генерації скінченнноелементної сітки зайти в пункт меню **Mesh** > **Free Mesh Parameters** > **Remesh.**

42. Натиснути **OK**.

Для розв’язання задачі необхідно виконати наступні кроки.

43. Зайти в пункт меню **Solve** > **Solver Parameters**. Активізувати опції **Optimization / Sensitivity** (Оптимізація / Чуттєвість) та **Plot while solving** (Зобразити під час розв’язання).

44. На закладці **Optimization / Sensitivity** в меню, що розгортається обрати тип аналізу **Analysis: Optimization**.

45. В поле **Variables** потрібно записати позначення параметра, варіювання якими дозволить досягти мінімального (в даній задачі максимального) значення цільової функції. Відповідно до умов даної задачі, параметром, варіювання яким дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, є температура **T** в реакторі.

46. Натиснути **OK**.

47. Для розв’язання задачі зайти в пункт меню **Solve** > **Solve Problem**.

Отриманий розв’язок потрібно порівняти з розв’язком, наведеним Хофером в літературі [5].

Для закріплювання вивченого матеріалу розглянемо аналогічну

задачу.

В літературі [6] наводиться приклад.

Приклад 4.2.

Декілька послідовних хімічних реакцій мають місце в трубчастому хімічному реакторі. Для контролю за швидкістю хімічних реакцій застосовується охолоджувальна оболонка. Схема трубчастого хімічного реактора наведена на рисунку 4.2.

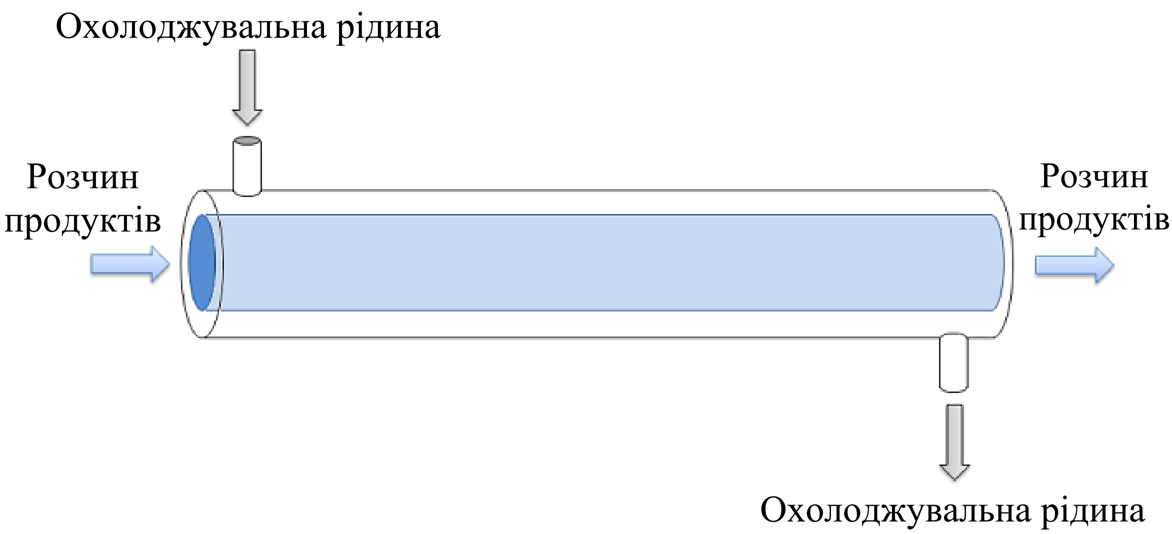

Рисунок 4.2- Схема трубчастого хімічного реактора

Дві послідовні хімічні реакції відбуваються в трубчастому хімічному реакторі:

$$A \xrightarrow{k_1} B$$

$$B \xrightarrow{k_2} C$$

Задача оптимізації полягає у тому, щоб при заданій довжині **L** реактора знайти таке значення температури охолоджувальної рідини **T_okhladitel** на вході до охолоджувальної оболонки, при якому на виході з реактора продукт "B" матиме максимальну концентрацію, тобто c_B набуватиме максимального значення.

Для спрощення задачі замість двовимірної розглядатиметься одновимірна задача, тобто задача з розподілом усих параметрів вздовж осі абсцис, тобто осі **x**.

Числові значення констант для даної задачі представлені в таблиці 4.12.

Таблиця 4.12- Числові значення констант до задачі

Позначення величини	Значення
Концентрація продукту ”A” на вході в реактор c_{A0}, $моль/м^3$	700
Константа A_1, $1/c$	$1{,}6 \cdot 10^8$
Константа A_2, $1/c$	$1{,}0 \cdot 10^{15}$
Енергія активації хімічної реакції E_1, $Дж/моль$	75000
Енергія активації хімічної реакції E_2, $Дж/моль$	125000
Константа R, $Дж/(моль \cdot К)$	8,314
Довжина реактора L, м	2
Швидкість руху розчину продуктів в реакторі $u_{розчин}$, $м/с$	0,0042
Константа H_1, $Дж/моль$	200000
Константа H_2, $Дж/моль$	100000
Коефіцієнт дифузії D, $м^2/с$	$1{,}0 \cdot 10^{-8}$
Об’ємний коефіцієнт тепловіддачі UA, $Вт/(м^3 \cdot К)$	10000
Початкова температура розчину на вході в реактор, К	400
Швидкість руху охолоджувальної рідини в охолоджувальній оболонці $u_{охолоджувач}$, $м/с$	0,001
Температура розчину на вході до трубчастого хімічного реактора $T_{розчин0}$, К	400
Температура охолоджувальної рідини $T_{охолоджувач0}$ (первинне наближення), К	350

Процеси, які відбуваються в реакторі та системі його охолодження описуються системою рівнянь:

$$\frac{d}{dx}\left(-D\cdot\frac{dc_A}{dx}\right)=R_1-u_{розчин}\cdot\frac{dc_A}{dx} \tag{4.7}$$

$$\frac{d}{dx}\left(-D\cdot\frac{dc_B}{dx}\right)=R_2-u_{розчин}\cdot\frac{dc_B}{dx} \tag{4.8}$$

$$\frac{d}{dx}\left(-\lambda_{розчин}\cdot\frac{dT_{розчин}}{dx}\right)=Q_{розчин}-\rho_{розчин}\cdot c_{mp\ розчин}\cdot u_{розчин}\cdot\frac{dT_{розчин}}{dx} \tag{4.9}$$

$$\begin{aligned}&\frac{d}{dx}\left(-\lambda_{охолоджувач}\cdot\frac{dT_{охолоджувач}}{dx}\right)=\\&=Q_{охолоджувач}-\rho_{охолоджувач}\cdot c_{mp\ охолоджувач}\cdot u_{охолоджувач}\cdot\frac{dT_{охолоджувач}}{dx}\end{aligned} \tag{4.10}$$

де x- просторова координата, м;

D- коефіцієнт дифузії, $м^2/с$;

c_A - масова концентрація продукту “A” в реакторі, $кг/м^3$;

R_1- швидкість реакції перетворення продукту “A” в продукт “B”, $моль/(м^3\cdot с)$;

$u_{розчин}$ - швидкість руху розчину продуктів в реакторі, $м/с$;

c_B - масова концентрація продукту “B” в реакторі, $кг/м^3$;

R_2- швидкість реакції перетворення продукту “B” в продукт “C”, $моль/(м^3\cdot с)$;

$\lambda_{розчин}$ - коефіцієнт теплопровідності розчину, $Вт/(м\cdot К)$;

$T_{розчин}$ - температура розчину, К;

$Q_{розчин}$ - теплота, яка надходить в розчин, $Вт/м^3$;

$\rho_{розчин}$ - густина розчину, $кг/м^3$;

$c_{mp\ розчин}$ - середня масова ізобарна теплоємність розчину, $Дж/(кг\cdot К)$;

$\lambda_{охолоджувач}$ - коефіцієнт теплопровідності охолоджувальної рідини, $Вт/(м\cdot К)$;

$T_{охолоджувач}$ - температура охолоджувальної рідини, К;

$Q_{охолоджувач}$ - кількість теплоти, яка передається від розчину охолоджувальній рідині, $Вт/м^3$;

$\rho_{охолоджувач}$ - густина охолоджувальної рідини, $кг/м^3$;

$c_{mp\ охолоджувач}$ - середня масова ізобарна теплоємність охолоджувальної рідини, $Дж/(кг \cdot К)$;

$u_{охолоджувач}$ - швидкість руху охолоджувальної рідини в охолоджувальній оболонці, $м/с$.

