

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра механіки, автоматизації та інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Розробка інформаційної технології розв'язання задач вибухобезпеки»

(тема кваліфікаційної роботи українською мовою)

«Development of information technology for solving explosion safety problems»

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання

спеціальності 126 Інформаційні системи та технології

(код, назва спеціальності)

Освітня програма «Інформаційні системи та технології»

(назва)

Обертінський Олексій Олексійович

(прізвище, ім'я, по-батькові здобувача)

Керівник д.т.н., проф. Волков В.Е.

(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент ст.викл. Косирева Л.А.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ ____ від ____ . ____ . 20__ р.

Завідувач(ка) кафедри

Алла РАЧИНСЬКА

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК № ____

протокол № ____ від ____ . ____ . 20__ р.

Оцінка ____ / ____ / ____

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

Олена АРСІРІЙ

(підпис)

(прізвище, ім'я)

Одеса 2025

АНОТАЦІЯ

Дипломна робота присвячена дослідженню та впровадженню інформаційних технологій у сфері забезпечення вибухобезпеки. Основна увага приділена розробці програмного застосунку для розрахунку основних параметрів детонаційного вибуху газових сумішей з метою оперативної оцінки ризиків виникнення детонації і підвищення рівня безпеки на вибухонебезпечних об'єктах.

У рамках роботи проведено аналіз фізико-хімічних процесів, що супроводжують детонаційний вибух, досліджено основні чинники, які впливають на характер і силу вибуху. Здійснено аналіз предметної області, визначено вимоги до функціоналу застосунку та розроблено математичну модель для обчислення швидкості детонації та термодинамічних параметрів її продуктів.

Реалізацію програмного забезпечення виконано з використанням мови програмування C# та інструментів Windows Forms, що входять до складу платформи Microsoft .NET Framework. Архітектура застосунку побудована відповідно до принципів структурного програмування, без використання зовнішніх баз даних, з акцентом на легкість, автономність і швидкість роботи.

Результатом дослідження є створення програмного продукту з графічним інтерфейсом, який дає змогу інженерам, експертам і дослідникам швидко отримувати необхідні дані для оцінювання небезпечних умов. Застосунок дає можливість здійснювати вибір типу пального та окислювача, дозволяє вводити початкові параметри суміші, а також автоматично обчислює вихідні характеристики.

Отримані результати можуть бути використані у навчальному процесі, на підприємствах хімічної, нафтової, газової та енергетичної галузей, а також у структурах, що займаються техногенною безпекою.

ABSTRACT

The thesis is devoted to the research and implementation of information technologies in the field of explosion safety. The main focus is placed on the development of a software application for calculating the key parameters of gas-air mixture explosions in order to enable rapid risk assessment and enhance safety at facilities with increased explosion hazards.

Within the scope of the work, an analysis of the physico-chemical processes associated with explosions was carried out, and the main factors influencing the nature and intensity of explosions were investigated. The subject area was analyzed, the functional requirements for the application were defined, and a mathematical model was developed to calculate pressure, density, temperature, sound velocity, and other critical detonation parameters.

The software was implemented using the C# programming language and Windows Forms tools, which are part of the Microsoft .NET Framework. The application architecture follows the principles of structured programming, does not rely on external databases, and emphasizes simplicity, autonomy, and fast performance.

The result of the study is a software product with a graphical user interface that allows engineers, experts, and researchers to quickly obtain the necessary data for assessing hazardous conditions. The application supports the selection of fuel and oxidizer types, enables the input of initial environmental and mixture parameters, and automatically calculates the resulting characteristics.

The developed solution can be used in educational settings, by enterprises in the chemical, oil, gas, and energy industries, as well as by organizations involved in industrial and technological safety.

ЗМІСТ

ВСТУП	6
1. ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	8
1.1 Введення у предметну область	8
1.2 Аналіз аналогів: висновок	9
1.3 Постановка задачі	10
2. МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ	12
2.1 Теоретичні основи розрахунку параметрів газових сумішей	12
2.2 Основні закономірності класичного прямого скачку в газі без притоку енергії	23
2.3 Основні закономірності прямого скачку з притоком енергії та співвідношенням між різними режимами горіння	36
2.4 Концентраційні межі вибуховості горючих речовин	58
3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	60
3.1 Архітектура застосунку	60
3.2 Інформаційне моделювання предметної області	60
3.3 Вибір програмного забезпечення для створення інформаційної системи	62
3.4 Програмна модель застосунку	64
4. ПРОГРАМНА РЕАЛІЗАЦІЯ	66
4.1 Загальний опис	66
4.2 Інтерфейс користувача	67
4.3 Функціональність програми	68
4.4 Обробка помилок введення	71
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТОК А	78

СПИСОК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

1. НКМВ — Нижня концентраційна межа вибуховості — мінімальна концентрація горючої речовини у повітрі, за якої можлива її детонація або займання.
2. ВКМВ — Верхня концентрація межа вибуховості — максимальна концентрація горючої речовини, за якої суміш ще здатна вибухати.
3. ЧЖ (Режим Чепмена–Жуге) — стан, за якого хвиля детонації поширюється з мінімально можливою швидкістю при $u_2 = a_2$.
4. Сильна детонація — Режим, за якого швидкість детонаційної хвилі перевищує нормальне значення.
5. Слабка детонація — Режим, за якого хвиля поширюється з меншою швидкістю, ніж у ЧЖ-режимі.
6. Дефлаграція — Повільне поширення горіння з дозвуковою швидкістю.
7. Політропний газ — це модель газу, в якій під час термодинамічного процесу виконується політропне співвідношення між тиском і об'ємом.
8. GUI (Graphical User Interface) — графічний інтерфейс користувача.
9. p, p_1, p_2 — Тиск, тиск до та після стрибка відповідно (Па).
10. T, T_1, T_2 — Температура, температура до та після стрибка (К).
11. ρ, ρ_1, ρ_2 — Густина суміші до та після стрибка (кг/м^3).
12. u — Швидкість потоку газу (м/с).
13. a — Швидкість звуку у середовищі (м/с).
14. q — Питома теплота хімічної реакції, тепловиділення (Дж/кг).
15. γ — Адіабатичний показник (відношення теплоємностей c_p/c_v).
16. R — Питома газова стала (Дж/кг·К).
17. M — Число Маха — відношення швидкості потоку до швидкості звуку.

ВСТУП

Вибухи газових сумішей часто виникають на різноманітних об'єктах промисловості та транспорту. Насамперед випадкові вибухи характерні для хімічної, енергетичної, нафтогазової галузей [1-4].

Двома головними проблемами забезпечення вибухобезпеки є

- 1) проблема передбачення виникнення вибухів та своєчасного реагування на можливість виникнення вибухонебезпечних ситуацій [1,2,5];
- 2) проблема оцінювання наслідків вибухів [3,4].

Помилки у розрахунках чи недооцінка ризиків вибуху можуть призводити до техногенних катастроф з людськими жертвами та великими збитками. На жаль, частота виникнення подібних ситуацій лише зростає через велику кількість об'єктивних та суб'єктивних факторів – ускладнення технологічних процесів, появу нових горючих та паливно-мастильних матеріалів, хімізацію виробництва, порушення норм експлуатації та недостатньо високу кваліфікацію персоналу (людський фактор).

Враховуючи усі перелічені вище проблеми, використання сучасних інформаційних технологій як інструменту для підвищення рівня вибухобезпеки стає все більш актуальним. Особливе значення мають програмні засоби, які дозволяють виконувати швидкий і точний розрахунок параметрів вибуху в залежності від типу газової суміші, концентрації пального та окислювача, температури, тиску та інших умов.

Одним з ефективних рішень у таких випадках є розробка спеціалізованого програмного забезпечення, що дозволяє моделювати сценарії вибуху, враховуючи фізико-хімічні параметри вибухонебезпечного середовища. Такі програми мають бути зручними для користувача, простими в експлуатації, зрозумілими не тільки для інженерів та фахівців з безпеки, але й для людей з середньою освітою, які можуть працювати операторами технологічних процесів, та не вимагати наявності занадто великих зовнішніх баз даних.

У даній роботі реалізовано програмний застосунок, що дозволяє вводити початкові параметри газоповітряної суміші (температура, тиск, концентрація компонентів тощо) та автоматично виконувати розрахунки основних характеристик вибуху: тиску, температури, швидкості звуку, густини і т. д. Програму створено засобами C# та Windows Forms, що входять до складу платформи Microsoft .NET Framework. Архітектура застосунку передбачає простоту, автономність, високу швидкодію та точність розрахунків.

Дана розробка орієнтована як на практичне використання в умовах виробництва, так і на навчальний процес, де потрібне математичне та програмне моделювання реальних вибухонебезпечних ситуацій. Крім того, програмне забезпечення може бути розширене у майбутньому – наприклад, з додаванням відповідних баз даних, графіків, інтерфейсу API чи підключенням сенсорних даних у реальному часі.

Отже, метою роботи є створення програмного засобу для розрахунку параметрів вибуху газових сумішей, що дозволяє підвищити ефективність оцінки ризиків при виникненні вибуху в техногенних умовах.

Для реалізації поставленої мети необхідно розв'язати такі підзадачі:

- 1) провести аналіз предметної області з урахуванням основних характеристик та фізико-хімічних властивостей, що мають значення для розрахунку параметрів вибуху;
- 2) визначити вимоги до функціональності та точності майбутнього застосунку;
- 3) розробити математичну модель розрахунків параметрів вибуху;
- 4) створити програмне забезпечення за допомогою Windows Forms на мові програмування C#;
- 5) здійснити тестування точності обчислень та зручності інтерфейсу користувача;
- 6) проаналізувати можливі напрями подальшого вдосконалення застосунку.

1 ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Введення у предметну область

Предметна область, що досліджується в даній роботі, охоплює аналіз вибухових процесів, пов'язаних із горючими газоповими сумішами, а також методи їх математичного моделювання з використанням сучасних інформаційних технологій. Знання про поведінку таких сумішей в умовах змінного тиску, температури та концентрації компонентів є критично важливими для оцінки ризиків виникнення та наслідків вибуху і запобігання надзвичайним ситуаціям на промислових та транспортних об'єктах.

До вибухонебезпечних газів відносяться, зокрема, метан, водень, пропан, ацетилен, а також деякі пари летких речовин. Їх змішування з повітрям у певних концентраціях створює суміші, здатні до вибуху при наявності джерела займання. Основними параметрами, які характеризують такі суміші, є: нижня і верхня межі вибуховості, температура займання, тиск та густина.

У сучасних умовах виникає необхідність в автоматизації процесів аналізу вибухонебезпечних сумішей, оскільки ручні розрахунки або використання застарілих методик можуть бути недостатньо ефективними або навіть помилковими. Це особливо важливо у випадках, коли необхідно оперативно оцінити умови безпеки при проектуванні, експлуатації або зберіганні вибухонебезпечних матеріалів.

У зв'язку з цим розробляється інформаційна система, яка має на меті забезпечити виконання таких завдань:

- 1) автоматизований розрахунок основних параметрів вибуху газових сумішей;
- 2) зручне введення початкових даних (тип газу, температура, тиск, концентрація тощо);

- 3) перевірка відповідності заданих параметрів межах вибуховості;
- 4) виведення результатів на екран у зручній для користувача формі (таблиці, числові значення, можливе графічне представлення);
- 5) підвищення обізнаності користувача щодо потенційної вибухонебезпеки суміші.

Користувач інформаційної системи (інженер, технолог, викладач або студент) виконує наступні дії:

- 1) обирає тип газу або паливної речовини зі списку;
- 2) задає вхідні параметри: температуру, тиск та концентрацію суміші;
- 3) запускає розрахунок;
- 4) отримує результати у вигляді значень тиску вибуху, густини та температури продуктів вибуху, швидкості звуку в продуктах та, якщо потрібно, інших характеристик;
- 5) використовує отримані дані для подальшого аналізу, проєктування або навчання.

1.2 Аналіз аналогів

У ході аналізу наявних аналогів було розглянуто кілька програмних рішень, що дозволяють виконувати розрахунки параметрів вибуху газових сумішей або пов'язані із загальними задачами хімічної безпеки. Серед них — як академічні програми для моделювання термодинамічних процесів, так і комерційні продукти з вузькою спеціалізацією.

Проте більшість з існуючих рішень мають низку суттєвих недоліків [6,7]:

- 1) складний інтерфейс, не орієнтований на широке коло користувачів;
- 2) обмеженість у виборі початкових параметрів;
- 3) відсутність можливості виконання швидкого аналізу без глибоких знань з хімії чи термодинаміки;

4) відсутність автономності — багато рішень потребують підключення до зовнішніх баз даних або серверів;

5) закритий вихідний код або відсутність можливості адаптації під специфічні вимоги.

Крім того, деякі з програм орієнтовані виключно на професіоналів з галузі хімічного моделювання, що ускладнює їх використання у навчальному процесі або в умовах обмеженого часу на аналіз ситуації.

З огляду на виявлені недоліки аналогів, було зроблено висновок про доцільність створення власного програмного засобу, який:

- 1) буде мати простий та інтуїтивно зрозумілий інтерфейс;
- 2) дозволить швидко вводити початкові параметри та отримувати результати;
- 3) працюватиме автономно без необхідності підключення до Інтернету;
- 4) забезпечить точні розрахунки на основі фізико-хімічних моделей;
- 5) буде придатним як для використання у виробничих умовах, так і в освітньому середовищі.

Це обґрунтовує актуальність та практичну цінність розробки запропонованого застосунку в межах даної кваліфікаційної роботи.

1.3 Постановка задачі

На основі аналізу предметної області, вивчення фізико-хімічних властивостей газоповітряних сумішей, а також результатів аналізу аналогічних програмних рішень, сформульовано основну задачу, яку має розв'язати дане дослідження — створення інформаційної системи, що дозволяє виконувати розрахунок параметрів вибуху газових сумішей у заданих умовах середовища.

Інформаційна система повинна забезпечити зручний інтерфейс для користувача, можливість введення основних початкових параметрів, реалізацію математичної моделі розрахунку основних вибухових

характеристик (тиску вибуху, густини, температури, швидкості звуку тощо), а також оперативний вивід результатів на екран монітора.

Основна задача складається з таких підзадач:

1. Формалізація вхідних даних — визначення параметрів, що характеризують газоповітряну суміш, та обмеження їхніх припустимих значень.
2. Розробка математичної моделі — опис формул та алгоритмів, що використовуються для обчислення фізичних характеристик вибуху на основі вхідних даних.
3. Проектування архітектури програми — вибір технологій, структурна організація компонентів застосунку, логіка їх взаємодії.
4. Розробка графічного інтерфейсу користувача (GUI) — забезпечення зручного способу введення даних, відображення результатів і навігації.
5. Реалізація програмного коду — створення основної функціональності засобами мови програмування C# та технології Windows Forms.
6. Тестування і підтвердження результатів — перевірка точності обчислень, відповідність результатів теоретичним і довідковим даним.
7. Оцінка перспектив розвитку — визначення можливостей інтеграції з іншими системами, розширення функціоналу, додавання підтримки бази даних, графіків або API.

Метою розробки є створення програмного забезпечення, що дозволяє:

- 1) зменшити час і складність виконання інженерних розрахунків;
- 2) підвищити точність і достовірність оцінки вибухових параметрів;
- 3) підвищити рівень безпеки на об'єктах, де використовуються або зберігаються вибухонебезпечні речовини.

2 МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ

У розрахунках параметрів вибуху газових сумішей особливу роль відіграють точні фізико-хімічні моделі. Це стосується не лише самого процесу горіння, а й побудови математичних залежностей для визначення питомої газової сталої суміші, густини, парціальних тисків компонентів тощо. У даному розділі наведено ключові рівняння термодинаміки ідеальних газів, а також методи розрахунку питомої теплоти згорання. Ця теорія має пряме прикладне значення і реалізована в програмному забезпеченні, описаному в наступних розділах.

2.1 Теоретичні основи розрахунку термодинамічних параметрів газових сумішей

Перед побудовою математичної моделі вибухового процесу необхідно розглянути теоретичні основи термодинаміки сумішей ідеальних газів. Особливо важливими для подальших розрахунків є рівняння стану, визначення питомої газової сталої суміші, а також методи розрахунку теплоти згорання. Це дозволяє перейти до точнішої моделі розрахунку вибуху, що враховує як хімічний склад, так і фізичні властивості компонентів суміші.

Для початку розглянемо систему, що складається з n ($n \geq 2$) ідеальних газів. Припустимо, що суміш займає об'єм V , а її абсолютна температура (в Кельвінах) становить T . Вважатимемо, що кожен компонент — i -ий газ — у складі суміші має парціальний тиск p_i ($i=1, \dots, n$).