Швидкість реакції перетворення продукту "A" в продукт "B" в формулі (4.7) визначається за формулою:

$$R_1 = -c_A \cdot A_1 \cdot e^{\frac{-E_1}{R \cdot T_{розчин}}} \tag{4.11}$$

Швидкість реакції перетворення продукту "B" в продукт "C" в формулі (4.8) визначається за формулою:

$$R_2 = c_A \cdot A_1 \cdot e^{\frac{-E_1}{R \cdot T_{розчин}}} - c_B \cdot A_2 \cdot e^{\frac{-E_2}{R \cdot T_{розчин}}} \tag{4.12}$$

Кількість теплоти, яка надходить в розчин, в формулі (4.9) визначається за формулою:

$$\begin{aligned} Q_{розчин} = -H_1 \cdot \left(c_A \cdot A_1 \cdot e^{\frac{-E_1}{R \cdot T_{розчин}}} \right) - H_2 \cdot \left(c_B \cdot A_2 \cdot e^{\frac{-E_2}{R \cdot T_{розчин}}} \right) - \\ - UA \cdot \left(T_{розчин} - T_{охолоджува\ ч} \right) \end{aligned} \tag{4.13}$$

Кількість теплоти, яка передається від розчину охолоджувальній рідині, в формулі (4.10) визначається за формулою:

$$Q_{\text{охолоджувач}} = UA \cdot \left(T_{\text{розчин}} - T_{\text{охолоджувач}}\right) \quad (4.14)$$

Граничні умови задачі для рівняння (4.7) та рівняння (4.8):

- на вході розчину продуктів “A” та “B” до трубчастого хімічного реактора задаються їх концентрації, тобто задається гранична умова Діріхле: $c_A = c_{A0}$ та $c_B = c_{B0}$;

- на виході розчину продуктів “A” та “B” з трубчастого хімічного реактора задається гранична умова Неймана: $\frac{dc_A}{dx} = 0$ та $\frac{dc_B}{dx} = 0$ оскільки точне значення концентрацій продуктів “A” та “B” на виході з трубчастого хімічного реактора є невідомими.

Граничні умови задачі для рівняння (4.9) та рівняння (4.10):

- на вході розчину продуктів “A” та “B” до трубчастого хімічного реактора задається температура розчину, тобто задається гранична умова Діріхле: $T_{\text{розчин}} = T_{\text{розчин}\,0}$;

- на вході охолоджувальної рідини до охолоджувальної оболонки задається температура охолоджувальної рідини, тобто задається гранична умова Діріхле: $T_{\text{охолоджувач}} = T_{\text{охолоджувач}\,0}$;

- на виході розчину продуктів “A” та “B” з трубчастого хімічного реактора та охолоджувальної рідини з охолоджувальної оболонки задається гранична умова Неймана: $\frac{dT_{\text{розчин}}}{dx} = 0$ та $\frac{dT_{\text{охолоджувач}}}{dx} = 0$ оскільки точне значення температури розчину на виході з трубчастого хімічного реактора та температури охолоджувальної рідини на виході з охолоджувальної оболонки є невідомими.

Формулювання задачі, таким чином, завершено і можна розпочати її розв’язання в середовищі COMSOL MULTIPHYSICS®. Для цього потрібно виконати наступні кроки.

1. Відкрити **Model Navigator**.

2. Вибрати в списку **Space Dimension 1D**.

3. Натиснути на довгасту кнопку **Multiphysics**, яка знаходиться у правому нижньому кутку меню **Model Navigator**.

4. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **Convection and Diffusion** > **Convection and Diffusion** > **Steady-state analysis**.

5. В області **Dependent variables** замість розрахункової функції **c** (яка є такою поза вибором) потрібно написати **c_A** (концентрація продукту “A”).

6. Натиснути на кнопку **Add** в полі **Multiphysics**.

7. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **Convection and Diffusion** > **Convection and Diffusion** > **Steady-state analysis**.

8. В області **Dependent variables** замість розрахункової функції **c** (яка є такою поза вибором) потрібно написати **c_B** (концентрація продукту “B”).

9. Натиснути на кнопку **Add** в полі **Multiphysics**.

Увага ! За допомогою наведених в пунктах 4…9 кроків до програмного продукту додаються рівняння (4.7) та рівняння (4.8).

---

10. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **Heat Transfer** > **Convection and Conduction** > **Steady-state analysis**.

11. В області **Dependent variables** замість розрахункової функції **T** (яка є такою поза вибором) потрібно написати **T_rastvor** (температура розчину).

12. Натиснути на кнопку **Add** в полі **Multiphysics**.

13. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **Heat Transfer** > **Convection and Conduction** > **Steady-state analysis**.

14. В області **Dependent variables** замість розрахункової функції **T** (яка є такою поза вибором) потрібно написати **T_okhladitel** (температура охолоджувальної рідини).

15. Натиснути на кнопку **Add** в полі **Multiphysics**.

Увага ! За допомогою наведених в пунктах 10…15 кроків до програмного продукту додаються рівняння (4.9) та рівняння (4.10).

---

16. Натиснути **ОК**.

Виконання перелічених вище дій призводить до формування системи, що складається з чотирьох диференціальних рівнянь відповідно до кількості невідомих величин, що містяться в цих рівняннях. В даній задачі невідомими величинами є концентрація продукту “А” c_A, концентрація продукту “В” c_B, температура розчину $T_{розчин}$ та температура охолоджувальної рідини $T_{охолоджува ч}$.

Відкриється графічний інтерфейс користувача (GUI).

Тепер потрібно зберегти файл, попередньо присвоївши йому ім’я. Для цього потрібно зайти в пункт меню **File** > **Save As**. В полі **File name** потрібно ввести ім’я файлу, наприклад, **reactor_2**.

Для відображення геометричної області, в якій буде відшукуватись розв’язок задачі, потрібно:

17. Зайти в пункт меню **Draw** > **Specify Objects** > **Line**.

У вікні в полі **Coordinates** потрібно ввести значення радіуса тіла сферичної форми:

x: 0 2.0 (через пробіл).

18. Натиснути **ОК**.

Задамо константи, які входять до даної задачі.

Для цього потрібно виконати наступні кроки:

19. Зайти в пункт меню **Options** > **Constants**.

Значення коефіцієнтів в закладку **Constants** потрібно ввести так, як показано у вікні таблиці 4.13 за умовами даної програми.

20. Зайти в пункт меню **Multiphysics**. Переконатися, що перемикач встановлено навпроти першого диференціального рівняння (диференціального рівняння для розрахункової функції **c_A**), тобто саме

воно є в даний момент активним.

Таблиця 4.13- Вікно присвоєння значень коефіцієнтів

Name	Expression	Value	Description
A_1	$1.6e8[1/s]$	$1.6e8[1/s]$	
A_2	$1e15[1/s]$	$10e14[1/s]$	
E_1	75000[J/mol]	75000[J/mol]	
E_2	125000[J/mol]	125000[J/mol]	
R	8.314[J/mol/K]	8.314[J/mol/K]	
u_rastvor	0.0042[m/s]	0.0042[m/s]	
c_A_init	700[mol/m^3]	700[mol/m^3]	
H_1	200000[J/mol]	2e5[J/mol]	
H_2	100000[J/mol]	1e5[J/mol]	
D	1e-8[m^2/s]	(1e-8)[m^2/s]	
UA	10000[W/m^3/K]	$10000\left[W/\left(m^3 \cdot K\right)\right]$	
T_rastvor_init	400[K]	400[K]	
u_okhladitel	0.001[m/s]	0.001[m/s]	

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
● 1 Convection and Diffusion (cd) 2 Convection and Diffusion (cd2) 3 Convection and Conduction (cc) 4 Convection and Conduction (cc2)

Для задання диференціального рівняння (4.7) потрібно виконати наступні кроки.