Очевидно, кожен з компонентів займає увесь доступний об'єм V і має одну і ту саму температуру T . Таким чином, термодинамічний стан кожного газу у суміші характеризується такими параметрами:

$$p_1, V, T;$$

$$p_2, V, T;$$

.....

$$p_n, V, T.$$

Оскільки всі гази вважаються ідеальними, кожен з них задовольняє рівняння стану ідеального газу (Клапейрона):

$$p_i V = m_i R_i T \quad (i = 1, \dots, n), \quad (2.1)$$

де m_i це маса i -го газу в суміші та R_i – його питома газова стала.

У формулі (2.1) враховується маса компонента, його питома газова стала, що обчислюється на основі універсальної газової сталої та молярної маси відповідної речовини. Значення універсальної газової сталої становить

$R_{\mu} = 8,31$ (Дж/моль·К), де μ_i — молярна маса i -го газу.

Маса суміші визначається як сума мас усіх складових компонентів.

$$m = \sum_{i=1}^n m_i \quad (2.2)$$

В такому разі рівняння стану можна записати вже для всієї газової суміші, використовуючи не індивідуальні значення R_i , а узагальнену величину — питому газову сталу суміші. Це значення визначається складом суміші та долею кожного з її компонентів

$$pV = mR_{\text{см}}T, \quad (2.3)$$

де

$$R_{\text{см}} = \frac{R_{\mu}}{\mu_{\text{см}}} \quad (2.4)$$

($R_{\text{см}}$ – питома газова стала суміші, $\mu_{\text{см}}$ – молярна маса суміші).

Окрім цього, можна використати цю формулу та підставити замість маси - густину суміші або її питомий об'єм. У такому випадку тиск виражається через густину, газову сталу та температуру, що є зручним у практиці інженерних обчислень.

На цій основі рівняння записується у такій формі:

$$pv_{\text{см}} = R_{\text{см}}T \quad (2.5)$$

або

$$p = \rho_{\text{см}} R_{\text{см}} T, \quad (2.6)$$

де $v_{\text{см}} = V/m$ — питомий об'єм суміші, $\rho_{\text{см}} = 1/v_{\text{см}} = m/V$ — густина суміші, а P — тиск суміші.

Для розрахунку ρ , згідно з формулою 2.6, ми можемо розрахувати її наступним чином:

$$\rho = \frac{P}{R_{\text{см}} T}, \quad (2.7)$$

де

$$R_{\text{см}} = \frac{1}{\left(\frac{r_1}{R_1}\right) + \left(\frac{r_2}{R_2}\right)} \quad (2.8)$$

Для визначення термодинамічних властивостей газової суміші необхідно провести:

1. Розрахунок питомої газової сталої або, що еквівалентно, середньої молярної маси суміші;
2. Розрахунок густини або питомого об'єму газової системи;
3. За потреби — обчислення парціальних тисків окремих складників.

Варто усвідомлювати, що парціальний тиск конкретного газу — це тиск, який він би створював, займаючи весь об'єм суміші при фіксованій температурі. Відповідно до закону Дальтона, загальний тиск суміші є сумою парціальних тисків усіх компонентів.

Газовий склад може бути описаний через масові, об'ємні або мольні частки. Ці характеристики використовуються в подальших обчисленнях і є основою при розрахунках тепловиділення та при визначенні вибухових властивостей сумішей.

Якщо газова суміш задається масовими долями, то використовуються наступні формули:

$$g_i = \frac{m_i}{m} \quad (i = 1, \dots, n),$$

$$\mu_{\text{см}} = \frac{\sum_{i=1}^n m_i}{\sum_{i=1}^n M_i} \quad (2.16)$$

або

$$\mu_{\text{см}} = \frac{m}{M} \quad (2.17)$$

Для розрахунку парціальних тисків можна скористуватися рівнянням

$$p = p \frac{m_i}{m} \frac{R_i}{R_{\text{см}}} \quad (i = 1, \dots, n) \quad (2.18)$$

або

$$p = p g_i \frac{R_i}{R_{\text{см}}} \quad (i = 1, \dots, n) \quad (2.19)$$

Газова суміш, що задана об'ємними долями :

Об'ємна доля r_i кожного i -го компонента суміші визначається як відношення його парціального (приведеного) об'єму V_i цього газу до загального об'єму суміші V :

$$r_i = \frac{V_i}{V} \quad (i = 1, \dots, n) \quad (2.20)$$

При цьому під парціальним (приведеним) об'ємом V_i i -го газу називають об'єм, розуміють той об'єм, що займав би даний газ за температури T і тиску p , що дорівнюють температурі та тиску суміші газів.

Парціальний об'єм кожного газу може бути визначений на основі закону Бойля-Маріотта. При сталої температурі T маємо:

$$\left\{ \begin{array}{l} p_1 V = p V_1 \\ p_2 V = p V_2 \\ \dots\dots\dots \\ p_n V = p V_n \end{array} \right. \quad (2.21)$$

Враховуючи рівняння (2.21) маємо:

$$\left\{ \begin{array}{l} V_1 = \frac{p_1 V}{p} \\ V_2 = \frac{p_2 V}{p} \\ \dots\dots\dots \\ V_n = \frac{p_n V}{p} \end{array} \right. \quad (2.22)$$

Після підсумовування всіх рівнянь системи (2.22) отримаємо:

$$\sum_{i=1}^n V_i = \frac{\sum_{i=1}^n p_i}{p} V \quad (2.23)$$

З урахуванням закону Дальтона (2.6) маємо наступне

$$\sum_{i=1}^n V_i = V \quad (2.24)$$

В результаті рівняння (2.9) випливає, що

$$\sum_{i=1}^n r_i = 1 \quad (2.25)$$

Між питомими об'ємами, густинами, молярними масами та питомими газовими сталими для окремого i -го газу та для суміші в цілому існує така залежність:

$$\frac{\rho_i}{\rho_{\text{см}}} = \frac{v_{\text{см}}}{v_i} = \frac{\mu_i}{\mu_{\text{см}}} = \frac{R_i}{R_{\text{см}}} (i = 1, \dots, n) \quad (2.26)$$

Рівняння (2.26) є слідством рівняння стану ідеального газу (рівняння Клапейрону) та закону Авогадро.

Отже, можна записати:

$$g_i = \frac{m_i}{m} = \frac{\rho_i V_i}{\rho_{\text{см}} V} = \frac{\rho_i}{\rho_{\text{см}}} r_i (i = 1, \dots, n), \quad (2.27)$$

$$r = \frac{\rho_{\text{см}}}{\rho_i} g_i = \frac{v_i}{v_{\text{см}}} g_i = \frac{R_i}{R_{\text{см}}} g_i = \frac{\mu_{\text{см}}}{\mu_i} g_i (i = 1, \dots, n), \quad (2.28)$$

$$g_i = \frac{\rho_i}{\rho_{\text{см}}} r_i = \frac{\mu_i}{\mu_{\text{см}}} r_i = \frac{R_{\text{см}}}{R_i} r_i (1, \dots, n) \quad (2.29)$$

З виразів (2.9), (2.26), (2.27), (2.28) випливає, що

$$g_i = \frac{R_{\text{см}}}{R_i} r_i(1, \dots, n), \quad (2.30)$$

$$\sum_{i=1}^n g_i = R_{\text{см}} \sum_{i=1}^n \frac{r_i}{R_i} = 1 \quad (2.31)$$

Отже

$$R_{\text{см}} = \frac{1}{\sum_{i=1}^n \frac{r_i}{R_i}} \quad (2.32)$$

З (2.16) отримуємо

$$\sum_{i=1}^n m_i = \mu_{\text{см}} \sum_{i=1}^n M_i \quad (2.33)$$

Маса m_i i -го газу в суміші обчислюється за рівнянням

$$m_i = \mu_i M_i (i = 1, \dots, n) \quad (2.34)$$

Із рівнянь (2.34) та (2.16) випливає наступне

$$\sum_{i=1}^n \mu_i M_i = \mu_{\text{см}} M \quad (2.35)$$

Поділивши обидві частини отриманого виразу на M , отримаємо

$$\mu_{\text{см}} = \sum_{i=1}^n \mu_i \frac{M_i}{M} \quad (2.36)$$

Як наслідок із закону Авогадро, маємо

$$\frac{M_i}{M} = r_i (i = 1, \dots, n) \quad (2.37)$$

Подальший аналіз рівнянь (2.36) і (2.37) дозволяє дійти до висновку, що

$$\mu_{\text{см}} = \sum_{i=1}^n \mu_i r_i \quad (2.38)$$

Для обчислення парціальних тисків у випадку, коли газова суміш задана через об'ємні частки, застосовуються рівняння Бойля–Маріотта, з яких випливає наступне

$$p_i = p r_i (i = 1, \dots, n) \quad (2.39)$$

Газова суміш, може бути задана мольними частками (мольним складом):

Мольна частка y_i для кожного i -го газу в суміші визначається як відношення кількості молів M_i цього газу до сумарної кількості молів у суміші M :

$$y_i = \frac{M_i}{M} \quad (i = 1, \dots, n) \quad (2.40)$$

Очевидно, що:

$$\sum_{i=1}^n M_i = M \quad (2.41)$$

При цьому

$$\sum_{i=1}^n y_i = 1 \quad (2.42)$$

Нескладно довести, що завдання складу суміші через мольні частки є еквівалентним опису її об'ємними частками.

Дійсно, відповідно до закону Авогадро, за однакових температур і тисків усі гази мають однаковий молярний об'єм, а отже:

$$V_i = V_\mu M_i \quad (i = 1, \dots, n), \quad (2.43)$$

де V_μ – молярний об'єм суміші.

Звідси випливає:

$$V = V_\mu M \quad (2.44)$$

Аналізуючи рівняння (2.44) і (2.43), можна зробити висновок (поділивши одне на інше), що:

$$\frac{V_i}{V} = \frac{M_i}{M} \quad (i = 1, \dots, n) \quad (2.45)$$

або

$$r_i = y_i \quad (i = 1, \dots, n) \quad (2.46)$$

Тобто об'ємна частка дорівнює мольній частці при сталих температурі та тиску.

Таким чином, встановивши еквівалентність мольної та об'ємної часток за рівних умов, можемо перейти до розгляду одного з ключових параметрів, що характеризує енергетичний потенціал газової суміші — питомої теплоти згоряння.

Теплота згоряння, віднесена до одиниці маси або об'єму пального, називається питомою теплотою згоряння. Вона вимірюється у Дж/кг або Дж/м³ (також можуть застосовуватись одиниці кал/кг і кал/м³). Для твердих видів палива, як правило, питомою вважається теплота згоряння, віднесена до одиниці маси (Дж/кг), а для газів частіше використовується теплота згоряння, віднесена до одиниці об'єму (Дж/м³).

У нашому випадку під теплотою згоряння Q будемо розуміти повну кількість тепла, що виділяється при згорянні газової суміші — як стехіометричної, так і нестехіометричної. При цьому припускається, що газова суміш складається лише з двох компонентів: горючого газу та окислювача (яким виступає або чистий кисень, або атмосферне повітря).

Питомою теплотою згоряння q будемо називати відношення повної теплоти згоряння Q до маси всієї суміші ($q = \frac{Q}{m}$, m – маса суміші) або до її

об'єму ($q = \frac{Q}{V}$, V – об'єм суміші). В конкретному контексті завжди буде зрозуміло, про який саме вид питомої величини йдеться. Більш природним на перший погляд здається вважати питомою теплотою згоряння газової суміші саме відношення теплоти згоряння до об'єму суміші, бо газові суміші задаються зазвичай в об'ємних долях або в мольних долях (що є тим самим, бо, як доведено вище, об'ємні долі газів в суміші дорівнюють їх мольним долям). Проте в рамках математичного опису процесів горіння та вибуху — зокрема, у системах рівнянь газової динаміки (Ейлера, неперервності,

енергетичного балансу та хімічної кінетики) — використовується саме питома теплота згоряння Q , віднесена до маси суміші

Розглянемо стехіометричне згоряння газової суміші, де маса пального компонента дорівнює m_1 , а маса окислювача – m_2 . Тоді загальна маса суміші становить:

$$m = m_1 + m_2 \quad (2.47)$$

Під стехіометричним складом горючої суміші розумітимемо таке співвідношення між компонентами, за якого кількість окислювача точно відповідає кількості, необхідній для повного окиснення пального — без його надлишку. Інакше кажучи, стехіометрична суміш — це така суміш, у якій співвідношення пального та окислювача забезпечує повне згоряння палива без залишку незреагованого кисню (як чистого, так і в складі повітря).

Теплота згоряння Q та відповідна їй питома теплота згоряння q для даної суміші вважаються відомими величинами, причому

$$q = \frac{Q}{m} = \frac{Q}{m_1 + m_2} \quad (2.48)$$

Введемо такі умовні позначення

$$x\% = \frac{m_1}{m} \cdot 100\% = \frac{m_1}{m_1 + m_2} \cdot 100\% \quad , \quad (2.49)$$

$$y\% = \frac{m_2}{m} \cdot 100\% = \frac{m_2}{m_1 + m_2} \cdot 100\% \quad (2.50)$$

Тобто $x\%$ та $y\%$ являють собою відсотковий вміст (за масою) відповідно горючого газу та окислювача у горючій газовій суміші. При цьому

$$g_1 = \frac{m_1}{m} = \frac{m_1}{m_1 + m_2} = \frac{x\%}{100\%} \quad , \quad (2.51)$$

$$g_2 = \frac{m_2}{m} = \frac{m_2}{m_1 + m_2} = \frac{y\%}{100\%} \quad , \quad (2.52)$$

$$x\% + y\% = 100\%, \quad (2.53)$$

$$g_1 + g_2 = 1 \quad (2.54)$$

Припустимо, що до цієї стехіометричної суміші додано ще певну кількість пального (горючого газу) масою m'_1 . Тоді загальна маса суміші стає рівною $m_1 + m_2 + m'_1 = m + m'_1$ і суміш перестає бути стехіометричною (суміш, в якій вміст горючого газу більший ніж в стехіометричній, називається багатомішню).

Звичайно, що при згорянні такої нестехіометричної суміші виділяється та сама кількість тепла Q , що й при згорянні відповідної стехіометричної суміші (частка горючого газу m'_1 лишається «невикористаною»). Тобто теплота згорання Q при цьому не змінюється.

Але питома теплота згорання q'_1 при цьому зменшується і приймає значення

$$q'_1 = \frac{Q}{m + m'_1} = \frac{Q}{m_1 + m_2 + m'_1} \quad (2.55)$$

Введемо нові умовні позначення

$$x'\% = \frac{m_1 + m'_1}{m + m'_1} \cdot 100\% = \frac{m_1 + m'_1}{m_1 + m_2 + m'_1} \cdot 100\% \quad (2.56)$$

$$y'\% = \frac{m_2}{m + m'_1} \cdot 100\% = \frac{m_2}{m_1 + m_2 + m'_1} \cdot 100\% \quad (2.57)$$

Тобто величини $x'\%$ та $y'\%$ характеризують масову частку відповідно горючого газу та окислювача у складі стехіометричної газової суміші. При цьому виконується:

$$g'_1 = \frac{m_1 + m'_1}{m + m'_1} = \frac{m_1 + m'_1}{m_1 + m_2 + m'_1} = \frac{x'\%}{100\%}, \quad (2.58)$$

$$g'_2 = \frac{m_2}{m + m'_1} = \frac{m_2}{m_1 + m_2 + m'_1} = \frac{y'\%}{100\%}, \quad (2.59)$$

$$x'\% + y'\% = 100\%, \quad (2.60)$$

$$g'_1 + g'_2 = 1 \quad (2.61)$$

Тут g'_1 та g'_2 є масовими долями відповідно горючого газу та окислювачу у нестехіометричній газовій суміші.

На основі викладених термодинамічних положень переходять до розгляду механізмів хвильових явищ, які супроводжують вибух — зокрема, прямого стрибка ущільнення в газовому середовищі без зовнішнього підведення енергії.

2.2 Основні закономірності класичного прямого скачку в газі без притоку енергії

Формальна газотермодинамічна сторона теорії ударних хвиль, а точніше — теорії прямого стрибка в середовищі (газі), що стискається, широко відома і полягає у встановленні основних закономірностей переходу через стаціонарну площину розриву, нормальну до напрямку швидкості течії середовища. Площина прямого стрибка зображена на рис. 2.1

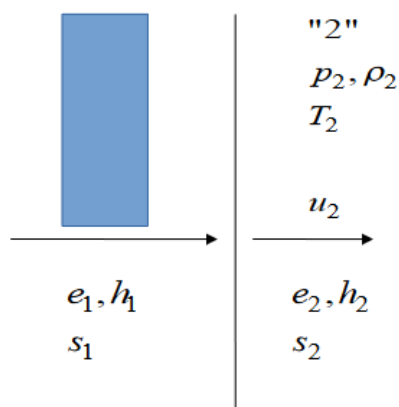


Рис 2.1 – Площина прямого стрибка в середовищі, що стискається (газі)

Параметри газу по різні боки від поверхні стрибка позначимо індексами "1" і "2" (індекс "1" відповідає вихідному газу, а індекс "2" - газу, перетвореному на стрибку, але має той же хімічний склад, що і газ "1". Рух розглядається в рухомій системі координат, пов'язаної безпосередньо з самим стрибком: у цій системі координат поверхня стрибка спочиває, газ «1» протікає через поверхню стрибка зі швидкістю u_1 , а газ «2» витікає з поверхні стрибка зі швидкістю u_2 . При цьому стрибок зазнають всіх термодинамічних параметрів середовища – тиск p , щільність ρ , абсолютна температура T , питома

внутрішня енергія e і питома ентальпія (тепловміст) h , а також питома ентропія s .

Параметри з обох боків поверхні стрибка пов'язані трьома основними законами збереження – маси (кількості речовини), імпульсу (кількості руху) та енергії.

Закон збереження маси має вигляд

$$\rho_1 u_1 = \rho_2 u_2 \quad (2.62)$$

і виражає сталість щільності потоку маси

$$j = \rho u = \rho_1 u_1 = \rho_2 u_2 = \text{const} \quad (2.62.a)$$

під час переходу через стрибок.

Закон збереження імпульсу має вигляд

$$p_1 + \rho_1 u_1^2 = p_2 + \rho_2 u_2^2, \quad (2.63)$$

а закон збереження енергії можна представити як

$$h_1 + \frac{u_1^2}{2} = h_2 + \frac{u_2^2}{2} \quad (2.64)$$

або

$$e_1 + \frac{p_1}{\rho_1} + \frac{u_1^2}{2} = e_2 + \frac{p_2}{\rho_2} + \frac{u_2^2}{2}, \quad (2.64.a)$$

враховуючи, що між питомою ентальпією та питомою внутрішньою енергією існує зв'язок, виражений формулою

$$h = e + \frac{p}{\rho} = e + pv,$$

де $v = \frac{1}{\rho}$ – питомий обсяг середовища.

Використовуючи той факт, що при $k = 1,2$

$$v_k = \frac{1}{\rho_k}, \quad \rho_k = \frac{1}{v_k}, \quad (2.65)$$

щільність потоку речовини j через поверхню стрибка можна подати у вигляді

$$j = \frac{u_1}{v_1} = \frac{u_2}{v_2}, \quad (2.66)$$

звідки

$$u_1 = jv_1, \quad u_2 = jv_2 \quad (2.67)$$

Переходячи від густини ρ до питомого обсягу v у рівнянні (2.7.a), представимо це рівняння у вигляді

$$p_1 + \frac{u_1^2}{v_1} = p_2 + \frac{u_2^2}{v_2} \quad (2.68)$$

або з урахуванням (2.64.a),

$$p_1 + j^2 v_1 = p_2 + j^2 v_2 \quad (2.69)$$

З формули 2.5 безпосередньо випливає, що

$$j^2 = \frac{p_2 - p_1}{v_1 - v_2} \quad (2.70)$$

Цілком очевидно, що так, як величини $p_2 - p_1$, $\rho_2 - \rho_1$ і, відповідно, $v_1 - v_2$ мають однакові знаки, величина j^2 може бути негативною. При цьому

$$j = \sqrt{\frac{p_2 - p_1}{v_1 - v_2}} \quad (2.71)$$

Розглянемо тепер рівняння (2.63), яке виражає закон збереження енергії. Безпосередньо з (2.64) випливає, що

$$h_2 - h_1 = \frac{1}{2}(u_1^2 - u_2^2) \quad (2.72)$$

Підставляючи (2.72) вирази (2.67) для u_1, u_2 , маємо

$$h_2 - h_1 = \frac{1}{2}j^2(v_1^2 - v_2^2) \quad (2.73)$$

Після нескладних перетворень, замінюючи (2.71) j^2 величиною $\frac{p_2 - p_1}{v_1 - v_2}$ з (2.68), маємо

$$h_2 - h_1 = \frac{1}{2}(p_2 - p_1)(v_1 + v_2) \quad (2.74)$$

Враховуючи, що при $k = 1, 2$

$$h_k = e_k + \frac{p_k}{\rho_k} = e_k + pv_k, \quad (2.75)$$

рівняння (2.74) можна також записати у вигляді

$$e_2 - e_1 = \frac{1}{2}(p_2 + p_1)(v_1 - v_2) \quad (2.76)$$

Рівняння (2.74) або (2.76) добре відомо в літературі, як рівняння Гюгоніо (Hugoniot), а умови (2.1-3), що виражають закони збереження на поверхні стрибка, - Як рівняння Ренкіна (Rankine) або Ренкіна-Гюгоніо.

Зазначимо, що у рівнянні Гюгоніо (2.75-76) функції h_1 (або e_1) і h_2 (або e_2) передбачаються відомими функціями термодинамічних параметрів p_1, v_1 та p_2, v_2 відповідно (ця залежність визначається рівнянням стану газу).

У припущенні, що вид рівняння стану газу не змінюється при переході через поверхню прямого стрибка (а це дійсно має місце, якщо хімічний склад

газу не змінюється і перепад температур $T_2 - T_1$ не дуже великий), безліч всіх можливих термодинамічних станів, (p, v) які може перейти в результаті перетворення на стрибку газ з вихідним станом (p_1, v_1) , описується рівнянням

$$h(p, v) - h(p_1, v_1) = \frac{1}{2}(p - p_1)(v + v_1) \quad (2.77)$$

або

$$e(p, v) - e(p_1, v_1) = \frac{1}{2}(p + p_1)(v_1 - v) \quad (2.78)$$

Рівняння (2.22-23) задає площині (v, p) криву, яку у літературі прийнято називати адіабатою Гюгоніо або ударною адіабатою «див. рис. 2.2». Адіабата Гюгоніо проходить, природно, через точку 1 з координатами (v_1, p_1) , звану іноді центром адіабат Гюгоніо (або центром ударної адіабати).

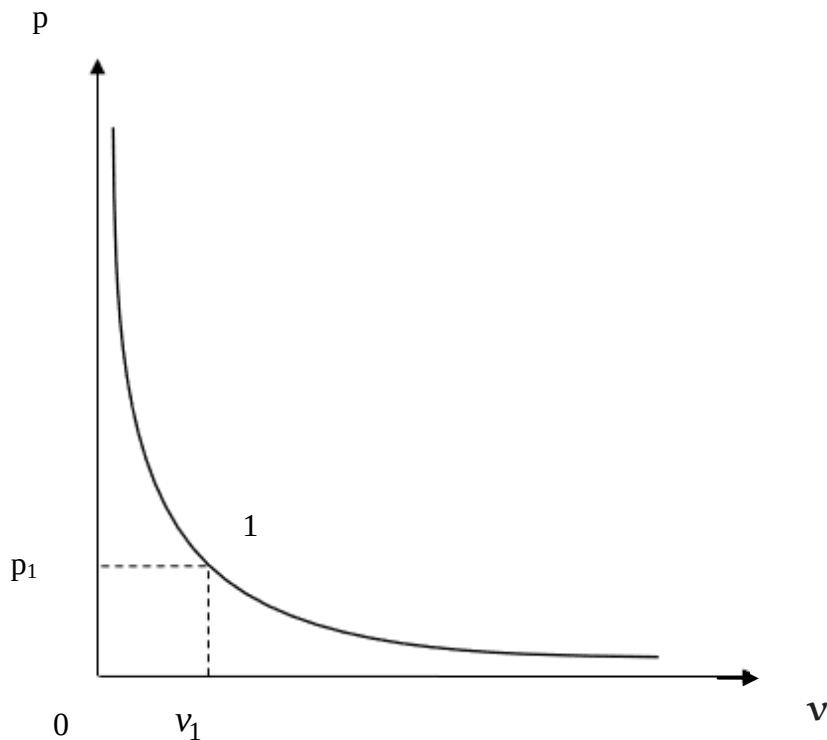


Рис. 2.2 – Загальний вигляд адіабати Гюгоніо

Слід зазначити, що хоча з погляду законів збереження, що включають і закон збереження енергії, що виражає перший початок термодинаміки, всі

стани газу, описані адіабатою Гюгоніо, в принципі, можливі, не всі вони можуть бути реалізовані з термодинамічних причин.

Справді, процес перетворення газу на стрибку є (згідно з нашим припущенням) адіабатичним, т.к. протікає без припливу тепла ззовні, але не є ізоентропічним (питома ентропія s_1 вихідного газу «1» не дорівнює питомій ентропії s_2 газу "2"). Необоротність процесу стрибкоподібного перетворення газу очевидна. Оскільки система є ізольованою (замкнутою), то, згідно з другим початком термодинаміки маємо

$$s_2 > s_1, \quad (2.79)$$

тому не всі стани кривої Гюгоніо можуть бути реалізовані.

Вдається цілком коректно довести, що як для слабких стрибків, коли

$$\frac{p_2 - p_1}{p_1} \ll 1,$$

так і для стрибків довільної інтенсивності, у природних припущеннях

$$\left(\frac{\partial p}{\partial v}\right)_s < 0 \quad \text{та} \quad \left(\frac{\partial^2 p}{\partial v^2}\right)_s > 0$$

(Цим умовам задовольняє і досконалий газ), стрибки розрідження (коли $p_2 < p_1$) неможливі, а можливі лише стрибки стиснення. Іншими словами, в цих випадках можуть бути реалізовані тільки ті термодинамічні стани, які характеризуються точками на адіабаті Гюгоніо, що лежать вище точки 1. У літературі цей висновок відомий як теорема Цемплена (Zemplen), хоча Жуге (Jouguet) і Цемплен незалежно один від одного, довели цю для політропного газу, а для довільної середовища доказ було проведено Л.Д. Ландау 1944 року.

Умови зростання ентропії, крім нерівності

$$p_2 > p_1 \quad (2.80)$$

з необхідністю призводить також до умов

$$u_1 > a_1 \quad (2.81)$$

та

$$u_2 < a_2, \quad (2.82)$$

де $a_k (k=1,2)$ – швидкість звуку у газі.

Прямий стрибок, що задовольняє нерівності (2.81,82) називають ударною хвилею стиснення або просто ударною хвилею.

Зауважимо, що до нерівностей (2.81,82) з необхідністю наводить також міркування Л.Д.Ландау про еволюційність ударних хвиль.

Розглянемо тепер прямий стрибок в ідеальному (досконалому) газі з постійними теплоємностями c_p, c_v , тобто. у законі збереження енергії с(2.3) для $k=1,2$ покладемо

$$h_k = \frac{\gamma}{\gamma+1} \frac{p_k}{\rho_k} = \frac{\gamma}{\gamma+1} p_k v_k, \quad (2.83)$$

$$e_k = \frac{1}{\gamma+1} \frac{p_k}{\rho_k} = \frac{1}{\gamma+1} p_k v_k, \quad (2.84)$$

Слід зазначити, що взагалі кажучи, (2.83,84) записані з точністю до довільної постійної адитивної.

У рівняннях (2.84) γ є відносини теплоємностей, тобто.

$$\gamma = \frac{c_p}{c_v} = \frac{c_{p_1}}{c_{v_1}} = \frac{c_{p_2}}{c_{v_2}} \quad (2.85)$$

Взагалі, рівняння стану ідеального газу має вигляд

$$pv=RT \quad (2.86)$$

або

$$p=\rho RT, \quad (2.86.a)$$

де R - Питома газова стала. Питома постійна газова R є величиною, постійною для кожного газу з незмінною хімічною формулою або для суміші (незмінного складу) таких газів – в останньому випадку прийнято говорити про питому газової постійної суміші. При цьому $R = R_\mu / \mu$, де $R_\mu = 8,31 \text{ Дж}/(\text{моль} \cdot \text{К})$ – універсальна постійна газова, однакова для всіх газів, μ

– молярна маса ($\mu=1\text{кг/моль}$). Для політропного газу, так і в більш загальному випадку досконалого газу, коли, як і питома внутрішня енергія та ентальпія, питомі теплоємності є функціями тільки абсолютної температури, вони задовольняють рівнянню Майєра (Mayer)

$$c_p - c_v = R \quad (2.87)$$

Оскільки удельна газова стала R завжди є позитивною величиною, рівняння Майєра (2.87) підкреслює той факт, що $c_p > c_v$ и $\gamma > 1$ (див. нижче). Підставляючи вирази (2.83) в рівняння Гюгоніо (2.84) (або, що те саме, підставляючи (2.84) (2.76)), після серії нескладних перетворень маємо

$$\frac{p_2}{p_1} = \frac{(\gamma + 1)v_1 - (\gamma - 1)v_2}{(\gamma + 1)v_2 - (\gamma - 1)v_1} \quad (2.88)$$

або

$$\frac{v_2}{v_1} = \frac{(\gamma + 1)p_1 + (\gamma - 1)p_2}{(\gamma - 1)p_1 + (\gamma + 1)p_2}, \quad (2.89)$$

Отже, безліч всіх можливих термодинамічних станів (p, v) , які можуть перейти в результаті перетворення на стрибку політропний газ з вихідним станом (p_1, v_1) , описується рівнянням

$$\frac{p}{p_1} = \frac{(\gamma + 1)v_1 - (\gamma - 1)v}{(\gamma + 1)v - (\gamma - 1)v_1} \quad (2.90)$$

або

$$\frac{p}{p_1} = \frac{1 - \frac{\gamma - 1}{\gamma + 1} \frac{v}{v_1}}{\frac{v}{v_1} - \frac{\gamma - 1}{\gamma + 1}} \quad (2.90.a)$$

Графік ударної адіабати (2.90) у безрозмірних координатах p/p_1 та v/v_1 зображено на рис. 2.3:

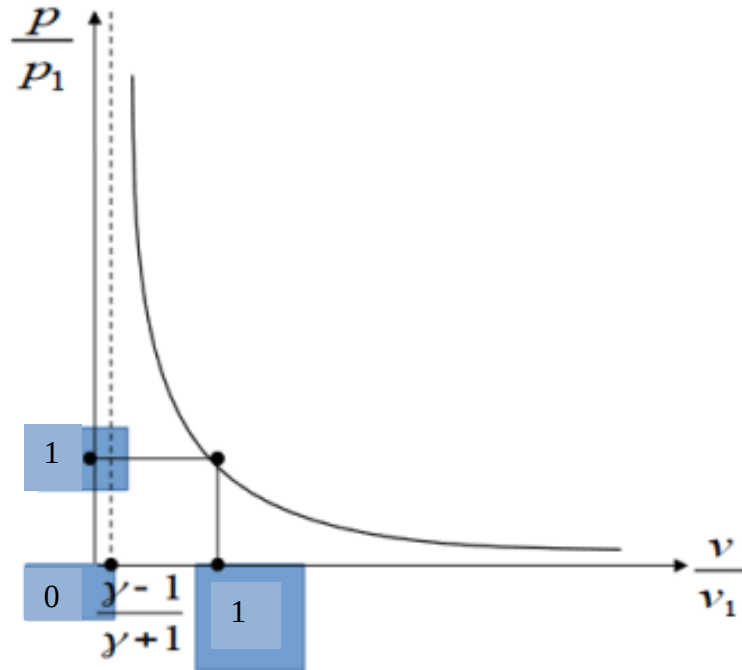


Рис. 2.3 – Ударна адіабата для досконалого газу

Адiabата (2.90.а) має вертикальну асимптоту

$$\frac{v}{v_1} = \frac{\gamma - 1}{\gamma + 1},$$

тобто,

$$p/p_1 \rightarrow \infty \quad \text{за умов} \quad \frac{v}{v_1} \rightarrow \frac{\gamma - 1}{\gamma + 1},$$

Інакше,

$$\frac{\rho}{\rho_1} \rightarrow \frac{\gamma + 1}{\gamma - 1} \quad \text{за умов} \quad \frac{p}{p_1} \rightarrow \infty$$

Іншими словами, досконалий газ з постійними теплоємностями при одноразовому ударному стисканні може бути стиснутий не більше ніж у

$$\frac{\gamma + 1}{\gamma - 1} \text{ разів.}$$

Відомо, що величина γ визначається як

$$\gamma = \frac{f + 2}{f}, \quad (2.91)$$

де f – число ступенів свободи молекули газу.

Для одноатомного газу $f = 3$ і, відповідно,

$$\gamma = 5/3 \quad \text{та} \quad \frac{\gamma + 1}{\gamma - 1} = 4 \quad (2.92)$$

Для двоатомного газу, а також для триатомного газу з лінійною молекулою $f=5$ і, відповідно

$$\gamma = 7/5 = 1,4 \quad \text{та} \quad \frac{\gamma + 1}{\gamma - 1} = 6 \quad (2.93)$$

Причому, випадок двоатомного газу є, з практичної погляду найважливішим, т.к. під цей випадок підпадають кисень (O_2), водень (H_2), азот (N_2), повітря.