21. Зайти в пункт меню **Physics** > **Subdomain Settings**.

22. На закладці **Subdomains** в полі **Subdomain selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

23. Поле **Species** на закладці **c_A** поза вибором виглядає наступним чином (табл. 4.14).

Таблиця 4.14- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
D	1	m^2/s	Diffusion coefficient
R	1	$mol/\left(m^3 \cdot s\right)$	Reaction rate
u	0	m/s	x- velocity

Як видно з таблиці 4.14 до неї потрібно ввести значення коефіцієнта дифузії, швидкість реакції, а також швидкість руху продукту “A” вздовж

координатної осі **x** реактора.

Існує три способи задання цих значень:

1) Безпосередньо в полі **Species**.

2) На закладці **Constants** в пункті меню **Options**, якщо розв'язується стаціонарна задача і значення коефіцієнта дифузії, швидкість реакції, а також швидкість руху продукту "A" вздовж координатної осі **x** реактора не залежать від температури рідини, тобто є константами.

3) В пункті меню **Options** > **Expressions**, якщо розв'язується нестаціонарна спряжена задача дифузії, конвекції та теплопровідності. В такому випадку температура розчину продуктів буде змінюватися в часі, у зв'язку з чим будуть змінюватися в часі коефіцієнта дифузії, швидкість реакції, а також швидкість руху продукту "A" вздовж координатної осі **x** реактора, оскільки вони є функціями температури, тобто залежать від неї.

Увага ! Задача формально є стаціонарною, оскільки в рівняннях (4.7)-(4.10) відсутні похідні розрахункових величин (c_A, c_B, T_rastvor, T_okhladitel) за часом. Проте, оскільки розв'язується задача оптимізації, в якій необхідно знайти таке значення температури охолоджувальної рідини на вході до охолоджувальної оболонки, при якому на виході з реактора продукт "B" матиме максимальну концентрацію, очевидно, що температура охолоджувальної рідини, а, разом з тим, і температура розчину змінюватимуть своє значення вздовж координатної осі **x** реактора. Таким чином, задачу можна вважати нестаціонарною і для задання коефіцієнта дифузії, швидкість реакції, а також швидкість руху продукту "A" вздовж координатної осі **x** реактора придатними є лише перший та третій способи.

Скористаємось першим способом задання вказаних вище величин.

Для цього поле **Species** на закладці **c_A** потрібно оформити так, як показано в таблиці 4.15.

Таблиця 4.15- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
D	D	m^2/s	Diffusion coefficient
R	-c_A*A_1* *exp(-E_1/(R*T_rastvor))	$mol/(m^3 \cdot s)$	Reaction rate
u	u_rastvor	m/s	x- velocity

24. Натиснути **ОК**.

Увага ! Налаштування закладки **Init** та закладки **Element** потрібно залишити без змін, тобто такими, якими вони є поза вибором.

Для задання граничних умов рівняння (4.7) потрібно виконати наступні кроки:

25. Зайти в пункт меню **Physics** > **Boundary Settings**.

26. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

На закладці **c_A** в полі **Boundary condition** в меню, що розгортається, встановити граничну умову **Concentration**. Потрібно ввести значення коефіцієнтів в закладку **c_A** так, як показано в таблиці 4.16.

Таблиця 4.16- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
c_A_0	c_A_init	mol/m^3	Concentration
N_0	0 (поле не активне)	$mol/(m^2 \cdot s)$	Inward flux
D	0 (поле не активне)	m^2/s	Diffusion coefficient
d	1 (поле не активне)	m	Thickness

27. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. Після цього поле з цифрою **2** зафарбується в синій колір.

На закладці **c_A** в полі **Boundary condition** в меню, що розгортається, встановити граничну умову **Convective flux** (тепловий потік). Всі поля цієї закладки є неактивними і виглядають так, як показано в таблиці 4.17.

Таблиця 4.17- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
c_A_0	0 (поле не активне)	mol/m^3	Concentration
N_0	0 (поле не активне)	$mol/(m^2 \cdot s)$	Inward flux
D	0 (поле не активне)	m^2/s	Diffusion coefficient
d	1 (поле не активне)	m	Thickness

28. Натиснути **OK**.

Виконанням цієї дії завершується задання рівняння (4.7), а також граничних умов до нього в COMSOL MULTIPHYSICS®.

Тепер можна перейти до введення в COMSOL MULTIPHYSICS®

рівняння (4.8).

Для цього потрібно зайти в пункт меню **Multiphysics**. Встановити перемикач навпроти другого диференціального рівняння (диференціального рівняння для розрахункової функції **c_B**), тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
1 Convection and Diffusion (cd) ● 2 Convection and Diffusion (cd2) 3 Convection and Conduction (cc) 4 Convection and Conduction (cc2)

Для задання диференціального рівняння (4.8) потрібно виконати наступні кроки.

29. Зайти в пункт меню **Physics** > **Subdomain Settings**.

30. На закладці **Subdomains** в полі **Subdomain selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

31. Поле **Species** на закладці **c_B** поза вибором виглядає наступним чином (табл. 4.18).

Таблиця 4.18- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
D	1	m^2/s	Diffusion coefficient
R	1	$mol/(m^3 \cdot s)$	Reaction rate
u	0	m/s	x- velocity

Як видно з таблиці 4.18 до неї потрібно ввести значення коефіцієнта дифузії, швидкість реакції, а також швидкість руху продукту “B” вздовж координатної осі **x** реактора.

Для цього поле **Species** на закладці **c_B** потрібно оформити так, як показано в таблиці 4.19.

32. Натиснути **OK**.

Увага ! Налаштування закладки **Init** та закладки **Element** потрібно залишити без змін, тобто такими, якими вони є поза вибором.

Для задання граничних умов рівняння (4.8) потрібно виконати наступні кроки:

33. Зайти в пункт меню **Physics** > **Boundary Settings**.

Таблиця 4.19- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
D	D	m^2/s	Diffusion coefficient
R	c_A*A_1* *exp(-E_1/(R*T_rastvor))- -c_B*A_2* *exp(-E_2/(R*T_rasvor))	$mol/(m^3 \cdot s)$	Reaction rate
u	u_rastvor	m/s	x- velocity

34. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

На закладці **c_B** в полі **Boundary condition** в меню, що розгортається, встановити граничну умову **Concentration**. Потрібно ввести значення коефіцієнтів в закладку **c_B** так, як показано в таблиці 4.20.

Таблиця 4.20- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
c_B_0	0	mol/m^3	Concentration
N_0	0 (поле не активне)	$mol/(m^2 \cdot s)$	Inward flux
D	0 (поле не активне)	m^2/s	Diffusion coefficient
d	1 (поле не активне)	m	Thickness

35. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. Після цього поле з цифрою **2** зафарбується в синій колір.

На закладці **c_B** в полі **Boundary condition** в меню, що розгортається, встановити граничну умову **Convective flux**. Всі поля цієї закладки є неактивними і виглядають так, як показано в таблиці 4.21.

Таблиця 4.21- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
c_B_0	0 (поле не активне)	mol/m^3	Concentration
N_0	0 (поле не активне)	$mol/(m^2 \cdot s)$	Inward flux
D	0 (поле не активне)	m^2/s	Diffusion coefficient
d	1 (поле не активне)	m	Thickness

36. Натиснути **ОК**.

Виконанням цієї дії завершується задання рівняння (4.8), а також граничних умов до нього в COMSOL MULTIPHYSICS®.

Тепер можна перейти до введення в COMSOL MULTIPHYSICS® рівняння (4.9).