Для триатомного газу із трикутною молекулою $f = 6$ та, відповідно,

$$\gamma = 4/3 \quad \text{та} \quad \frac{\gamma + 1}{\gamma - 1} = 7 \quad (2.94)$$

Чим складніший вигляд має молекула газу, тим, як правило, менша величина γ і більша величина максимально можливого ударного стиску газу. При цьому, у будь-якому випадку

$$\gamma > 1 \quad (c_p > c_v) \quad (2.95)$$

та

$$\frac{\gamma + 1}{\gamma - 1} \rightarrow \infty \quad \text{при} \quad f \rightarrow \infty$$

Отже, на прямому стрибку у досконалому газі з постійними теплоємностями (політропному газі) для вихідного газу «1» і ударно-стиснутого газу «2» справедливе співвідношення (2.96)

$$\frac{p_2}{p_1} = \frac{(\gamma + 1)v_1 - (\gamma - 1)v_2}{(\gamma + 1)v_2 - (\gamma - 1)v_1},$$

яке можна записати також у вигляді

$$\frac{p_2}{p_1} = \frac{(\gamma + 1) - (\gamma - 1) \frac{v_2}{v_1}}{(\gamma + 1) \frac{v_2}{v_1} - (\gamma - 1)} . \quad (2.96)$$

З (2.96) або (2.89) безпосередньо випливає

$$\frac{v_2}{v_1} = \frac{(\gamma + 1) + (\gamma - 1) \frac{p_2}{p_1}}{(\gamma - 1) + (\gamma + 1) \frac{p_2}{p_1}} \quad (2.97)$$

Для сильних ударних хвиль, коли $p_2 \gg p_1$ ($p_2 / p_1 \gg 1$, $\frac{p_2 - p_1}{p_1} \gg 1$),
приблизно можна вважати

$$\frac{v_2}{v_1} = \frac{\gamma - 1}{\gamma + 1} \quad (2.98)$$

та

$$\frac{\rho_2}{\rho_1} = \frac{\gamma + 1}{\gamma - 1} \quad (2.99)$$

З урахуванням рівняння стану досконалого газу маємо також

$$\frac{T_2}{T_1} = \frac{p_2}{p_1} \frac{(\gamma + 1) - (\gamma - 1) \frac{p_2}{p_1}}{(\gamma - 1) + (\gamma + 1) \frac{p_2}{p_1}} \quad (2.100)$$

Очевидно, що при $p_2/p_1 \rightarrow \infty$ справедливо $T_2/T_1 \rightarrow \infty$.

У прикладних завданнях корисні також формули, що виражають відносини тисків, щільностей і температур у ударній хвилі через число Маха (Mach) у газі «1»

$$M_1 = \frac{u_1}{a_1} \quad (2.101)$$

Ці формули мають вигляд

$$\frac{p_2}{p_1} = \frac{2\gamma M_1^2}{(\gamma+1)} - \frac{\gamma-1}{\gamma+1}, \quad (2.102)$$

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma+1)M_1^2}{(\gamma-1)M_1^2 + 2}, \quad (2.103)$$

$$\frac{v_2}{v_1} = \frac{(\gamma-1)M_1^2 + 2}{(\gamma+1)M_1^2}, \quad (2.104)$$

$$\frac{T_2}{T_1} = \frac{[2\gamma M_1^2 - (\gamma-1)][(\gamma-1)M_1^2 + 2]}{(\gamma+1)^2 M_1^2}. \quad (2.105)$$

Причому, $u_1 > a_1$, то

$$M_1 > 1 \quad (2.106)$$

При цьому

$$M_2 = u_2/a_2 < 1 \quad (2.107)$$

Для сильних ударних хвиль ($p_2 \gg p_1$) має місце $M_1 \gg 1$.

Зазначимо також, що у випадку досконалого газу при $\kappa=1,2$

$$a_k^2 = \gamma p_k / \rho_k = \gamma p_k v_k \quad (2.108)$$

або, що те саме

$$a_k = \sqrt{\frac{\gamma p_k}{\rho_k}} = \sqrt{\gamma p_k v_k} \quad (2.109)$$

Як із загальних позицій газової динаміки, і з погляду вибухо- і пожежобезпеки, і навіть з погляду необхідності оцінки можливих наслідків вибуху, особливу роль грає вирішення завдання розрахунку термо- і газодинамічних параметрів за ударним фронтом, тобто. у сфері «2», при заданому стані газу перед ударним фронтом, тобто. у сфері «1».

Іншими словами, це завдання вимагає розрахунку тиску p_2 , щільності ρ_2 , абсолютної температури T_2 і т.п. за заданими значеннями p_1 і ρ_1 (рівняння стану газу по обидва боки поверхні ударного стрибка передбачається відомим, температура T_1 та інші параметри газу «1» відомі або визначаються за рівнянням стану).

Очевидно, що таке завдання має безліч рішень, які для ударних хвиль стиснення відповідають точкам верхньої частини ударної адіабати (усі точки, розташовані на ударній адіабаті вище точки 1 на «див. рис. 2.2 та 2.3». Математично це пояснюється тим, що система рівнянь Ренкіна-Гюгоніо (2.1–3) є системою 3-х рівнянь алгебри щодо 4-х невідомих p_2 , ρ_2 , u_2 та u_1 . Щоб замкнути цю систему рівнянь, потрібно зменшити кількість шуканих невідомих, задавши або швидкість розповсюдження ударної хвилі u_1 , або її інтенсивність p_2/p_1 , – лише тоді завдання має єдине рішення. У практичних завданнях інтенсивність ударної хвилі відома частіше, ніж її поширення.

При заданій швидкості розповсюдження ударної хвилі u_1 , алгоритм розрахунку параметрів газу за провідним ударним фронтом такий:

1. З рівняння стану вихідного газу «1» визначається швидкість звуку a_1 у цьому газі; у разі досконалого газу з постійними теплоємностями для цього застосовується рівняння (2.109). Потім за заданою швидкістю поширення ударної хвилі u_1 розраховується квадрат числа Маха M_1^2 згідно (2.101). Необхідне виконання нерівності (2.106), в іншому випадку аналізований перебіг не є ударною хвилею стиснення.

2. Визначається швидкість u_2 потоку за ударною хвилею, навіщо у разі скоєного газу використовується формула

$$\frac{u_2}{u_1} \equiv \frac{\rho_1}{\rho_2} = \frac{(\gamma - 1)M_1^2 + 2}{(\gamma + 1)M_1^2}, \quad (2.110)$$

точніше,

$$u_2 = u_1 \frac{(\gamma - 1)M_1^2 + 2}{(\gamma + 1)M_1^2}, \quad (2.110.a)$$

3. За формулами (2.102), (2.103) та (2.104) відповідно розраховуються послідовно тиск p_2 , щільність ρ_2 та абсолютна температура T_2 .

При заданій інтенсивності p_2/p_1 ударної хвилі, алгоритм розрахунку параметрів газу за провідним ударним фронтом такий:

1. За формулою

$$M_1^2 = \frac{\gamma + 1}{2\gamma} \frac{p_2}{p_1} - \frac{\gamma - 1}{2\gamma}, \quad (2.111)$$

яка є прямим наслідком (2.102), розраховується квадрат числа Маха M_1^2 .

2. Визначається спочатку квадрат швидкості ударної хвилі u_1^2 за формулою

$$u_1^2 = M_1^2 a_1^2, \quad (2.112)$$

а потім і сама швидкість ударної хвилі u_1 .

3. За формулами (2.113) та (2.113) розраховуються відповідно щільність ρ_2 та абсолютна температура T_2 . Тиск p_2 фактично відомий спочатку.

Зазначимо, що замість інтенсивності p_2/p_1 ударної хвилі може бути заданий надлишковий тиск - абсолютний ($p_2 - p_1$) або відносне $((p_2 - p_1) / p_1)$.

2.3 Основні закономірності прямого скачку з притоком енергії та співвідношенням між різними режимами горіння

Розглянемо ту ж модель прямого стрибка, що в розділі 2.2, з тією різницею, що на поверхні стрибка допускаються хімічні перетворення з тепловиділенням (або теплопоглинанням).

$$h_1 + q + \frac{u_1^2}{2} = h_2 + \frac{u_2^2}{2} \quad (2.113)$$

або

$$e_1 + q + \frac{p_1}{\rho_1} + \frac{u_1^2}{2} = e_2 + \frac{p_2}{\rho_2} + \frac{u_2^2}{2} \quad (2.113.a)$$

Де, q – питома теплота хімічної реакції (питоме тепловиділення).

При цьому можна вважати, що для екзотермічної реакції $q > 0$, а для ендотермічної реакції $q < 0$. Величину q вважатимемо постійною (для цього виду хімічних перетворень).

Рівняння (2.74) і (2.76) у разі набувають вигляду

$$h_2 - h_1 - q = \frac{1}{2}(p_2 - p_1)(v_1 - v_2) \quad (2.114)$$

та

$$e_2 - e_1 - q = \frac{1}{2}(p_2 + p_1)(v_1 - v_2) \quad (2.115)$$

відповідно.

В даному випадку вже не можна, взагалі кажучи, вважати, що вид рівняння стану газу не змінюється при переході через поверхню прямого стрибка, тому що на стрибку мають місце хімічні перетворення. Тому, безліч усіх можливих термодинамічних станів (p, v) , які може перейти в результаті перетворення на стрибку газ з вихідним станом (p_1, v_1) , описується рівнянням:

$$h_2(p, v) - h_1(p_1, v_1) = \frac{1}{2}(p - p_1)(v + v_1) + q \quad (2.116)$$

або

$$e_2(p, v) - e_1(p_1, v_1) = \frac{1}{2}(p + p_1)(v_1 - v) + q \quad (2.117)$$

де h_1, h_2 або e_1, e_2 відомі, але, взагалі кажучи, різні функції від тиску та питомого обсягу.

Рівняння (2.116, 117) задає площині (v, p) криву, яку в літературі прийнято називати детонаційною або ударно-детонаційною адіабатою «див. Рис. 2.3» Ця крива, природно, при $q \neq 0$ вже не проходить через точку 1 з координатами (v_1, p_1) . У разі $q > 0$ (саме цей випадок відповідає детонації та горінню) адіабату лежить вище за точку 1.

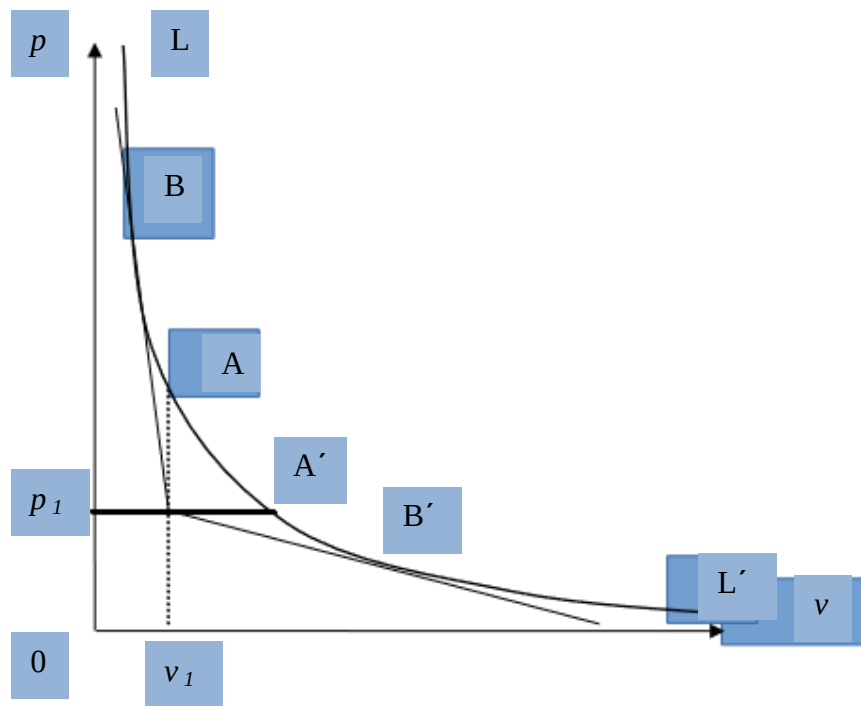


Рис. 2.4 – Загальний вигляд детонаційної адіабати

Різні ділянки кривої «див. рис. 2.4» відповідають різним режимам горіння. Ділянка кривої AA' не має фізичного сенсу, тому що відповідає уявній величині потоку маси через стрибок ($j^2 < 0$). З цієї причини ударно-детонаційна адіабата, яку іноді також називають кривою Гюгонію або адіабатою Гюгонію для детонації, хоча історично це і не зовсім вірно, розпадається на дві окремі гілки:

- 1) детонаційну ABL ;
- 2) дефлаграційну $A'B'L'$.

Точка B є точка торкання верхньої (детонаційної) гілки адіабати та прямої, проведеної з точки 1. У цій точці має місце співвідношення

$$u_2 = a_2 \quad (2.118)$$

та

$$M_2 = 1 \quad (2.119)$$

тобто. детонаційна хвиля, що відповідає точці B , рухається щодо газу, що залишається безпосередньо за нею (продуктів детонації) зі швидкістю, що дорівнює швидкості звуку. Чепмен (Chapman) і Жуге (Jouguet), а також

Михельсон, незалежно один від одного припустили, що саме точка **B** на детонаційній адіабаті відповідає тому єдиному можливому режиму детонації, що самопідтримується, який спостерігається в експериментах. Пізніше (наприкінці 1930-х ХХ ст.) це припущення було доведено суворо Я.Б.Зельдовичем. Точка **B** називається точкою Чепмена-Жуге (ЧЖ).

Ділянка BL детонаційної гілки адіабати відповідає нерівності

$$u_2 < a_2 \quad (2.120)$$

або

$$M_2 < 1 \quad (2.121)$$

При цьому, вздовж ділянки BL (як, втім, і вздовж усієї детонаційної гілки ABL кривою Гюгоніо), залишається в силі нерівність

$$u_1 > a_1 \quad (2.122)$$

або, що те саме

$$M_1 > 1, \quad (2.123)$$

оскільки хвилі детонації завжди поширюються із надзвуковою швидкістю.

Стану на кривій вище точки **B**, тобто на ділянці BL , відповідають так звані сильні чи перестиснуті детонації. Перестиснута детонація, якщо не підштовхувати хвилю поршнем або направляючи в трубку, що звужується, не може бути стаціонарною і переходить у стійку нормальну детонацію з мінімальною швидкістю, що визначається точкою торкання **B**, тобто. у детонацію Чепмена-Жуге.

Ділянка AB детонаційної гілки адіабати Гюгоніо відповідає нерівності

$$u_2 > a_2 \quad (2.124)$$

або

$$M_2 > 1 \quad (2.125)$$

Це ділянка про слабких чи недостигнутих детонацій. Слабу детонацію можна отримати лише штучно, відмовившись від механізму займання газу ударною хвилею. Як наочний приклад слабкої детонації К.І.Щелкін наводить одночасне займання газу по всій довжині труби (флеш-процес і т.п.). Підпалюючи суміш безліччю іскор, що включаються в потрібній послідовності, або спалахуючи її променями світла, що швидко переміщаються, можна здійснити будь-який стан на ділянці AB , і, відповідно, будь-яку швидкість слабкої детонації. У цьому випадку швидкість детонації визначається зовнішнім пристроєм, що переміщує пучок світла або створює іскри.

Якщо вздовж усієї детонаційної ділянки ABL виконуються нерівності (2.122,123), то вздовж усієї дефлаграційної ділянки $A'B'L'$ виконується нерівність:

$$u_1 < a_1 \quad (2.126)$$

або, що те саме,

$$M_1 < 1 \quad (2.127)$$

тобто, дефлаграційна хвиля (хвиля горіння) завжди поширюється із дозвуковою швидкістю.

Слово "дефлаграція", взагалі кажучи, буквально перекладатися як "швидке горіння", "вибух", "спалах". До цього часу це поняття чітко не визначено. Іноді під дефлаграцією розуміють усі види поширення горіння з дозвуковою (на противагу детонації) швидкістю.

Основи систематики горіння створені Жуге, що скористалися для цієї мети ударно-детонаційною адіабатою «див. Рис. 2.3» . Жуге, а потім Крюсар (Crussard) визначають дефлаграцію як горіння, що відповідає ділянці $A'B'$ кривою Гюгонію, при цьому крапка B' є точка дотику нижньої гілки адіабати з прямою, проведеною з точки 1.

Очевидно, що при горінні, на відміну від детонації, відбувається зниження тиску і розширення продуктів реакції. Це властивість іноді

вважається головною ознакою дефлаграції. дефлаграція”, хоча при зміні швидкості полум'я стан продуктів горіння не зображується точкою, що переміщається по нижній гілці ударно-детонаційної адіабати. Нижня гілка відповідає тільки горінню з тією чи іншою постійною швидкістю відповідної точки адіабати Гюгонію.

Курант (Courant) і Фрідріхс (Friedrichs), наприклад, відносять до дефлаграції якісно розглянуте ними поширення повільного горіння з ударною хвилею, що біжить перед його фронтом, інтенсивність якої, при постійній швидкості горіння, залежить від граничних умов у продуктах згоряння. в більшості випадків нестационарно. Завдяки відмінності в швидкостях фронт ударної хвилі віддаляється від фронту горіння, і таке явище в цілому істотно відрізняється від дефлаграції. Але якщо взяти за вихідне значення стан незбуреного газу, стан продуктів згоряння не описується адіабатою Гюгонію.

Тобто, у класичному сенсі під дефлаграцією розуміються режими поширення полум'я з постійною швидкістю, що описуються нижньою гілкою $A'B'L'$ ударно-детонаційної адіабати.

Це визначення дефлаграції набагато ширше визначення тих авторів, які відносять до дефлаграції поширення полум'я лише з нормальною швидкістю, яка визначається теплопровідністю газу. Але з іншого боку, таке визначення, здавалося б, дуже збіднює зміст поняття дефлаграції, т.к. рівномірного поширення полум'я в трубах та обсягах іншої форми практично ніколи не спостерігається. Горіння, що супроводжується значним виділенням тепла, спричиняє інтенсивні рухи незгорілого газу, рух газу обурює фронт полум'я, виникають нестационарні режими горіння.

Однак, наведене вище визначення дефлаграції, дозволяє віднести до неї з тими чи іншими наближеннями більшість видів технічного горіння в топках, у тому числі і в камерах згоряння ракетних і реактивних двигунів.

Ділянка $A'B'$ відповідає так званим слабким дефлаграціям, для яких справедлива нерівність

$$u_2 < a_2 \quad (2.128)$$

або, що те саме,

$$M_2 < 1 \quad (2.129)$$

Ділянка $B'L'$ відповідає сильним дефлаграціям, для яких

$$u_2 > a_2 \quad (2.130)$$

або

$$M_2 > 1 \quad (2.131)$$

Точки B' , де виконані співвідношення (2.118,119) називають точкою дефлаграції Чепмена-Жуге, хоча з історичної точки зору така назва не цілком виправдана.

У разі дефлаграції в принципі здійсненні будь-які стани на ділянці $A'B'$. Умови відбору швидкості, подібної до умови Жуге для детонації, в даному випадку немає. Здійсненність термодинамічного стану, що відповідає будь-якій точці на ділянці слабких дефлаграцій $A'B'$, пов'язана з незалежністю механізму поширення повільного горіння від параметрів стрибка розрідження, що становить дефлаграцію. Швидкість поширення полум'я залежить від фізико-хімічних властивостей суміші (нормальна швидкість), турбулентності (якщо вона має місце) та розподілу швидкостей незгорілого газу.

Сильні дефлаграції (всі точки на ділянці $B'L'$, тобто. нижче точки B') нездійсненні, хоча в принципі можливі термодинамічно можливі. Експериментально поширення сильних дефлаграційних хвиль немає. Проведені теоретичні дослідження хімічного, лазерного та ядерного горіння також не вказують на можливість існування таких хвиль. Не запропоновані і будь-які інші моделі реальних фізичних процесів тепловиділення і теплопередачі, які призводили б до хвиль, що самопоширюються, сильною дефлаграції.

Л.Д.Ландау введено поняття про еволюційність ударних хвиль як про необхідну умову можливості їх реалізації. Якщо розглядати з цього погляду

детонаційні та дефлаграційні хвилі, то очевидна нееволюційність слабких (недостигнутих) детонацій та сильних дефлаграцій.

Таким чином, з усіх режимів горіння і детонації, що описуються ударно-детонаційною адіабатою (кривою Гюгоніо), як стаціонарні режими, що самопідтримуються, можуть реалізовуватися тільки режим самопідтримуваної детонації Чепмена-Жуге ($u_1 > a_1$, $u_2 = a_2$), відповідний точці У кривій, і режим слабкої дефлаграції ($u_1 < a_1$, $u_2 < a_2$), відповідній ділянці $A'B'$ кривою, причому переважній більшості випадків нормального горіння відповідають точки, розташовані на ділянці $A'B'$ поблизу A' .

Розглянемо тепер прямий стрибок з тепловиділенням в ідеальному (досконалому) газі з постійними теплоємностями. c_p , c_v , тобто, у законі збереження енергії та в рівняннях ударно-детонаційної адіабати (2.61,62) покладемо

$$h_k = \frac{\gamma_k}{\gamma_k + 1} \frac{p_k}{\rho_k} = \frac{\gamma_k}{\gamma_k + 1} p_k v_k$$

або

$$e_k = \frac{1}{\gamma_k + 1} \frac{p_k}{\rho_k} = \frac{1}{\gamma_k + 1} p_k v_k \quad \text{для } k = 1, 2$$

(Див. співвідношення (2.83,84))

Маємо наступну систему 3-х рівнянь алгебри

$$\begin{cases} \rho_1 u_1 = \rho_2 u_2 \\ p_1 + \rho_1 u_1^2 = p_2 + \rho_2 u_2^2 \\ \frac{\gamma_1}{\gamma_1 - 1} \frac{p_1}{\rho_1} + \frac{u_1^2}{2} + q = \frac{\gamma_2}{\gamma_2 - 1} \frac{p_2}{\rho_2} + \frac{u_2^2}{2}, \end{cases} \quad (2.132)$$

З системи (2.132) впливає співвідношення

$$(\gamma_2 + 1)u_2^2 - 2\gamma_2\left(\frac{1}{\gamma_1 M_1^2} + 1\right)u_1 u_2 + \left[\frac{2(\gamma_2 - 1)}{(\gamma_1 - 1)M_1^2} + (\gamma_2 - 1)\right]u_1^2 + 2(\gamma_2 - 1)q = 0 \quad (2.133)$$

У випадку, коли $\gamma_1 = \gamma_2 = \gamma$ та $q=0$, тобто. за відсутності тепловиділення та хімічних перетворень на стрибку, рівняння (2.133) набуває вигляду

$$(\gamma + 1)u_2^2 - 2\gamma\left(\frac{1}{\gamma M_1^2} + 1\right)u_1 u_2 + \left[\frac{2}{M_1^2} + (\gamma - 1)\right]u_1^2 = 0 \quad (2.134)$$

Останнє рівняння добре відоме з теорії ударних хвиль.

Співвідношення (2.78) для хвилі самопідтримуваної детонації Чепмена-Жуге можна розглядати як квадратне рівняння щодо швидкості поширення детонації u_1 , тому що в цьому випадку для швидкості u_2 виконано умову Жуге $u_2 = a_2$. Для перетиснутих детонацій, коли швидкість поширення детонаційної хвилі u_1 визначається деякими зовнішніми по відношенню до самої хвилі факторами і може вважатися заданою, співвідношення (2.133) можна розглядати як рівняння щодо швидкості u_2 . Для слабких дефлаграцій (повільного горіння) співвідношення (2.133) можна знов-таки розглядати як рівняння щодо u_2 , оскільки швидкість u_1 в даному випадку є швидкість нормального горіння, яка визначається як характеристика середовища (палива газової суміші «1») і вважається заданою.

Як із загальних позицій теорії детонації, так і з погляду вибухо- та пожежної безпеки, а також з точки зору необхідності оцінки можливих наслідків вибуху, особливу роль відіграє вирішення задачі розрахунку термо- і газодинамічних властивостей за ударно-детонаційним фронтом, тобто. у сфері продуктів детонації «2», при заданому стані газу перед ударно-детонаційним фронтом, тобто. в ділянці вихідної горючої газової суміші «1». Іншими словами, це завдання вимагає розрахунку тиску p_2 , щільності ρ_2 , абсолютної температури T_2 і т.п. за заданими значеннями p_1 і ρ_1 (рівняння стану газу по обидва боки поверхні ударного стрибка передбачається

відомим, температура T_1 та інші термодинамічні параметри газу «1» відомі або визначаються за рівнянням стану).

Очевидно, що таке завдання для ударних хвиль стиснення має безліч рішень. Ці рішення відповідають точкам верхньої частини ударної адіабати (всі точки, розташовані на ударній адіабаті вище за точку 1 на « див.рис. 2.2 і 2.3». Математично цей факт пояснюється тим, що система рівнянь Ренкіна-Гюгоніо (2.7–9) є системою 3-х рівнянь алгебри щодо 4-х невідомих p_2 , ρ_2 , u_2 и u_1 , тобто. не є замкнутою системою. З цієї системи ні швидкість розповсюдження ударної хвилі u_1 , ні параметри середовища за ударним фронтом (p_2 , ρ_2 , T_2) не можуть бути визначені однозначно.

Однак для самопідтримуваної детонації виконання умови Жуге (2.118,119) дозволяє (при заданому питомому тепловиділенні q і відомих рівняннях стану вихідного пального середовища «1» і продуктів детонації «2») замкнути систему рівнянь (2.113), (2.114) і (2.115) (Швидкість поширення ударно-детонаційного фронту) u_1 , так і параметри продуктів детонації. Побудуємо алгоритм такого розрахунку для найцікавішого з практичної точки зору випадку, коли середовище по обидві сторони ударно-детонаційного фронту може бути представлене моделлю досконалого газу з постійними теплоємностями (політропного газу).

Прямим наслідком умови Жуге (2.118,119) є співвідношення $u_2^2 = a_2^2$, яке для досконалого газу «2» з постійними теплоємностями набуває вигляду

$$u_2^2 = \frac{\gamma_2 p_2}{\rho_2} \quad (2.135)$$

Розглянемо рішення системи (2.132) як системи нелінійних рівнянь алгебри щодо 4-х невідомих p_2 , ρ_2 , u_2 та u_1 під час виконання умови (2.134).
Маємо

$$\left\{ \begin{array}{l} \rho_2 u_2 = \rho_1 u_1 \\ p_2 + \rho_2 u_2^2 = p_1 + \rho_1 u_1^2 \\ \frac{\gamma_2}{\gamma_2 - 1} \frac{p_2}{\rho_2} + \frac{u_2^2}{2} = \frac{\gamma_1}{\gamma_1 - 1} \frac{p_1}{\rho_1} + \frac{u_1^2}{2} + q \\ u_2^2 = \frac{\gamma_2 p_2}{\rho_2} \end{array} \right. , \quad (2.136)$$

або після зведення в квадрат першого рівняння системи (2.136),

$$\left\{ \begin{array}{l} \rho_2^2 u_2^2 = \rho_1^2 u_1^2 \\ p_2 + \rho_2 u_2^2 = p_1 + \rho_1 u_1^2 \\ \frac{\gamma_2}{\gamma_2 - 1} \frac{p_2}{\rho_2} + \frac{u_2^2}{2} = \frac{\gamma_1}{\gamma_1 - 1} \frac{p_1}{\rho_1} + \frac{u_1^2}{2} + q \\ u_2^2 = \frac{\gamma_2 p_2}{\rho_2} \end{array} \right. \quad (2.136.a)$$

Виключаючи величину u_2 (точніше, u_2^2) з перших трьох рівнянь системи (2.136.a), маємо

$$\left\{ \begin{array}{l} \gamma_2 \rho_2 p_2 = \rho_1^2 u_1^2 \\ (\gamma_2 + 1) p_2 = p_1 + \rho_1 u_1^2 \\ \frac{\gamma_2(\gamma_2 + 1)}{2(\gamma_2 - 1)} \frac{p_2}{\rho_2} = \frac{\gamma_1}{\gamma_1 - 1} \frac{p_1}{\rho_1} + \frac{u_1^2}{2} + q \\ u_2^2 = \frac{\gamma_2 p_2}{\rho_2} \end{array} \right. \quad (2.137)$$

Розглянемо систему із перших трьох рівнянь системи (2.137).

$$\left\{ \begin{array}{l} \gamma_2 \rho_2 p_2 = \rho_1^2 u_1^2 \\ (\gamma_2 + 1) p_2 = p_1 + \rho_1 u_1^2 \\ \frac{\gamma_2(\gamma_2 + 1)}{2(\gamma_2 - 1)} \frac{p_2}{\rho_2} = \frac{\gamma_1}{\gamma_1 - 1} \frac{p_1}{\rho_1} + \frac{u_1^2}{2} + q \end{array} \right. \quad (2.138)$$

Маємо

З першого рівняння системи (2.138) випливає, що

$$u_1^2 = \frac{\gamma_2 \rho_2 p_2}{\rho_1^2} \quad (2.139)$$

Співвідношення (2.139) дозволяє виключити з другого та третього рівнянь системи (2.138) величину u_1 (точніше, u_1^2). У результаті маємо таку систему двох рівнянь щодо двох невідомих p_2, ρ_2

$$\begin{cases} (\gamma_2 + 1)p_2 = p_1 + \frac{\gamma_2 \rho_2 p_2}{\rho_1} \\ \frac{\gamma_2(\gamma_2 + 1)}{2(\gamma_2 - 1)} \frac{p_2}{\rho_2} = \frac{\gamma_1}{\gamma_1 - 1} \frac{p_1}{\rho_1} + \frac{\gamma_2 \rho_2 p_2}{2\rho_1^2} + q \end{cases} \quad (2.140)$$

Розділивши перше рівняння системи (2.139) на величину p_1 ($p_1 \neq 0$) та

помноживши друге рівняння цієї системи на величину $\frac{\rho_2}{p_1}$ ($\frac{\rho_2}{p_1} \neq 0$), маємо

$$\begin{cases} (\gamma_2 + 1) \frac{p_2}{p_1} = 1 + \gamma_2 \frac{\rho_2}{\rho_1} \frac{p_2}{p_1} \\ \frac{\gamma_2(\gamma_2 + 1)}{2(\gamma_2 - 1)} \frac{p_2}{p_1} = \frac{\gamma_1}{\gamma_1 - 1} \frac{\rho_2}{\rho_1} + \frac{\gamma_2}{2} \left(\frac{\rho_2}{\rho_1} \right)^2 \frac{p_2}{p_1} + \frac{q \rho_1}{p_1} \frac{\rho_2}{\rho_1} \end{cases} \quad (2.141)$$

Слідуючи К.І. Щолкіну та Я.К. Трошину, введемо такі умовні позначення

$$r = \frac{\rho_2}{\rho_1}, \pi = \frac{p_2}{p_1} \quad (2.142)$$

Зазначимо, що позначення (2.142) є нестандартними і використовуються в такому сенсі лише в рамках цього додатку.

Введемо також позначення

$$Q = \frac{q\rho_1}{p_1} \quad (2.143)$$

З формули 2.108 випливає, що

$$Q = \frac{\gamma_1 q}{a_1^2} \quad (2.144)$$

Фізичний зміст безрозмірної величини Q цілком очевидний: дана величина характеризує відношення питомого тепловиділення в результаті екзотермічної хімічної реакції до питомої внутрішньої енергії паливної газової суміші.