Для цього потрібно зайти в пункт меню **Multiphysics**. Встановити перемикач навпроти третього диференціального рівняння (диференціального рівняння для розрахункової функції **T_rastvor**), тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
1 Convection and Diffusion (cd) 2 Convection and Diffusion (cd2) ● 3 Convection and Conduction (cc) 4 Convection and Conduction (cc2)

Для задання диференціального рівняння (4.9) потрібно виконати наступні кроки.

37. Зайти в пункт меню **Physics** > **Subdomain Settings**.

38. На закладці **Subdomains** в полі **Subdomain selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

39. Поле **Thermal properties and heat sources/sinks** (термічні властивості та джерела теплоти/відведення теплоти) на закладці **Physics** поза вибором виглядає наступним чином (табл. 4.22).

Таблиця 4.22- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
k	0.025	$W/(m \cdot K)$	Thermal conductivity
ρ	1.205	kg/m^3	Density
C_p	1006	$J/(kg \cdot K)$	Heat capacity at constant pressure
γ	1		Ratio of specific heats
Q	0	W/m^3	Heat source
u	0	m/s	Velocity field

Як видно з таблиці 4.22 до неї потрібно ввести значення коефіцієнта теплопровідності розчину продуктів “A” та “B”, густини розчину продуктів “A” та “B” в реакторі, середню ізобарну масову теплоємність розчину продуктів “A” та “B”, показник адіабати, кількість теплоти, яка

надходить в розчин та швидкість руху розчину продуктів “А” та “В” в реакторі.

Для цього потрібно виконати наступні кроки:

40. Оскільки продукти “А” та “В” розчинені у воді, потрібно натиснути на кнопку **Load…**, яка знаходиться в полі **Thermal properties and heat sources/sinks** на закладці **Physics** та обрати воду зі списку пропонованих матеріалів.

Увага ! Кнопка **Load…** використовується для доступу до бібліотеки матеріалів (**Library materials**). В бібліотеці матеріалів знаходиться певна кількість твердих матеріалів, рідин, газів тощо, з введеними до програмного продукту COMSOL MULTIPHYSICS® властивостями, які не потрібно задавати власноруч. Властивості матеріалів, які містяться в бібліотеці програмного продукту, не є константами. Вони залежать від температури, агрегатного стану матеріалу тощо.

Якщо матеріал відсутній в бібліотеці програмного продукту, його властивості потрібно вводити власноруч.

Разом з тим, власноруч можна вводити властивості навіть тих матеріалів, які містяться в бібліотеці, що потрібно в окремих випадках.

Після натискання на кнопку **Load…** відкривається вікно з трьома полями. В лівій частині вікна розташовано поле **Materials** (Матеріали) та поле **Search** (Пошук), а в лівій частині вікна розташовано поле **Material properties** (Властивості матеріалу).

Оскільки жодного матеріалу з поля **Materials** не вибрано, поле **Material properties** є неактивним. Поле **Search** є незаповненим тобто порожнім, оскільки пошук матеріалу в бібліотеці не здійснювався.

Увага ! В полі **Materials** містяться матеріали, з введеними до програмного продукту COMSOL MULTIPHYSICS® властивостями. Зокрема, бібліотека містить такі розділи:

- **Material library** (Бібліотека матеріалів), в якій містяться всі матеріали, з введеними до програмного продукту COMSOL MULTIPHYSICS® властивостями. Кількість таких матеріалів дорівнює 2542.

- **Basic material properties** (Основні властивості матеріалів).

- **Liquids and gases** (Рідини та гази).

- **MEMS** material properties (Властивості матеріалів мікроелектромеханічних систем).

- **Heat Transfer Coefficients** (Коефіцієнти тепловіддачі).

- **Electric (AC/DC) Material Properties** (Електричні (змінний/постійний струм) властивості матеріалів).

- **Piezoelectric Material Properties** (Властивості п'єзоелектричних матеріалів).

- **User Defined Materials** (Матеріали, визначені користувачем).

Оскільки вода, в якій розчинено продукти "A" та "B", є рідиною, очевидно, що властивості цього матеріалу, тобто води, розташовуються в розділі бібліотеки, що має назву **Liquids and gases**.

Для вибору води з розділу бібліотеки **Liquids and gases** потрібно здійснити подвійне натискання на назві розділу бібліотеки, тобто на словосполученні **Liquids and gases**.

Увага ! Обрати потрібний матеріал з певного розділу бібліотеки можна також двома іншими способами.

Для реалізації першого способу потрібно натиснути на знак "+", який розташовано ліворуч від назви розділу бібліотеки.

Для реалізації другого способу в порожній стрічці пошуку **Search string** поля **Search** потрібно ввести слово "Water" (без лапок) та натиснути на кнопку **Search**.

Після подвійного натискання на словосполученні **Liquids and gases** відкриваються чотири підрозділи бібліотеки, які мають назви:

- **Gases** (Гази);

- **Liquids** (Рідини);

- **Liquid/Gas surface tension** (Рідина/газ поверхневий натяг);

- **Liquid/liquid interfacial tension** (Рідина/рідина поверхневий натяг на границі розділу).

Очевидно, що вода знаходиться в підрозділі бібліотеки, який має назву **Liquids**. Для відкриття цього підрозділу бібліотеки потрібно здійснити подвійне натискання на назві підрозділу бібліотеки, тобто на слові **Liquids**.

Після подвійного натискання на слові **Liquids** відкривається

підрозділ бібліотеки, в якому останньою в списку знаходиться вода (**Water**).

Після натискання на слові **Water** в лівій частині вікна полі **Material properties** з'являється ім'я рідини:

Name: Water

В закладках:

- **Physics** (Фізика);
- **Elastic** (еластичні);
- **Electric** (електричні);
- **Fluid** (Рідина);
- **Piezoelectric** (П'єзоелектричні);
- **Thermal** (Теплові);
- **All** (Всі);

містяться фізичні, еластичні, електричні, рідинні, п'єзоелектричні та теплові властивості матеріалу, тобто води. В закладці **All** зібрано всі властивості матеріалу.

41. Натиснути **OK**.

Поле **Thermal properties and heat sources/sinks** на закладці **Physics** виглядатиме наступним чином (табл. 4.23).

Таблиця 4.23- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
k	**k(T_rastvor[1/K])[W/(m*K)]**	$W/(m \cdot K)$	Thermal conductivity
ρ	**rho(T_rastvor[1/K])[kg/m^3]**	kg/m^3	Density
C_p	**Cp(T_rastvor[1/K])[J/(kg*K)]**	$J/(kg \cdot K)$	Heat capacity at constant pressure
γ	**1.0**		Ratio of specific heats
Q	0	W/m^3	Heat source
u	0	m/s	Velocity field

Як видно з таблиці 4.23 коефіцієнт теплопровідності розчину продуктів "A" та "B", його густина, а також середня ізобарна масова теплоємність розчину продуктів "A" та "B" залежать від температури

розчину **T_rastvor** і не є сталими. Це дозволяє уникнути значної похибки під час розрахунку наведених вище параметрів.

Показник адіабати потрібно залишити таким, яким він є поза вибором.

В поле для виразу кількості теплоти, яка надходить в розчин потрібно ввести праву частину рівняння (4.13).

В поле для виразу швидкості руху розчину продуктів “A” та “B” в реакторі потрібно ввести її вираз згідно з умовами задачі (табл. 4.13).

Таким чином, поле **Thermal properties and heat sources/sinks** на закладці **Physics** виглядатиме наступним чином (табл. 4.24).

Таблиця 4.24- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
k	k(T_rastvor[1/K])[W/(m*K)]	$W/(m \cdot K)$	Thermal conductivity
ρ	rho(T_rastvor[1/K])[kg/m^3]	kg/m^3	Density
C_p	Cp(T_rastvor[1/K])[J/(kg*K)]	$J/(kg \cdot K)$	Heat capacity at constant pressure
γ	1.0		Ratio of specific heats
Q	-H_1*(c_A*A_1* *exp(-E_1/(R*T_rasvor)))- -H_2*(c_B*A_2* *exp(-E_2/(R*T_rastvor)))- -UA*(T_rastvor-T_okhladitel)	W/m^3	Heat source
u	u_rastvor	m/s	Velocity field

Увага ! Дані, які розміщено в закладках **Stabilization** (Стабілізація) , **Init** (Початкові умови) та **Element** (Елемент) в переважній більшості випадків залишаються без змін.