Підстановка співвідношень (2.142) і (2.143) рівняння (2.141) призводить до наступних співвідношень:

$$\begin{cases} (\gamma_2 + 1)\pi = 1 + \gamma_2 r \pi \\ \frac{\gamma_2(\gamma_2 + 1)}{2(\gamma_2 - 1)}\pi = \frac{\gamma_1}{\gamma_1 - 1}r + \frac{\gamma_2}{2}r^2\pi + Qr \end{cases} \quad (2.145)$$

З першого рівняння системи (2.90) маємо

$$r = \frac{(\gamma_2 + 1)\pi - 1}{\gamma_2\pi} \quad (2.146)$$

Підстановка виразу (2.145) для величини r в друге рівняння системи (2.144) призводить до співвідношення

$$\frac{\gamma_2(\gamma_2 + 1)}{2(\gamma_2 - 1)}\pi = \frac{\gamma_1[(\gamma_2 + 1)\pi - 1]}{(\gamma_1 - 1)\gamma_2} + \frac{\gamma_2}{2} \frac{[(\gamma_2 + 1)\pi - 1]^2\pi}{(\gamma_2\pi)^2} + Q \frac{(\gamma_2 + 1)\pi - 1}{\gamma_2\pi} \quad (2.147)$$

Помноживши обидві частини рівняння (2.147) величину $2(\gamma_1 - 1)(\gamma_2 - 1)(\gamma_2\pi)^2$, після нескладних алгебраїчних перетворень маємо

$$\begin{aligned} \gamma_2^3(\gamma_2 + 1)(\gamma_1 - 1)\pi^3 &= \gamma_2(\gamma_2 - 1)(\gamma_1 - 1)\pi[(\gamma_2 + 1)\pi - 1]^2 + \\ &+ 2\gamma_2(\gamma_2 - 1)[\gamma_1 + Q(\gamma_1 - 1)]\pi[(\gamma_2 + 1)\pi - 1] \end{aligned} \quad (2.148)$$

Розділивши обидві частини рівняння (2.148) на величину π ($\pi \neq 0$), після перенесення всіх членів у ліву частину та приведення подібних маємо квадратне рівняння щодо π наступного виду

$$-\gamma_2(\gamma_2 + 1)(\gamma_1 - 1)\pi^2 + 2\gamma_2(\gamma_2 + 1)(\gamma_2 - 1)[(Q - 1)(\gamma_1 - 1) + \gamma_1]\pi + \gamma_2(\gamma_2 - 1)(\gamma_1 - 1) - 2\gamma_2(\gamma_2 - 1)[Q(\gamma_1 - 1) + \gamma_1] = 0 \quad (2.149)$$

Розділивши обидві частини квадратного рівняння (2.149) на старший коефіцієнт $-\gamma_2(\gamma_2 + 1)(\gamma_1 - 1)$, отримаємо наведене квадратне рівняння

$$\pi^2 - 2(\gamma_2 - 1)(Q + \frac{1}{\gamma_1 - 1})\pi + \frac{\gamma_2 - 1}{\gamma_2 + 1}[2(Q + \frac{\gamma_1}{\gamma_1 - 1}) - 1] = 0 \quad (2.150)$$

Розв'яжемо квадратне рівняння (2.149). Обчислимо величину $D/4$ (Четверту частину дискримінанта даного квадратного рівняння). Після низки алгебраїчних перетворень маємо

$$\frac{D}{4} = \frac{(\gamma_2 - 1)[(\gamma_2^2 - 1)(Q + \frac{1}{\gamma_1 - 1})^2 - 2(Q + \frac{1}{\gamma_1 - 1}) - 1]}{\gamma_2 + 1}, \quad (2.151)$$

або

$$\frac{D}{4} = \frac{\gamma_2 - 1}{\gamma_2 + 1}[(\gamma_2^2 - 1)(Q + \frac{1}{\gamma_1 - 1})^2 - 2(Q + \frac{1}{\gamma_1 - 1}) - 1] \quad (2.151.a)$$

Визначимо знак величини $D/4$. Очевидно, що

$$\text{sign}(\frac{D}{4}) = \text{sign}[(\gamma_2^2 - 1)(Q + \frac{1}{\gamma_1 - 1})^2 - 2(Q + \frac{1}{\gamma_1 - 1}) - 1], \quad (2.152)$$

так як $\frac{\gamma_2 - 1}{\gamma_2 + 1} > 0$ ($\gamma_2 > 1$).

Вираз

$$(\gamma_2^2 - 1)(Q + \frac{1}{\gamma_1 - 1})^2 - 2(Q + \frac{1}{\gamma_1 - 1}) - 1 \quad (2.153)$$

є квадратний тричлен щодо величини $Q + \frac{1}{\gamma_1 - 1}$.

Квадратне рівняння (2.150) має дійсне коріння тоді і тільки тоді, коли квадратний тричлен (2.153) невід'ємний.

Вводячи позначення

$$X = Q + \frac{1}{\gamma_1 - 1}, \quad (2.154)$$

визначимо коріння квадратного тричлена (2.97), який розглядається як

квадратний тричлен щодо $X \equiv Q + \frac{1}{\gamma_1 - 1}$. Маємо

$$X_{1,2} = \frac{1 \pm \gamma_2}{\gamma_2^2 - 1}, \quad (2.155)$$

або

$$X_1 = -\frac{1}{\gamma_2 + 1}, X_2 = \frac{1}{\gamma_2 - 1} \quad (2.156)$$

Очевидно, що квадратний тричлен (2.154) негативний тоді і лише тоді, коли

$$X_1 < X \equiv Q + \frac{1}{\gamma_1 - 1} < X_2, \quad (2.157)$$

тобто

$$-\frac{1}{\gamma_2 + 1} < X \equiv Q + \frac{1}{\gamma_1 - 1} < \frac{1}{\gamma_2 - 1} \quad (2.158)$$

Нерівність (2.158) рівносильна наступній нерівності

$$-\left(\frac{1}{\gamma_1 - 1} + \frac{1}{\gamma_2 + 1}\right) < Q < \frac{1}{\gamma_2 - 1} - \frac{1}{\gamma_1 - 1} \quad (2.159)$$

Ліва частина подвійної нерівності (2.159) виконується завжди, оскільки

величина $-\left(\frac{1}{\gamma_1 - 1} + \frac{1}{\gamma_2 + 1}\right)$ негативна, а величина Q для будь-якої екзотермічної хімічної реакції позитивна. Однак права частина нерівності

(2.104) для реальних випадків самопідтримується детонації Чепмена-Жуге ніколи не виконується, оскільки позитивна величина Q є досить великою, а

величина $\frac{1}{\gamma_2 - 1} - \frac{1}{\gamma_1 - 1}$, якщо вона і позитивна, є досить малою величиною

(крім того, у принципі не виключені випадки $\frac{1}{\gamma_2 - 1} - \frac{1}{\gamma_1 - 1} \cong 0$ при $\gamma_1 \cong \gamma_2$ та

навіть $\frac{1}{\gamma_2 - 1} - \frac{1}{\gamma_1 - 1} < 0$ при $\gamma_1 < \gamma_2$). Таким чином, нерівність (2.159) не виконується, квадратний тричлен (2.153) невід'ємний і, як наслідок, квадратне рівняння (2.150) має дійсне коріння.

Прийmemo позначення

$$A = \sqrt{\frac{D}{4}} = \sqrt{\frac{\gamma_2 - 1}{\gamma_2 + 1} \left[(\gamma_2^2 - 1) \left(Q + \frac{1}{\gamma_1 - 1} \right)^2 - 2 \left(Q + \frac{1}{\gamma_1 - 1} \right) - 1 \right]}, \quad (2.160)$$

причому очевидно, що $A \geq 0$.

В окремому випадку $\gamma_1 = \gamma_2$ та $Q = 0$ ($q = 0$), тобто за відсутності хімічних перетворень та тепловиділення на стрибку, $\frac{D}{4} = 0$ та $A = 0$, а рівняння (2.150) має два збігаються дійсні корені $\pi_{1,2} = 1$. Виконання рівності $\pi = 1$, означає, що $p_2/p_1 = 1$, тобто. $p_2 = p_1$. Останній факт означає відсутність стрибка (тривіальний випадок): $p_2 = p_1$, $\gamma_2 = \gamma_1$ та $u_2 = u_1$. Отриманий для цього окремого випадку результат підтверджує правильність вироблених вище алгебраїчних перетворень.

Загалом наведене квадратне рівняння (2.150) має коріння

$$\pi_{1,2} = (\gamma_2 - 1) \left(Q + \frac{1}{\gamma_1 - 1} \right) \pm A \quad (2.161)$$

З (2.161), таким чином, впливає, що відношення тисків p_2/p_1 може приймати одне з двох можливих значень

$$\frac{p_2}{p_1} = (\gamma_2 - 1)(Q + \frac{1}{\gamma_1 - 1}) \pm A \quad (2.162)$$

Величину A можна подати у вигляді

$$A = \sqrt{(\gamma_2 - 1)^2 (Q + \frac{1}{\gamma_1 - 1})^2 - \frac{2(\gamma_2 - 1)}{\gamma_2 + 1} (Q + \frac{\gamma_1}{\gamma_1 - 1}) - \frac{\gamma_2 - 1}{\gamma_2 + 1}} \quad (2.163)$$

Так як детонаційна хвиля є хвиля стиснення, то $p_2 > p_1$ і, отже,

$\frac{p_2}{p_1} > 1$ ($\pi > 1$). З цієї причини, з урахуванням виразу A (2.163), Цілком очевидно, що у правій частині співвідношення (2.162) слід вибрати знак «+». Таким чином

$$\frac{p_2}{p_1} = (\gamma_2 - 1)(Q + \frac{1}{\gamma_1 - 1}) + A \quad (2.164)$$

З урахуванням (2.143) рівність (2.164) набуває вигляду:

$$\frac{p_2}{p_1} = (\gamma_2 - 1)(\frac{\gamma_1 q}{a_1^2} + \frac{1}{\gamma_1 - 1}) + A, \quad (2.165)$$

причому

$$A = \sqrt{(\gamma_2 - 1)^2 (\frac{\gamma_1 q}{a_1^2} + \frac{1}{\gamma_1 - 1})^2 - \frac{2(\gamma_2 - 1)}{\gamma_2 + 1} (\frac{\gamma_1 q}{a_1^2} + \frac{\gamma_1}{\gamma_1 - 1}) - \frac{\gamma_2 - 1}{\gamma_2 + 1}} \quad (2.166)$$

Інакше,

$$p_2 = [(\gamma_2 - 1)(\frac{\gamma_1 q}{a_1^2} + \frac{1}{\gamma_1 - 1}) + A] p_1, \quad (2.167)$$

або

$$p_2 = [(\gamma_2 - 1)(\frac{\rho_1 q}{p_1} + \frac{1}{\gamma_1 - 1}) + A] p_1, \quad (2.168)$$

де

$$A = \sqrt{\frac{\gamma_2 - 1}{\gamma_2 + 1} [(\gamma_2^2 - 1)(\frac{\rho_1 q}{p_1} + \frac{1}{\gamma_1 - 1})^2 - 2(\frac{\rho_1 q}{p_1} + \frac{1}{\gamma_1 - 1}) - 1]} \quad (2.169)$$

Відповідно до співвідношень (2.132) та (2.136) маємо

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma_2 + 1)(p_2 / p_1) - 1}{\gamma_2(p_2 / p_1)}, \quad (2.170)$$

або ж

$$\rho_2 = \frac{[(\gamma_2 + 1)(p_2 / p_1) - 1]\rho_1}{\gamma_2(p_2 / p_1)} \quad (2.171)$$

Відповідно, оскільки $\frac{u_2}{u_1} = \frac{\rho_1}{\rho_2}$, маємо

$$\frac{u_2}{u_1} = \frac{\gamma_2(p_2 / p_1)}{(\gamma_2 + 1)(p_2 / p_1) - 1}, \quad (2.172)$$

або ж

$$u_2 = \frac{\gamma_2(p_2 / p_1)u_1}{(\gamma_2 + 1)(p_2 / p_1) - 1} \quad (2.173)$$

При цьому згідно (2.128) швидкість детонації u_1 виражається формулою

$$u_1 = \sqrt{\frac{\gamma_2 \rho_2 p_2}{\rho_1^2}} \quad (2.174)$$

При заданих значеннях тиску p_1 , ρ_1 вихідного пального середовища, при відомих величинах γ_1 та γ_2 , характеризують вихідне пальне середовище та продукти детонації відповідно, а також при відомому значенні питомого тепловиділення на ударно-детонаційному фронті q , алгоритм розрахунку швидкості поширення детонаційної хвилі, що самопідтримується. u_1 та параметрів продуктів детонації такий:

1. За формулою (2.169) визначається величина A .
2. Визначається тиск p_2 за формулою (2.168). За формулою (2.165) визначається відношення p_2/p_1 , яке можна назвати інтенсивністю детонаційної хвилі за аналогією з інтенсивністю ударної хвилі.
3. Визначається щільність ρ_2 за формулою (2.171). За формулою (2.170) визначається відношення густин ρ_2/ρ_1 .

4. Швидкість звуку a_2 у продуктах детонації визначається за формулою (2.109) при $\kappa=2$, а швидкість u_2 потоку за детонаційною хвилею (швидкість продуктів детонації щодо фронту) визначається із співвідношення Жуге (2.118).

5. Швидкість детонації u_1 розраховується за формулою (2.174). Число Маха визначається як M_1 , де швидкість звуку a_1 у вихідному пальному середовищі розраховується за формулою (2.109) при $\kappa=1$.

6. Абсолютна температура T_2 продуктів детонації може бути розрахована за формулою

$$T_2 = \frac{p_2}{R_2 \rho_2}, \quad (2.175)$$

де R_2 – питома газова постійна продуктів детонації.

З загальних позицій теорії горіння і з точки зору забезпечення вибухотехнічної безпеки, а також з точки зору оцінки можливості переходу горіння в детонацію (вибух), досить важливим є рішення задачі розрахунку термо- і газодинамічних параметрів за фронтом полум'я, тобто. у сфері продуктів згоряння «2», при заданому стані газу перед фронтом полум'я, тобто. в області вихідної горючої газової суміші «1», за так званих слабких дефлаграцій (зокрема, при нормальному ламінарному горінні). Іншими словами, це завдання вимагає розрахунку тиску p_2 , щільності ρ_2 , абсолютної температури T_2 і т.п. за заданими значеннями p_1 та ρ_1 (рівняння стану газу по обидва боки поверхні стрибка передбачається відомим, температура T_1 та інші термодинамічні параметри газу «1» відомі або визначаються рівнянням стану). Відомою передбачається і величина u_1 , яка є швидкість поширення полум'я по вихідному горючому середовищі, тобто. швидкість нормального горіння.

З метою розв'язання цього завдання розглянемо рівняння (2.123) як квадратне рівняння щодо швидкості u_2 (швидкість продуктів згоряння щодо фронту полум'я).

Обчисливши четверту частину дискримінанта цього квадратного рівняння $D/4$, після низки алгебраїчних перетворень маємо

$$\frac{D}{4} = \left\{ \frac{\gamma_2^2}{\gamma_1^2 M_1^4} + 2 \left[\frac{\gamma_2^2}{\gamma_1 M_1^2} - \frac{2(\gamma_2^2 - 1)}{(\gamma_1 - 1) M_1^2} \right] + 1 \right\} u_1^2 - 2(\gamma_2^2 - 1)q, \quad (2.176)$$

або

$$\frac{D}{4} = \left[\frac{\gamma_2^2}{\gamma_1^2 M_1^4} + \frac{4\gamma_1(1 - \gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1 - 1) M_1^2} + 1 \right] u_1^2 - 2(\gamma_2^2 - 1)q \quad (2.177)$$

Цілком очевидно, що у реальних випадках горіння величина $D/4$ у співвідношеннях (2.176,177) невід'ємна (для слабких дефлаграцій $0 < M_1 < a_1$, причому для нормального ламінарного горіння $M_1^2 \ll 1$ і, відповідно, $M_1^4 \ll M_1^2 \ll 1$, тобто, $1/M_1^4$ дуже велика величина). Тому рівняння (2.123) має дійсне коріння виду

$$u_{21,2} = \frac{\gamma_2 \left(\frac{1}{\gamma_1 M_1^2} + 1 \right) u_1 \pm \sqrt{\frac{D}{4}}}{\gamma_2 + 1}, \quad (2.178)$$

або ж

$$u_{21,2} = \frac{\gamma_2 \left(\frac{1}{\gamma_1 M_1^2} + 1 \right) u_1 \pm \sqrt{\left[\frac{\gamma_2^2}{\gamma_1^2 M_1^4} + \frac{4\gamma_1(1 - \gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1 - 1) M_1^2} + 1 \right] u_1^2 - 2(\gamma_2^2 - 1)q}}{\gamma_2 + 1} \quad (2.179)$$

При цьому для слабких дефлаграцій має виконуватись співвідношення $u_2 > u_1$ (це впливає із закону збереження маси, оскільки $\rho_2 < \rho_1$), у правій частині (2.179) слід вибрати знак «+».