---

42. Натиснути **ОК**.

Для задання граничних умов рівняння (4.9) потрібно виконати наступні кроки:

43. Зайти в пункт меню **Physics** > **Boundary Settings**.

44. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

На закладці **Coefficients** в полі **Boundary conditions** в меню, що

розгортається, встановити граничну умову **Temperature**. Потрібно ввести значення температури розчину продуктів в поле $\mathbf{T_0}$ так, як показано в таблиці 4.25.

Таблиця 4.25- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
q_0	0 (поле не активне)	W/m^2	Inward heat flux
T_0	T_rastvor_init	K	Temperature

45. Натиснути **ОК**.

Увага ! Нагадаємо, що **T_rastvor_init** становить собою температуру розчину на вході до трубчастого хімічного реактора. Значення цієї величини наведено в таблиці 4.12 і вже задано в меню **Options** > **Constants**.

Значення температури розчину на вході до трубчастого хімічного реактора **T_rastvor_init** є довільним, впливає лише на швидкість відшукання абсолютного мінімуму або максимуму цільової функції і не впливає на точність відшукання мінімуму або максимуму цільової функції.

В даній задачі в якості температури розчину на вході до трубчастого хімічного реактора **T_rastvor_init** задається $T_{розчин0} = 400K$.

---

46. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. Після цього поле з цифрою **2** зафарбується в синій колір.

На закладці **Coefficients** в полі **Boundary conditions** в меню, що розгортається, встановити граничну умову **Convective flux**. Всі поля цієї закладки є неактивними і виглядають так, як показано в таблиці 4.26.

Таблиця 4.26- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
q_0	0 (поле не активне)	W/m^2	Inward heat flux
T_0	273.15(поле не активне)	K	Temperature

47. Натиснути **ОК**.

Виконанням цієї дії завершується задання рівняння (4.9), а також граничних умов до нього в COMSOL MULTIPHYSICS®.

Тепер можна перейти до введення в COMSOL MULTIPHYSICS® рівняння (4.10).

Для цього потрібно зайти в пункт меню **Multiphysics**. Встановити перемикач навпроти четвертого диференціального рівняння (диференціального рівняння для розрахункової функції **T_okhladitel**), тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
1 Convection and Diffusion (cd)
2 Convection and Diffusion (cd2)
3 Convection and Conduction (cc)
● 4 Convection and Conduction (cc2)

Для задання диференціального рівняння (4.10) потрібно виконати наступні кроки.

48. Зайти в пункт меню **Physics** > **Subdomain Settings**.

49. На закладці **Subdomains** в полі **Subdomain selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

50. Оскільки матеріал (воду) вже обрано в полі **Library Material** при заданні рівняння (4.9), поле **Thermal properties and heat sources/sinks** на закладці **Physics** виглядає наступним чином (табл. 4.27).

Таблиця 4.27- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
k	k(T_rastvor[1/K])[W/(m*K)]	$W/(m \cdot K)$	Thermal conductivity
ρ	rho(T_rastvor[1/K])[kg/m^3]	kg/m^3	Density
C_p	Cp(T_rastvor[1/K])[J/(kg*K)]	$J/(kg \cdot K)$	Heat capacity at constant pressure
γ	1.0		Ratio of specific heats
Q	0	W/m^3	Heat source
u	0	m/s	Velocity field

В поле для виразу кількість теплоти, яка передається від розчину охолоджувальній рідині, потрібно ввести праву частину рівняння (4.14).

В поле для виразу швидкості руху охолоджувальної рідини в охолоджувальній оболонці, потрібно ввести її вираз згідно з умовами задачі (табл. 4.13).

Таким чином, поле **Thermal properties and heat sources/sinks** на закладці **Physics** виглядатиме наступним чином (табл. 4.28).

Увага ! Дані, які розміщено в закладках **Stabilization** (Стабілізація) , **Init** (Початкові умови) та **Element** (Елемент) в переважній більшості випадків залишаються без змін.

Таблиця 4.28- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
k	k(T_rastvor[1/K])[W/(m*K)]	$W/(m \cdot K)$	Thermal conductivity
ρ	rho(T_rastvor[1/K])[kg/m^3]	kg/m^3	Density
C_p	Cp(T_rastvor[1/K])[J/(kg*K)]	$J/(kg \cdot K)$	Heat capacity at constant pressure
γ	1.0		Ratio of specific heats
Q	UA*(T_rastvor-T_okhladitel)0	W/m^3	Heat source
u	u_okhladitel	m/s	Velocity field

51. Натиснути **ОК**.

Для задання граничних умов рівняння (4.10) потрібно виконати наступні кроки:

52. Зайти в пункт меню **Physics** > **Boundary Settings**.

53. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **1**. Після цього поле з цифрою **1** зафарбується в синій колір.

На закладці **Coefficients** в полі **Boundary conditions** в меню, що розгортається, встановити граничну умову **Temperature**. Потрібно ввести значення температури розчину продуктів в поле $\mathbf{T_0}$ так, як показано в таблиці 4.29.

Таблиця 4.29- Вікно присвоєння значень коефіцієнтів

Quantity	Value / Expression	Unit	Description
q_0	0 (поле не активне)	W/m^2	Inward heat flux
T_0	T_okhladitel_init	K	Temperature

54. Натиснути **ОК**.

Увага ! Нагадаємо, що **T_okhladitel_init** становить собою температуру охолоджувальної рідини на вході до охолоджувальної оболонки. Значення цієї величини наведено в таблиці 4.12 і вже задано в меню **Options** > **Constants**.

Значення температури охолоджувальної рідини на вході до трубчастого хімічного реактора **T_okhladitel_init** є довільним, впливає лише на швидкість відшукання абсолютного мінімуму або максимуму цільової функції і не впливає на точність відшукання мінімуму або максимуму цільової функції.

В даній задачі в якості температури охолоджувальної рідини на вході до охолоджувальної оболонки **T_okhladitel_init** задається $T_{охолоджувач0} = 350K$.

---

55. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. Після цього поле з цифрою **2** зафарбується в синій колір.

На закладці **Coefficients** в полі **Boundary conditions** в меню, що розгортається, встановити граничну умову **Convective flux**. Всі поля цієї закладки є неактивними і виглядають так, як показано в таблиці 4.30.

Таблиця 4.30- Вікно присвоєння значень коефіцієнтів

Quantity	**Value / Expression**	**Unit**	**Description**
q_0	0 (поле не активне)	W/m^2	Inward heat flux
T_0	273.15(поле не активне)	K	Temperature

56. Натиснути **ОК**.

Виконанням цієї дії завершується задання рівняння (4.10), а також граничних умов до нього в COMSOL MULTIPHYSICS®.

Наступним кроком є задання цільової функції та умов, тобто обмежень, які накладено на параметри, що описують дану задачу.

Для цього потрібно виконати наступні кроки.

57. Зайти в меню **Multiphysics** та відкрити пункт меню **Model Navigator**.

58. В **Application Modes** зайти в пункт меню **Comsol Multiphysics** > **Optimization and Sensitivity** > **Optimization**.

Увага ! В області **Dependent variables** поле з ім’ям розрахункової функції поза вибором є неактивним. Для модуля **Optimization** спосіб задання самої розрахункової функції, тобто цільової функції та її імені дещо відрізняються від способу задання розрахункової функції диференціальних рівнянь та їх імен. Спосіб задання самої цільової функції та її імені для модуля **Optimization** будуть розглянуті в рамках цієї задачі згодом.

---

59. Натиснути на кнопку **Add** в полі **Multiphysics**.

60. Натиснути **ОК.**

Виконання перелічених вище дій призводить до формування системи, що складається з чьотирьох диференціальних рівнянь, відповідно до кількості невідомих величин, та цільової функції.

Для задання цільової функції та інших параметрів задачі оптимізації необхідно виконати наступні кроки.

61. Зайти в пункт меню **Multiphysics**. Переконатися, що перемикач

встановлено навпроти третього диференціального рівняння, тобто саме воно є в даний момент активним.

Пункт меню **Multiphysics** повинен виглядати наступним чином:

Model Navigator…
1 Convection and Diffusion (cd) 2 Convection and Diffusion (cd2) 3 Convection and Conduction (cc) 4 Convection and Conduction (cc2) • 5 Optimization (opt)

Задання цільової функції та інших параметрів оптимізації здійснюється на основі таких міркувань.

Оскільки задача оптимізації полягає у тому, щоб при заданій довжині **L** реактора знайти таке значення температури охолоджувальної рідини **T_okhladitel** на вході до охолоджувальної оболонки, при якому на виході з реактора продукт "B" матиме максимальну концентрацію, тобто c_B набуватиме максимального значення, необхідно певним чином задати цільову функцію саме на границі **2** розрахункової області.

Для задання цільової функції на границі **2** розрахункової області необхідно виконати наступні кроки.

62. Зайти в пункт меню **Physics** > **Boundary Settings**.

63. На закладці **Boundaries** в полі **Boundary selection** натиснути на цифру **2**. Після цього поле з цифрою **2** зафарбується в синій колір.

На закладці **Objective** (цільова функція) в полі **Objective contribution** (вираз для цільової функції) поза вибором значення параметру $\mathbf{q_0}$ у вічці **Point contribution** (внесок для точки) є відсутнім.

У вічко **Point contribution** потрібно записати вираз або розрахункову величину, мінімальне значення якого (якої) потрібно відшукати.

Оскільки в рамках розв'язання даної задачі необхідно забезпечити максимальне значення концентрації продукту "B", тобто необхідно забезпечити максимальне значення розрахункової функції **c_B** на границі **2**, вічко **Point contribution** необхідно оформити так, як показано в таблиці 4.31.

Таблиця 4.31- Вікно присвоєння значень коефіцієнта

q_0	$-c\_B$	Point contribution

Увага ! Варто знову нагадати, що задавати цільову функцію завжди потрібно в тій частині розрахункової області, в якій необхідно відшукати максимум або мінімум цільової функції.

Перед позначенням розрахункової функції **c_B** стоїть знак "мінус", оскільки за умовами даної задачі необхідно відшукати максимальне значення цільової функції, в той час, як програмний продукт завжди відшукуватиме мінімальне значення цільової функції. Таким чином, для відшукування максимального значення цільової функції перед її виразом необхідно поставити знак "мінус".

В разі, якщо необхідно відшукати мінімальне значення цільової знак "мінус" ставити не потрібно.

---

64. Натиснути **ОК.**

Увага ! Налаштування закладки **Integral Constraints** (Інтегральні обмеження), закладки **Pointwise Constraints** (Точкові обмеження) та закладки **Variables** (Змінні) потрібно залишити без змін, тобто такими, якими вони є поза вибором.

Далі необхідно задати:

- параметр або декілька параметрів, варіювання якими дозволить досягти мінімального або максимального значення цільової функції;

- граничні значення варіювання значень параметра або декількох параметрів;

- межі варіювання (вся розрахункова область, границя області, точка області) параметра або декількох параметрів.

Відповідно до умов даної задачі, параметром, варіювання яким дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, є температура охолоджувальної рідини **T_okhladitel_init** на вході до охолоджувальної оболонки.

Для задання цього параметру, а також його граничних значень та меж варіювання необхідно виконати наступні кроки.

65. Зайти в пункт меню **Physics** > **Scalar Settings**.

На закладці **Variables** (змінні) в полі **Variable settings** (налаштування змінної) поза вибором значення параметрів **Variable**

(змінна) та **Init** є відсутніми.

У вічко **Variable** потрібно записати позначення параметра, варіювання якими дозволить досягти мінімального (в даній задачі максимального) значення цільової функції. Відповідно до умов даної задачі, параметром, варіювання яким дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, є температура охолоджувальної рідини **T_okhladitel_init** на вході до охолоджувальної оболонки.

У вічко **Init** потрібно записати початкове значення температури охолоджувальної рідини **T_okhladitel_init** на вході до охолоджувальної оболонки.

Увага! Початкове та, водночас, довільне значення температури охолоджувальної рідини **T_okhladitel_init** на вході до трубчастого хімічного реактора впливає лише на швидкість відшукання абсолютного мінімуму або максимуму цільової функції і не впливає на точність відшукання мінімуму або максимуму цільової функції.

В даній задачі в якості початкового та довільного значення температури охолоджувальної рідини **T_okhladitel_init** на вході до охолоджувальної оболонки задається 350К.

Таким чином, поле **Variable settings** закладки **Variables** має виглядати так, як показано в таблиці 4.32.

Таблиця 4.32- Вікно присвоєння значень коефіцієнтів

Variable	**Init**
T_okhladitel_init	350

Місце задання меж варіювання температури охолоджувальної рідини **T_okhladitel_init** на вході до охолоджувальної оболонки обумовлено тим, що цей параметр не є розрахунковою функцією, до яких в даній задачі належать концентрації продуктів “A” та “B”, температура розчину та температура охолоджувальної рідини.

Увага! Варто знову нагадати, що в разі, якщо параметр або декілька параметрів, варіювання якими дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, не є розрахунковими функціями, задання параметра або декількох параметрів, а також його (їх) граничних значень та меж

варіювання здійснюється пункті меню **Physics** > **Scalar Settings**.

66. Натиснути **OK**.

Задання всіх умов, що описують задачу, завершено.

67. Для генерації скінченнноелементної сітки зайти в пункт меню **Mesh** > **Free Mesh Parameters** > **Remesh.**

68. Натиснути **OK**.

Для розв'язання задачі необхідно виконати наступні кроки.

69. Зайти в пункт меню **Solve** > **Solver Parameters**. Активізувати опції **Optimization / Sensitivity** (Оптимізація / Чуттєвість) та **Plot while solving** (Зобразити під час розв'язання).

70. На закладці **Optimization / Sensitivity** в меню, що розгортається обрати тип аналізу **Analysis: Optimization**.

71. В поле **Variables** потрібно записати позначення параметра, варіювання якими дозволить досягти мінімального (в даній задачі максимального) значення цільової функції. Відповідно до умов даної задачі, параметром, варіювання яким дозволить досягти мінімального (в даній задачі максимального) значення цільової функції, є температура охолоджувальної рідини на вході до охолоджувальної оболонки **T_okhladitel_init**.

72. Натиснути **OK**.

73. Для розв'язання задачі зайти в пункт меню **Solve** > **Solve Problem**.

Отриманий розв'язок потрібно порівняти з розв'язком, наведеним в літературі [6].

Додаток А

Приклади задач безумовної оптимізації

Приклад А.1.

Необхідно знайти мінімум тестової функції:

$$f(x) = 100 \cdot \left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

Відповідь: $f_{min} = 0$ при $x_{1\,min} = 1{,}0$ та $x_{2\,min} = 1{,}0$.

Приклад А.2.

Необхідно знайти мінімум тестової функції:

$$f(x) = \left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

Відповідь: $f_{min} = 0$ при $x_{1\,min} = 1{,}0$ та $x_{2\,min} = 1{,}0$.

Приклад А.3.

Необхідно знайти мінімум тестової функції:

$$f(x) = \left(x_2 - x_1^2\right)^2 + 100 \cdot (1 - x_1)^2$$

Відповідь: $f_{min} = 0$ при $x_{1\,min} = 1{,}0$ та $x_{2\,min} = 1{,}0$.

Приклад А.4.

Необхідно знайти мінімум тестової функції:

$$f(x) = 100 \cdot \left(x_2 - x_1^3\right)^2 + (1 - x_1)^2$$

Відповідь: $f_{min} = 0$ при $x_{1\,min} = 1{,}0$ та $x_{2\,min} = 1{,}0$.

Приклад А.5.