Отже,

$$u_2 = \frac{\gamma_2 \left(\frac{1}{\gamma_1 M_1^2} + 1 \right) u_1 + \sqrt{\left[\frac{\gamma_2^2}{\gamma_1^2 M_1^4} + \frac{4\gamma_1(1 - \gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1 - 1) M_1^2} + 1 \right] u_1^2 - 2(\gamma_2^2 - 1)q}}{\gamma_2 + 1} \quad (2.180)$$

Після нескладних перетворень, з урахуванням (2.91,92), маємо

$$u_2 = \frac{\{\gamma_2(\frac{1}{\gamma_1} + M_1^2) + \sqrt{(\frac{\gamma_2}{\gamma_1})^2 + [\frac{4\gamma_1(1-\gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1-1)} - \frac{2(\gamma_2^2-1)Q}{\gamma_1}]} M_1^2 + M_1^4\} u_1}{(\gamma_2+1)M_1^2} \quad (2.181)$$

Інакше кажучи,

$$\frac{u_2}{u_1} = \frac{\gamma_2(\frac{1}{\gamma_1} + M_1^2) + \sqrt{(\frac{\gamma_2}{\gamma_1})^2 + [\frac{4\gamma_1(1-\gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1-1)} - \frac{2(\gamma_2^2-1)Q}{\gamma_1}]} M_1^2 + M_1^4}{(\gamma_2+1)M_1^2} \quad (2.182)$$

Вводячи умовне позначення

$$\delta = \frac{\rho_1}{\rho_2} \quad (2.183)$$

(величину $\delta > 1$ згідно з Дж. р. Маркштейну (Markstein) іноді називають ступенем розширення газу в полум'ї), маємо внаслідок закону збереження маси

$$\delta = \frac{u_2}{u_1} \quad (2.184)$$

та

$$\delta = \frac{\gamma_2(\frac{1}{\gamma_1} + M_1^2) + \sqrt{(\frac{\gamma_2}{\gamma_1})^2 + [\frac{4\gamma_1(1-\gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1-1)} - \frac{2(\gamma_2^2-1)Q}{\gamma_1}]} M_1^2 + M_1^4}{(\gamma_2+1)M_1^2} \quad (2.185)$$

Очевидно також, що

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma_2+1)M_1^2}{\gamma_2(\frac{1}{\gamma_1} + M_1^2) + \sqrt{(\frac{\gamma_2}{\gamma_1})^2 + [\frac{4\gamma_1(1-\gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1-1)} - \frac{2(\gamma_2^2-1)Q}{\gamma_1}]} M_1^2 + M_1^4} \quad (2.186)$$

або ж

$$\rho_2 = \frac{(\gamma_2 + 1)M_1^2 \rho_1}{\gamma_2 \left(\frac{1}{\gamma_1} + M_1^2 \right) + \sqrt{\left(\frac{\gamma_2}{\gamma_1} \right)^2 + \left[\frac{4\gamma_1(1 - \gamma_2^2) - 2\gamma_2^2}{\gamma_1(\gamma_1 - 1)} - \frac{2(\gamma_2^2 - 1)Q}{\gamma_1} \right] M_1^2 + M_1^4}} \quad (2.187)$$

Відповідно до співвідношень (2.112) та (2.136) маємо

$$\frac{p_2}{p_1} = \frac{1}{[(\gamma_2 + 1)(1 - \rho_2 / \rho_1) + 1]} \quad (2.188)$$

або

$$p_2 = \frac{p_1}{[(\gamma_2 + 1)(1 - \rho_2 / \rho_1) + 1]} \quad (2.189)$$

Таким чином, при заданих значеннях тиску p_1 і густини ρ_1 вихідного пального середовища, при відомих величинах γ_1 та γ_2 , що характеризують

вихідне пальне середовище та продукти згоряння відповідно, а також при відомому значенні питомого тепловиділення q на фронті полум'я та заданої швидкості поширення хвилі (нормального) горіння u_1 , алгоритм розрахунку параметрів продуктів згоряння такий:

1. За формулою (2.143) визначається безрозмірна величина Q .
2. Число Маха M_1 визначається як $M_1 = u_1 / a_1$, де швидкість звуку a_1 в вихідному паливному середовищі розраховується за формулою (2.109) при $\kappa=1$.
3. Швидкість u_2 продуктів згоряння щодо фронту полум'я розраховується за формулою (2.181).
4. Ступінь розширення газу в полум'ї $\delta = \frac{\rho_1}{\rho_2}$ розраховується за формулою (2.185). Величина визначається як $1/\delta$, а $\rho_2 = \rho_1 / \delta$.
5. Визначається тиск p_2 за формулою (2.189).
6. За формулою (2.188) визначається відношення p_2 / p_1 .
7. Абсолютна температура T_2 продуктів згоряння може бути розрахована за формулою (2.175).

Приведені вище, виведені аналітично формули параметрів детонаційної хвилі відсутні у відкритих виданнях.

Проведений теоретичний аналіз режимів горіння та закономірностей прямого скачку з урахуванням підведення енергії створює ґрунт для побудови моделі, здатної не лише описувати вибухові процеси, але й розраховувати їх у вигляді обчислюваних алгоритмів. Це, своєю чергою, дає змогу перейти до розробки інформаційної системи, яка реалізує зазначені математичні та фізичні залежності.

2.4 Концентраційні межі вибуховості горючих речовин

Важливою складовою аналізу вибухонебезпечності газових сумішей є визначення їх фізико-хімічних характеристик, які впливають на можливість виникнення вибуху за тих чи інших умов. Зокрема, до таких характеристик належать межі вибуховості, що вказують на концентраційні інтервали, у межах яких горюча суміш здатна вибухати за наявності джерела займання.

Межі вибуховості поділяються на:

- 1) нижню концентраційну межу вибуховості (НКМВ) — мінімальна концентрація горючого газу в повітрі, за якої можливе запалювання;
- 2) верхню концентраційну межу вибуховості (ВКМВ) — максимальна концентрація горючого газу, за якої суміш ще залишається вибухонебезпечною.

Ці параметри є специфічними для кожної газової суміші.

У таблиці нижче наведено довідкові дані щодо меж вибуховості для деяких типових горючих газів, які часто розглядаються в інженерних задачах безпеки.

Таблиця 2.1 – Межі вибухоздатності горючих сумішей та межа вибухоздатності за тиском при різних окисниках

№	Пальне	Окислювач	НКМВ(%)	ВКМВ(%)	Границя вибухоздатності за тиском (p0,мм.рт.ст.)
1	Водень (H ₂)	Кисень(O ₂)	4	94	20
		Повітря	4	75	50
2	Метан (CH ₄)	Кисень(O ₂)	5	16	20
		Повітря	5	15	50
3	Етан (C ₂ H ₆)	Кисень(O ₂)	3	56	30
		Повітря	3,2	12,5	50
4	Пропан (C ₃ H ₈)	Кисень(O ₂)	2,2	55	30
		Повітря	2,3	9,5	50
5	Етилен (C ₂ H ₄)	Кисень(O ₂)	3	80	20
		Повітря	3	55	30
6	Пропілен (C ₃ H ₆)	Кисень(O ₂)	2	53	30
		Повітря	2,4	10	50

Завдяки концентраційним межам вибухоздатності та значенню межі вибухоздатності за тиском можна відразу розв'язати питання про виникнення або відсутність вибуху.

У наступному розділі розглянемо структуру програмного забезпечення, основні функціональні модулі, логіку взаємодії між компонентами системи та архітектурні рішення, що забезпечують точність, гнучкість і зручність застосування створеної моделі на практиці.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Архітектура застосунку

Обрана архітектура програмного забезпечення базується на класичній клієнтській архітектурі настільного застосунку, реалізованого з використанням C# та Windows Forms — технології, що є частиною платформи Microsoft .NET Framework. Такий підхід забезпечує автономність, швидкодію і відсутність залежності від мережевих ресурсів, що особливо важливо у виробничих умовах або під час аварійного реагування.

Архітектура застосунку умовно поділяється на три основні логічні шари:

1) Шар представлення (Presentation Layer) — графічний інтерфейс користувача (GUI), який дозволяє вводити параметри суміші, запускати розрахунок і переглядати результати.

2) Шар логіки (Application Logic Layer) — реалізує алгоритми обробки введених даних, виконує математичні обчислення параметрів вибуху на основі заданих формул і моделей.

3) Шар доступу до даних (Data Layer) — у даному проєкті має обмежене використання, оскільки система не використовує зовнішніх баз даних, однак передбачено можливість збереження/експорту результатів у локальні файли для подальшого аналізу.

Така модульна структура дозволяє легко масштабувати застосунок, доповнювати його новими функціями (наприклад, підключенням бази даних, генерацією графіків, інтеграцією з сенсорами або хмарною платформою).

3.2 Інформаційне моделювання предметної області

Інформаційне моделювання предметної області полягає у формалізації знань про об'єкти, процеси та зв'язки, що мають значення для розрахунку

параметрів вибуху газоповітряних сумішей. Мета цього етапу — створити структуроване уявлення про систему, яке можна використати при реалізації її програмної частини.

Основні об'єкти предметної області:

1. Газоповітряна суміш:

- 1) назва газу (метан, водень, пропан тощо);
- 2) концентрація (%);
- 3) температура середовища (K);
- 4) тиск (Па або атм);
- 5) об'єм (м^3).

2. Фізико-хімічні параметри газу (довідкові дані):

- 1) нижня межа вибуховості (НМВ);
- 2) верхня межа вибуховості (ВМВ);
- 3) температура займання;
- 4) густина;
- 5) теплота згоряння;
- 6) коефіцієнт адіабати (γ).

3. Результати розрахунку:

- 1) теоретичний тиск вибуху;
- 2) температура після вибуху;
- 3) швидкість звуку в середовищі;
- 4) показники безпечності/небезпечності суміші.

4. Зв'язки між об'єктами:

- 1) газ є основою для створення суміші;
- 2) суміш формується з урахуванням даних про температуру, тиск і концентрацію;
- 3) розрахунки виконуються на основі даних суміші + довідкових параметрів газу;

4) результати залежать від коректності вхідних даних та використовуваних формул.

Модель взаємодії користувача з системою:

- 1) користувач вводить початкові параметри суміші;
- 2) система перевіряє правильність і допустимість значень;
- 3) обчислює параметри вибуху на основі математичної моделі;
- 4) виводить результати у візуальній та текстовій формі.

3.3 Вибір програмного забезпечення для створення інформаційної системи

Правильний вибір програмного забезпечення є ключовим етапом у процесі розробки інформаційної системи, оскільки він впливає на продуктивність, зручність у розробці, масштабованість та стабільність кінцевого продукту. З огляду на цілі даної дипломної роботи — створення настільного застосунку для інженерних розрахунків — було обрано технології, які забезпечують швидку розробку, доступність та широкі можливості для реалізації необхідного функціоналу.

Обрана мова програмування: C#

Мова C# (C-Sharp) була обрана як основна мова програмування, оскільки:

- 1) вона має зручний синтаксис, орієнтований на об'єктно-орієнтоване програмування;
- 2) підтримується середовищем розробки Visual Studio, що забезпечує зручний інтерфейс, відлагодження, підказки та управління проєктом;
- 3) має потужні бібліотеки для обробки числових даних, створення графічних інтерфейсів, побудови структур даних.

Для побудови інтерфейсу користувача було обрано технологію Windows Forms, яка входить до складу платформи .NET Framework. Цей вибір зумовлений:

- 1) простотою реалізації інтерфейсу;
- 2) наявністю великої кількості готових компонентів (TextBox, Button, Label тощо);
- 3) стабільною підтримкою для створення легких, автономних десктопних застосунків;
- 4) можливістю запуску без підключення до Інтернету.

Використовується Visual Studio — потужне середовище розробки, яке дозволяє:

- 1) працювати з проєктами на C#;
- 2) реалізовувати інтерфейс Windows Forms у візуальному редакторі;
- 3) виконувати налагодження (debugging), тестування і збірку застосунку;
- 4) зручно управляти структурами проєкту та залежностями.

Додаткові компоненти:

.NET Framework (версія 4.8) — як середовище виконання застосунку.

Бібліотеки для математичних обчислень (при потребі – Math.NET або власна реалізація формул).

Аргументація вибору:

Обраний стек технологій дозволяє створити автономний, простий у використанні і стабільний програмний продукт, який не потребує складного розгортання або навчання користувача. Застосунок легко адаптувати до майбутніх змін — зокрема, додавання підтримки бази даних, побудови графіків або імпорту/експорту файлів.

Таким чином, вибрані інструменти розробки повністю відповідають вимогам функціональності, надійності та простоти використання для реалізації поставленої задачі.

3.4 Програмна модель застосунку

Програмна модель застосунку визначає логічну структуру компонентів програми, принципи їх взаємодії та розподіл відповідальностей між окремими частинами системи. Це дозволяє забезпечити цілісність, масштабованість та зручність супроводу програмного коду.

Структура застосунку

Застосунок реалізовано з дотриманням принципів модульності та розділення відповідальностей. Основні модулі включають:

1) Модуль введення даних

Відповідає за збір параметрів від користувача: тип газу, температура, тиск, концентрація тощо. Реалізований через форми з елементами керування (TextBox, ComboBox, RadioButton).

2) Модуль обробки (логіки)

Містить класи, які реалізують математичні моделі вибухових процесів. Саме тут відбувається обробка введених даних та розрахунок результатів на основі формул (наприклад, використання коефіцієнта адіабати, теплоти згоряння, концентрації пального тощо).

3) Модуль відображення результатів

Призначений для виведення результатів користувачу — у вигляді числових полів, міток або таблиць. У майбутньому можливе розширення з використанням графічного представлення (графіки, діаграми).

4) Модуль перевірки коректності даних

Забезпечує контроль введених значень: перевірку на порожні поля, недопустимі значення (від'ємні числа, нуль), відповідність фізичним межам (наприклад, межам вибуховості).

Взаємодія між модулями

Програмна модель реалізує взаємодію модулів у послідовності:

1. Користувач вводить дані через графічний інтерфейс.
2. Дані перевіряються на коректність.
3. Якщо перевірка успішна — створюється об'єкт класу `GasMixture`, який передається до модуля розрахунків.
4. Модуль `Calculator` обчислює параметри вибуху.
5. Результати повертаються у вигляді об'єкта `CalculationResult` і відображаються на формі.

Класи, які реалізовано:

- 1) `Gas` — зберігає властивості вибраного газу.
- 2) `GasMixture` — зберігає вхідні параметри середовища та суміші.
- 3) `Calculator` — реалізує основні формули розрахунку.
- 4) `CalculationResult` — містить результати обчислень.
- 5) `MainForm` — головна форма інтерфейсу користувача.
- 6) Додаткові допоміжні класи для перевірки вводу та форматування виводу.

Особливості реалізації:

- 1) Обчислення виконуються в одному потоці (без багатопоточності), що достатньо для даного типу задач.
- 2) Програма не зберігає дані у базі, але передбачена можливість додати збереження у файл.
- 3) Усі обчислення виконуються локально, без доступу до Інтернету.
- 4) Такий підхід дозволяє створити стабільну, легку в обслуговуванні систему, придатну як для навчального застосування, так і для попередніх інженерних розрахунків у виробничому середовищі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Загальний опис

Програма поєднує візуальний інтерфейс з обчислювальною логікою, що базується на класичних фізико-хімічних формулах.

Програмна реалізація передбачає:

- 1) зручне введення даних користувачем;
- 2) вибір способу завдання суміші (масовий %, об'ємний %, маси, об'єми);
- 3) автоматичне обчислення теплоти згоряння;
- 4) розрахунок параметрів детонації (тиск, густина, швидкість, число Маха тощо);
- 5) обробку помилок введення.

The screenshot shows a software interface for inputting parameters. It is divided into two main sections: 'Початкові параметри середовища' (Initial environment parameters) and 'Горюча суміш' (Fuel mixture).

Початкові параметри середовища:

- Вихідне пальне сер-ще (γ_1): [input field]
- Продукти детонації (γ_2): [input field]
- Тиск до детонації (p_1 , Па): [input field]
- Густина до детонації (ρ_1 , кг/м³): [input field]
- Вибір питомого тепловиділення (q):
 - ☒ Викор. q для заданої суміші
 - q для суміші (МДж/кг): [input field]
 - ☐ Викор. q стехіометрич. суміші

Горюча суміш:

- Пальне: [dropdown menu showing 'Водень']
- Окислювач: [dropdown menu showing 'Кисень']
- ☒ Задана масовими частками (%)
 - ☐ Задана масами
- % Пального (за масою): [input field]

Рисунок 4.1 - Інтерфейс введення параметрів

The screenshot shows the results of the calculation. At the top, there are two buttons: 'Розрахувати' (Calculate) in green and 'Скинути' (Reset) in red. Below them is a section titled 'Результати розрахунку (Параметри за фронтом детонації)' (Calculation results (Parameters at the detonation front)).