Необхідно знайти мінімум тестової функції:

$$f(x) = [1{,}5 - x_1 \cdot (1 - x_2)]^2 + \left[2{,}25 - x_1 \cdot \left(1 - x_2^2\right)\right]^2 + \left[2{,}625 - x_1 \cdot \left(1 - x_2^3\right)\right]^2$$

Відповідь: $f_{min} = 0$ при $x_{1\,min} = 3{,}0$ та $x_{2\,min} = 0{,}5$.

Приклад А.6.

Необхідно знайти мінімум тестової функції:

$$f(x)=(x_1+10\cdot x_2)+5\cdot(x_3-x_4)^2+(x_2-2\cdot x_3)^4+10\cdot(x_1-x_4)^4$$

Відповідь: $f_{min}=0$ при $x_{1\,min}=0$, $x_{2\,min}=0$, $x_{3\,min}=0$, $x_{4\,min}=0$.

Приклад А.7.

Необхідно знайти мінімум тестової функції Павелла:

$$y=(x_1+10\cdot x_2)^2+5\cdot(x_3-x_4)^2+(x_2-2\cdot x_3)^4+10\cdot(x_1-x_4)^4$$

Відповідь: $x_{1\,min}=0$; $x_{2\,min}=0$; $x_{3\,min}=0$; $x_{4\,min}=0$.

Приклад А.8.

Функція Віллінга дозволяє оцінювати здібність алгоритму долати розриви функції:

$$f(x)=-3\cdot|x_1|-|x_2|$$

Необхідно знайти максимум функції.

Відповідь: $f_{max}=0$ при $x_{1\,max}=0$ та $x_{2\,max}=0$.

Приклад А.9.

Необхідно знайти значення максимуму та мінімуму функції:

$$y=\frac{x}{x^2+1}$$

Відповідь: $y_{max}=0{,}5$ при $x_{max}=1{,}0$; $y_{min}=-0{,}5$ при $x_{min}=-1{,}0$.

Приклад А.10.

Необхідно знайти мінімум функції:

$$y=|x|.$$

Відповідь: $y_{min}=0$ при $x_{min}=0$.

Приклад А.11.

Необхідно знайти мінімум функції:

$$y = -e^{-x} \cdot \sinh\left(\frac{x}{2}\right)$$

Відповідь: $y_{min} = -\frac{1}{3 \cdot \sqrt{3}} \approx 0{,}19245$.

Приклад А.12.

Необхідно знайти мінімум функції:

$$y = e^{-x} - \cos(x)$$

Відповідь: $y_{min} = -0{,}2766$ при $x_{min} = 0{,}5885$.

Приклад А.13.

Необхідно знайти мінімум функції:

$$y = 0{,}5 \cdot x^2 - \sin(x)$$

Відповідь: $y_{min} = -0{,}4005$ при $x_{min} = 0{,}7391$.

Приклад А.14.

Необхідно знайти мінімум функції:

$$y = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$$

Відповідь: $y_{min} = 0$ при $x_{1\,min} = 3$ та $x_{2\,min} = 2$.

Приклад А.15.

Необхідно знайти мінімум функції:

$$y = x_1^4 + x_2^4 + 2 \cdot x_1^2 \cdot x_2^2 - 4 \cdot x_1 + 3$$

Відповідь: $y_{min} = 0$ при $x_{1\,min} = 1$ та $x_{2\,min} = 0$.

Приклад А.16.

Щорічні витрати, що повязані з експлуатацією газового компрессора на магістральному газопроводі, виражаються формулою:

$$c = \frac{K \cdot Q \cdot Z}{10^6 \cdot L} \cdot \left(\ln \frac{P_1}{P_2} + b \right) + K_1 \cdot D^2 \cdot \left[\frac{P_1}{2 \cdot (S - P_1)} + \frac{P_1^2}{4 \cdot (S + P_1)^2} \right] \tag{А.1}$$

де c - експлуатаційні витрати, $грн/рік$;

Q - кількість газу, що прокачується через компресор, $м^3/доба$;

L - відстань між компресорними станціями, км;

P_1 - тиск газу на виході з компресора, Па;

P_2 - тиск газу на вході в компресор, Па;

D - діаметр трубопроводу, м;

K, K_1, Z, S, b - константи.

Кількість газу, що прокачується через компрессор, визначається по формулі:

$$Q = K_2 \cdot \frac{D^{2,6} \cdot \left(P_1^2 - P_2^2\right)^{0,54}}{L^{0,54} \cdot Z^{0,54}} \tag{А.2}$$

При $Z = 1$; $K = 1370$; $L = 20$; $b = 1,476$; $K_1 = 0,081$; $S = 100$; $K_2 = 1,13$ визначте P_1 и P_2, що мінімізують експлуатаційні витрати.

Приклад А.17.

Сумарні витрати, що пов'язані зі спорудженням ректифікаційної колони, можна представити у вигляді:

$$C = C_p \cdot A \cdot N + C_s \cdot H \cdot A \cdot N + C_f + C_d + C_b + C_L + C_x$$

де c - сумарні затрати, грн;

C_p - вартість одного квадратного метра горизонтального перекриття монолітними плитами, $\frac{грн}{м^2}$;

A - площа поперечного перерізу колони, $м^2$;

N - число монолітних плит для перекриття, шт.;

C_s - питомі затрати, що пов'язані зі спорудженням каркасу колони, $\frac{грн}{м^3}$;

H - відстань між горизонтальними перекриттями (монолітними плитами), $м$;

C_f - вартість подавального насоса, грн;

C_d - вартість системи насосів, що забезпечують ректифікаційний процес, грн;

C_b - вартість насоса відкачки низьких фракцій, грн;

C_L - вартість насоса, що забезпечує повторну перегонку проміжних фракцій, грн;

C_x - інші фіксовані затрати, грн.

Задача полягає у мінімізації сумарних витрат при заданих характеристиках вихідного продукту та пропускної здатності ректифікаційної колони, а також при фіксованих вартісних характеристиках, що відносяться до функціонування усіх видів насосних установок (тобто C_f, C_d, C_b та C_L фіксовані). Після підбору будівельного матеріалу становляться фіксованими C_p, C_s и C_x.

Змінні, що описують технологічний процес, пов'язані двома емпіричними співвідношеннями:

$$\frac{L}{D} = \left(\frac{1}{1 - \left(\frac{N_{min}}{N} \right)} \right) \cdot \left(\frac{L}{D} \right)_{min}$$

$$A = K \cdot (L + D)$$

Певні із типових ректифікаційних колон характеризується наступними значеннями параметрів:

$$C_p = 30;$$

$$C_s = 10;$$

$$H = 2;$$

$$C_f = 4000;$$

$$C_d = 3000;$$

$$C_b = 2000;$$

$$C_x = 8000;$$

$$F = 1500;$$

$$D = 1000;$$

$$N_{min} = 5;$$

$$\frac{L}{D} = 1;$$

$$K = 0{,}01.$$

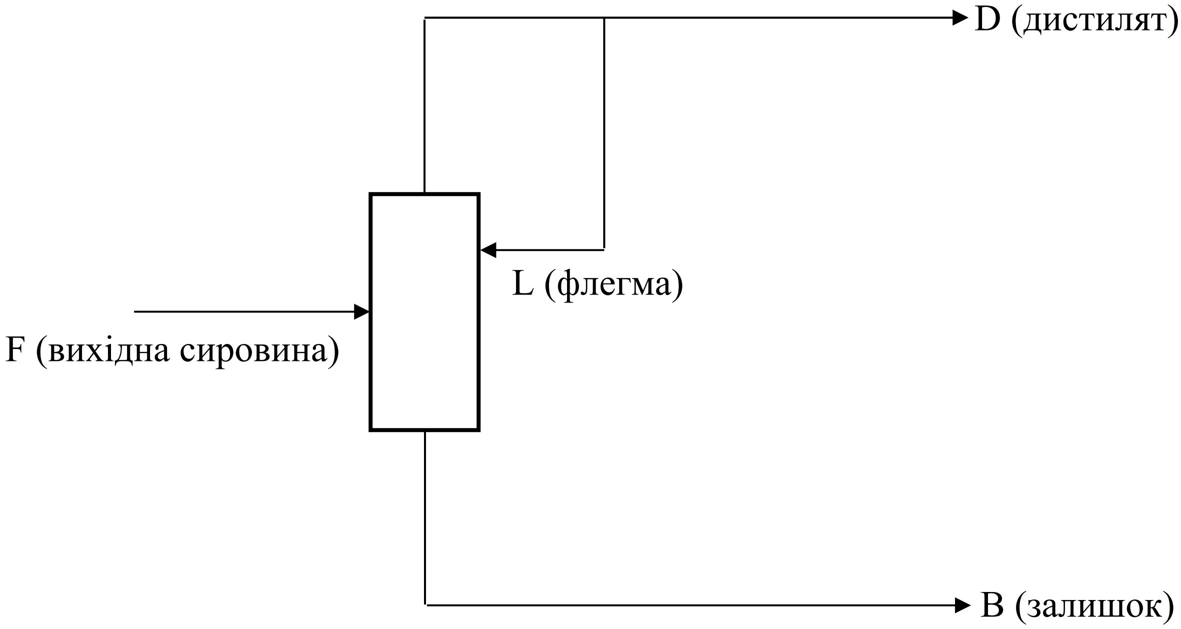

Рисунок А.1- Схема процесу ректифікації

Витрати на перекачку, що забезпечує повторну перегонку, виражаються формулою:

$$C_L = 5000 + 0{,}7 \cdot L$$

Схема процесу ректифікації представлена на рисунку А.1.

Додаток Б

Приклади задач оптимізації при наявності обмежень

Приклад Б.1.

Необхідно знайти мінімум функції:

$$f(x, y) = x^2 + y^2$$

при обмеженні:

$$x + y = 4.$$

Відповідь: $f_{min} = 8$ при $x_{min} = 2$ та $y_{min} = 2$.

Приклад Б.2.

Необхідно знайти мінімум функції:

$$f(x, y) = x^2 + y^2$$

при обмеженні:

$$3 \cdot x^2 + 4 \cdot x \cdot y + 6 \cdot y^2 = 140$$

Відповідь: $f_{min} = 20$.

Приклад Б.3.

Необхідно показати, що функція:

$$f(x, y) = x^2 + y^2$$

при обмеженні:

$$x - y = 5$$

досягає мінімуму при $x = 2{,}5$ та $y = -2{,}5$.

Приклад Б.4.

Необхідно знайти мінімум функції:

$$y = 2 \cdot x^2 - e^x$$

в інтервалі (0; 1).

Відповідь: $x_{min} = 0{,}357403$; $y_{min} = -1{,}174138$.

Приклад Б.5.

Необхідно знайти мінімум функції:

$$y = -e^{-x} \cdot \ln(x)$$

в інтервалі (0; 2).

Відповідь: $x_{min} = 1{,}76322211$; $y_{min} = -0{,}0972601313$.

Приклад Б.6.

Необхідно знайти мінімум функції:

$$y = 2 \cdot x^2 + 3 \cdot e^{-x}$$

в інтервалі (0; 1).

Відповідь: $x_{min} = 0{,}47$.

Приклад Б.7.

Необхідно знайти мінімум функції:

$$y = x^4 - 14 \cdot x^3 + 60 \cdot x^2 - 70 \cdot x$$

в інтервалі (5; 7).

Відповідь: мінімум дорівнює 5,96.

Приклад Б.8.

Необхідно знайти мінімум функції:

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2$$

при обмеженнях:

$$x_1 - 2 \cdot x_2 + 1 = 0$$

$$-0{,}25 \cdot x_1^2 - x_2^2 + 1 \geq 0$$

Відповідь: $f_{min} = 1{,}393$ при $x_{1\,min} = 0{,}823$ та $x_{2\,min} = 0{,}911$.

Приклад Б.9.

Необхідно знайти мінімум функції:

$$f(x)=1000-x_1^2-2\cdot x_2^2-x_3^2-x_1\cdot x_2-x_1\cdot x_3$$

при обмеженнях:

$$x_1^2+x_2^2+x_3^2-25=0$$

$$8\cdot x_1+14\cdot x_2+7\cdot x_3-56=0$$

$$x_1>0$$

$$x_2>0$$

$$x_3>0$$

Відповідь: $f_{min}=961{,}715$ при $x_{1\,min}=3{,}512$; $x_{2\,min}=0{,}217$ та $x_{3\,min}=3{,}552$.

Приклад Б.10.

Необхідно знайти мінімум функції:

$$f(x)=(x_1-2)^2+(x_2-1)^2$$

при обмеженнях:

$$-x_1^2+x_2\geq 0$$

$$-x_1-x_2+2\geq 0$$

Відповідь: $f_{min}=1{,}0$ при $x_{1\,min}=1{,}0$ и $x_{2\,min}=1{,}0$.

Приклад Б.11.

Необхідно знайти мінімум x_3 при обмеженнях:

$$x_1-2\cdot x_2=-1$$

$$0{,}25\cdot x_1^2+x_2^2-1\leq 0$$

$$(x_1-2)^2+(x_2-1)^2-x_3\leq 0$$

$$0\leq x_1\leq 100$$

$$0\leq x_2\leq 100$$

$$0 \le x_3 \le 100$$

Відповідь: $x_{3\,min} = 1{,}3934$ при $x_{1\,min} = 0{,}8228$; $x_{2\,min} = 0{,}9114$.

Приклад Б.12.

Необхідно знайти мінімум функції:

$$f(x) = -4 \cdot (x_1 - 1)^2 - 25 \cdot (x_2 - 2)^2$$

при обмеженнях:

$$8{,}3 \cdot x_1 + 20{,}5 \cdot x_2 \le 170{,}15$$

$$-7{,}5 \cdot x_1 + 18 \cdot x_2 \le 135$$

$$-10{,}5 \cdot x_1 + 7{,}7 \cdot x_2 \le 80{,}85$$

$$-3{,}7 \cdot x_1 - 10{,}2 \cdot x_2 \le 37{,}74$$

$$-2{,}7 \cdot x_1 - 13 \cdot x_2 \le 35{,}1$$

$$4{,}5 \cdot x_1 - 7 \cdot x_2 \le 31{,}5$$

$$-20 \le x_1 \le 20$$

$$-20 \le x_2 \le 20$$

Відповідь: $f_{min} = -871{,}947$ при $x_{1\,min} = 0{,}973$; $x_{2\,min} = 7{,}905$.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ануфриев, И.Е. MATLAB 7 / И.Е. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова.- СПб: БХВ-Петербург, 2005.- 1104с.

2. Дьяконов, В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов.- Питер, 2001.- 488с.

3. Банди Б. Методы оптимизации. Вводный курс / Б. Банди.- М.: Радио и связь, 1988.- 128с.

4. Жовтонога, М.М. Використання програмних продуктів COMSOL MULTIPHYSICS®, MATLAB®, SIMULINK® та SIMSCAPE™ при розв'язані задач гідрогазодинаміки та тепло масообміну: навчальний посібник / М.М. Жовтонога, А.С. Попова, В.О. Перцевий. - Дніпропетровськ: Ліра, 2016.- 108с.

5. Хофер Э. Численные методы оптимизации / Э. Хофер, Р. Лундерштедт.- М.: Машиностроение, 1981.- 192с.

6. Optimal cooling of a tubular reactor [електронний ресурс] - Режим доступу: https://www.comsol.com/model/download/391161/models.chem.optimal_cooling.pdf (дата звернення: 24.06.17). – Назва з екрана.

Підписано до друку 29.01.2018 р. Формат 60х84/16.
Папір офсетний. Друк офсетний. Ум. друк.арк. 6,74.
Наклад 100 пр. Зам. № 14.

Видавництво і друкарня «Ліра».
49107, м. Дніпро, вул. Наукова, 5.
Свідоцтво про внесення до Державного реєстру
ДК № 188 від 19.09.2000