The results are displayed in a grid of input fields:

- Тиск (p_2 , Па): [input field]
- Швидкість детонації (u_1 , м/с): [input field]
- Тиск (p_2 , атм): [input field]
- Шв. звуку (a_1 , м/с): [input field]
- Густина (ρ_2): [input field]
- Число Маха (M_1): [input field]
- Відн. густин ρ_2/ρ_1 : [input field]
- Шв. звуку (a_2): [input field]

Рисунок 4.2 - Інтерфейс результатів розрахунку детонаційної суміші

4.2 Інтерфейс користувача

Головна форма MainForm містить елементи керування, згенеровані у файлі MainForm.Designer.cs. Основні компоненти форми:

1) comboBoxFuel — випадаючий список для вибору пального (наприклад, "Метан (CH₄)");

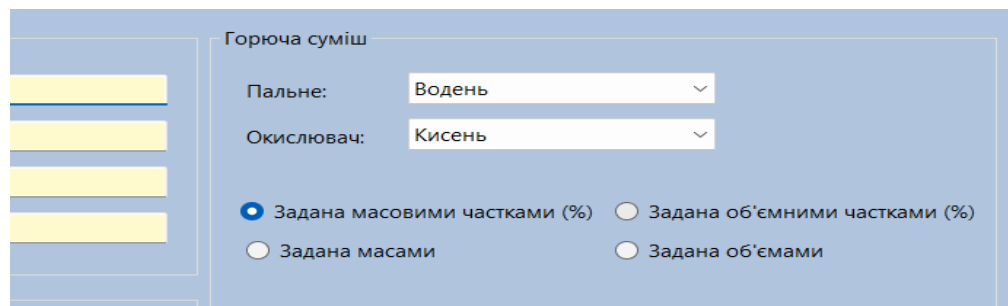


Рисунок 4.3 - Випадаючий список для вибору пального

2) comboBoxOxidizer — вибір окисника (наприклад, "Кисень (O₂)");

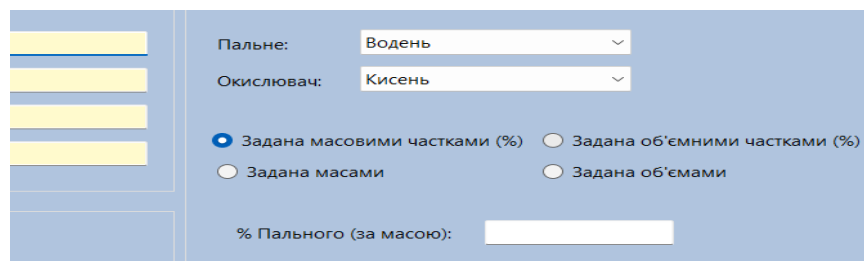


Рисунок 4.4 - Вибір окисника

3) textBoxGamma1, textBoxGamma2, textBoxP1, textBoxRho1 — поля введення основних термодинамічних параметрів середовища «див. рис. 4.1» ;

4) radioButtonMixMassPercent, radioButtonMixVolumePercent, radioButtonMixMasses, radioButtonMixVolumes — радіокнопки для вибору способу завдання суміші;

5) panelMassPercent, panelVolumePercent, panelMasses, panelVolumes — відповідні панелі, що містять поля для введення даних;

6) buttonCalculate, buttonReset — кнопки для запуску розрахунку та скидання даних.

У кодї реалізовано логіку переключення панелей залежно від вибраного режиму, наприклад:

```
private void RadioButtonMix_CheckedChanged(object sender,
EventArgs e)
{
    panelMassPercent.Visible =
radioButtonMixMassPercent.Checked;
    panelVolumePercent.Visible =
radioButtonMixVolumePercent.Checked;
    panelMasses.Visible = radioButtonMixMasses.Checked;
    panelVolumes.Visible = radioButtonMixVolumes.Checked;
}
```

Лістинг 4.1 - логіка переключення панелей залежно від вибраного режиму.

4.3 Функціональність програми

4.3.1 Ініціалізація

Під час запуску виконується ініціалізація списків газів

Приклад:

```
comboBoxFuel.Items.AddRange(new object[] {
    "Водень (H2)", "Метан (CH4)", "Етан (C2H6)",
    "Пропан (C3H8)", "Етилен (C2H4)", "Пропілен (C3H6)"
});
comboBoxOxidizer.Items.AddRange(new object[] {
    "Кисень (O2)", "Повітря (~21% O2)"
});
```

Лістинг 4.2 - ініціалізація списків газів

4.3.2 Розрахунок теплоти згоряння

Розрахунок теплоти згоряння здійснюється на основі даних із класу GasData, де для кожного пального вказано теплоти згоряння. Наприклад:

```
public static double GetHeatOfCombustion(string fuel, string
oxidizer)
{
    if (fuel == "Метан (CH4)" && oxidizer == "Кисень (O2)")
        return 50000.0; // умовне значення в кДж/кг
    // ... інші варіанти
}
```

Лістинг 4.3 - Розрахунок теплоти згоряння

Основний розрахунок виконується у методі класу DetonationCalculatorLogic:

```
public static DetonationResult Calculate(
    double gamma1, double gamma2, double p1, double rho1,
double q)
{
    double a1 = Math.Sqrt(gamma1 * p1 / rho1);
    double u1 = Math.Sqrt(2 * q);
    double p2 = p1 + rho1 * u1 * a1; // умовна формула
    // ... інші розрахунки
    return new DetonationResult { P2 = p2, A1 = a1, U1 = u1
};
}
```

Лістинг 4.4 - Розрахунок основних параметрів

4.3.3 Перевірка на вибухонебезпечність

Програма порівнює введену концентрацію пального з межами вибуховості (LEL/UEL), що зберігаються в GasData. Якщо суміш поза межами, користувача попереджають.

Приклад:

```

        if (explosionLimits.TryGetValue((fuelName, oxidizerName),
out ExplosionLimits limits))
        {
            if ((mixtureDefinition == "MassPercent" ||
mixtureDefinition == "VolumePercent") &&
                (value1 < limits.LEL || value1 > limits.UEL))
            {
                MessageBox.Show("Концентрація пального поза межами
вибухоздатності.\nВибух неможливий.", "Попередження",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                labelError.Text = "Вибух неможливий: концентрація
поза межами.";
                labelError.Visible = true;
                return;
            }

            if (p1 < limits.PressureLimitPa)
            {
                MessageBox.Show("Початковий тиск нижче порогу для
вибуху.\nВибух неможливий.", "Попередження",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                labelError.Text = "Вибух неможливий: тиск занадто
низький.";
                labelError.Visible = true;
                return;
            }
        }
    }

```

Лістинг 4.5 Перевірка на вибухонебезпечність

4.4 Обробка помилок введення

У процесі розробки застосунку було передбачено декілька рівнів перевірки коректності введених даних з метою уникнення збоїв виконання та забезпечення достовірності розрахунків.

Насамперед, усі текстові поля, в які користувач вводить числові значення (наприклад, γ_1 , γ_2 , p_1 , p_2 , масові або об'ємні частки компонентів), проходять первинну перевірку на відповідність числовому формату.

Це реалізовано за допомогою конструкцій:

```
if (!double.TryParse(textBoxGamma1.Text, out gamma1))
{
    labelError.Text = "Некоректне значення  $\gamma_1$ ";
    labelError.ForeColor = Color.Red;
    return;
}
```

Листинг 4.6 – Перевірка правильності числового введення значення γ_1

Також контролюється, чи заповнені всі необхідні поля перед запуском розрахунку. Кожне поле обробляється індивідуально, і у разі помилки користувач одразу отримує відповідне повідомлення в елементі labelError, яке підсвічується червоним кольором для привернення уваги.

Крім перевірки формату введення, система також контролює заповненість обов'язкових полів. У разі, якщо якийсь з ключових полів залишене порожнім, користувач отримує повідомлення про те, яке саме значення відсутнє. Така перевірка дозволяє уникнути ситуацій, коли розрахунок запускається з неповними даними, що могло б призвести до некоректного результату або помилки виконання.

Окрему увагу приділено перевірці фізично припустимих меж. Наприклад:

- 1) γ повинно бути більше 1 (оскільки це показник адіабати для газів);

- 2) тиск p_1 має бути додатним;
- 3) густина ρ_1 не може бути нульовою або від'ємною;
- 4) частки компонентів суміші повинні бути у межах від 0 до 100% і в сумі не перевищувати 100%.

Реалізація такого контролю виглядає наступним чином:

```
if (gamma1 <= 1.0)
{
    labelError.Text = "γ1 повинно бути більше 1";
    labelError.ForeColor = Color.Red;
    return;
}
```

Листинг 4.7 – Перевірка допустимості значення γ_1

Крім того, перед запуском розрахунку перевіряється, чи не перевищують масові або об'ємні частки допустиму суму:

```
if (fuelPart + oxidizerPart > 100)
{
    labelError.Text = "Сума часток перевищує 100%";
    labelError.ForeColor = Color.Red;
    return;
}
```

Листинг 4.8 – Контроль суми часток компонентів суміші

У ситуаціях, коли користувач не обрав пальне або окисник із випадального списку, також реалізована перевірка вибору:

```
if (comboBoxFuel.SelectedItem == null ||
comboBoxOxidizer.SelectedItem == null)
{
    labelError.Text = "Будь ласка, оберіть пальне та окисник";
}
```



```
        labelError.ForeColor = Color.Red;  
        return;  
    }
```

Листинг 4.9 – Перевірка вибору пального та окисника з випадających списків

Крім виведення текстових повідомлень, помилки можуть впливати на активність кнопок: наприклад, `buttonCalculate` може бути тимчасово заблокованою, поки не будуть введені всі необхідні значення.

Таким чином, система обробки помилок у розробленому застосунку виконує кілька функцій:

- 1) перевіряє тип введення (тільки числові значення);
- 2) контролює обов'язковість заповнення всіх ключових полів;
- 3) перевіряє фізичну допустимість значень (межі, сума часток тощо);
- 4) виводить чіткі повідомлення про помилки з візуальним виділенням;
- 5) допомагає уникнути логічно некоректних розрахунків і збоїв у роботі програми.

Завдяки цьому підхід до обробки помилок є не лише технічно ефективним, а й зручним для кінцевого користувача.

ВИСНОВКИ

У рамках виконання дипломної роботи було проведено повноцінне дослідження, спрямоване на розробку програмного забезпечення для розрахунку параметрів вибуху газоповітряних сумішей з метою підвищення ефективності та оперативності оцінки вибухонебезпечних ситуацій.

На основі аналізу предметної області було визначено фактори, що впливають на вибухонебезпечність газових сумішей, а також обґрунтовано потребу в автоматизації розрахунків за допомогою сучасних інформаційних технологій. Це підкреслило актуальність створення інструментів, здатних точно й швидко моделювати потенційно небезпечні ситуації.

На основі аналізу аналогічних програмних рішень виявлено, що існуючі продукти часто є складними у використанні, закритими, орієнтованими виключно на вузькопрофільних фахівців або вимагають підключення до Інтернету чи зовнішніх баз даних. У зв'язку з цим було обґрунтовано доцільність створення автономного настільного застосунку з відкритою структурою, орієнтованого на інженерів, викладачів, студентів і працівників техногенної безпеки.

В результаті реалізовано інформаційну систему у вигляді програмного застосунку на базі платформи Microsoft .NET Framework, з використанням мови програмування C# та інструментарію Windows Forms. Архітектура застосунку є модульною та легко розширюваною. У системі реалізовано інтуїтивно зрозумілий графічний інтерфейс, що забезпечує:

- 1) введення вхідних параметрів (тип газу, концентрація, температура, тиск);
- 2) перевірку допустимості значень;
- 3) розрахунок основних характеристик вибуху (тиск, температура, швидкість звуку, густина тощо);
- 4) вивід результатів у зручній формі.

Окрему увагу приділено математичному моделюванню. Реалізовані формули базуються на фізико-хімічних закономірностях і довідкових параметрах горючих газів. Програма дозволяє оперативно проводити розрахунки без потреби в складному налаштуванні чи підключенні до зовнішніх джерел даних.

У ході роботи виконано:

- 1) проєктування архітектури інформаційної системи;
- 2) розробку об'єктної моделі і класів об'єктів;
- 3) інформаційне моделювання предметної області;
- 4) створення програмної реалізації інформаційної системи;
- 5) тестування функціональності розробленого програмного забезпечення

на різних вхідних даних.

Розроблене програмне забезпечення може бути застосоване для розв'язання різноманітних задач з вибухобезпеки та вибухозахисту.

Таким чином, поставлену в дипломній роботі мету досягнуто. Результати роботи є актуальними та практично значущими.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zarko, V. E., Weiser, V., Eisenreich, N. & Vasil'ev, A. A. "Prevention of hazardous fires and explosions. The transfer to civil applications of military experiences". *Springer Netherlands - Kluwer Academic Publishers*. 1999. 360 p. DOI: <https://doi.org/10.1007/978-94-011-4712-5>.
2. Pekalski, A. A., Zevenbergen, J. F., Lemkowitz, S. M. & Pasman, H. J. "A review of explosion prevention and protection systems suitable as ultimate layer of protection in chemical process installations". *Process Safety and Environmental Protection*. 2005; Vol. 83 Issue 1: 1–17 DOI: <https://doi.org/10.1205/psep.04023>.
3. Nolan, D. P. "Handbook of fire and explosion protection engineering principles for oil, gas, chemical and related facilities". *William Andrew-Elsevier*. 2011. 325 p. DOI: <https://doi.org/10.1016/B978-1-4377-7857-1.00033-1>.
4. Lemkowitz, S. M. & Pasman, H. J. "A review of the fire and explosion hazards of particulate.s *KONA Powder and Particle Journal*. 2014; No. 31: 52–81. DOI: <https://10.14356/kona.2014010>.
5. Volkov, V. & Kryvchenko, Y. "Transition of combustion to explosion and decision support systems for explosion protection". In: Babichev, S., Lytvynenko, V., Wójcik, W., Vyshemyrskaya, S. (eds) "Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2020". *Advances in Intelligent Systems and Computing. Springer, Cham*. 2021; Vol. 1246: 437–447. DOI: https://doi.org/10.1007/978-3-030-54215-3_28.
6. Nettleton, M. A. "Gaseous detonations: their nature and control". *Springer*. 2013. 255 p. DOI: <https://doi.org/10.1007/978-94-009-3149-7>.
7. Volkov, V. E. "Decision support systems on hazards of industrial explosions". *Seventh International Symposium on Hazards, Prevention and Mitigation of Industrial Explosions: Thirteenth International Colloquium on Dust Explosions & Eighth Colloquium on Gas, Vapor, Liquid, and Hybrid Explosions*. St. Petersburg: Russian Federation, July 7-11. 2008; Vol. 3: 343–347.
8. Liberman, M. "Introduction to physics and chemistry of combustion". *Springer*. Berlin-Heidelberg: 2008. 349 p. DOI: <https://doi.org/10.1007/978-3-540-78759-4>.
9. Landau, L. D. "On the theory of slow combustion". In: "Collected Papers of L.D. Landau" (1st ed.). *Pergamon Press Ltd. and Gordon and Breach, Science Publishers*. 1965. p. 396–403. DOI: <https://doi.org/10.1016/B978-0-08-010586-4.50059-6>.

10. Shchelkin, K.I. & Troshin, Ya. K. "Gasdynamics of combustion". National Aeronautics and Space Administration. 1964.

11. Booch, G. & Maksimchuk, R. "Object-oriented analysis and design with applications. *Addison-Wesley Professional*. 2007. 720 p.

ДОДАТОК А

Код програмного застосунку

Основне вікно — MainForm.cs;

Призначення:

Містить головну логіку взаємодії з користувачем, обробку введених даних, виклик методів розрахунку, перевірку меж вибухоздатності, обробку подій елементів інтерфейсу та відображення результатів.

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Windows.Forms;
// using System.Drawing; // Якщо використовуєте Color напряду

namespace WinFormsApp1
{
    public partial class MainForm : Form
    {
        private DetonationCalculatorLogic calculatorLogic;
        private double calculatedQActualMix_J_kg = 0;
        private double calculatedQStoichRef_J_kg = 0;
        private const double PascalsToAtmospheres = 1.0 / 101325.0;

        // Прапорці для запобігання рекурсії подій
        private bool _isUpdatingRho1FromCode = false;
        private bool _isClearingCompositionFieldsFromCode = false;

        private class ExplosionLimits
        {
            public double LEL; // нижня межа
            public double UEL; // верхня межа
            public double PressureLimitPa;
        }

        private Dictionary<(string fuel, string oxidizer), ExplosionLimits>
explosionLimits = new()
    {
```

```

        { ("Водень", "Кисень"), new ExplosionLimits { LEL = 4, UEL = 94,
PressureLimitPa = 2666 } },
        { ("Водень", "Повітря"), new ExplosionLimits { LEL = 4, UEL =
75, PressureLimitPa = 6666 } },
        { ("Метан", "Кисень"), new ExplosionLimits { LEL = 5, UEL = 61,
PressureLimitPa = 2666 } }, // UEL уточнено
        { ("Метан", "Повітря"), new ExplosionLimits { LEL = 5, UEL = 15,
PressureLimitPa = 6666 } },
        { ("Етан", "Кисень"), new ExplosionLimits { LEL = 3.0, UEL = 66,
PressureLimitPa = 4000 } }, // LEL/UEL уточнено
        { ("Етан", "Повітря"), new ExplosionLimits { LEL = 3.0, UEL =
12.5, PressureLimitPa = 6666 } }, // LEL уточнено
        { ("Пропан", "Кисень"), new ExplosionLimits { LEL = 2.1, UEL =
55, PressureLimitPa = 4000 } }, // LEL/UEL уточнено
        { ("Пропан", "Повітря"), new ExplosionLimits { LEL = 2.1, UEL =
9.5, PressureLimitPa = 6666 } }, // LEL уточнено
        // Бутан відсутній у вашому початковому списку, але є в
comboBox. Додамо приблизні значення.
        { ("Бутан", "Кисень"), new ExplosionLimits { LEL = 1.8, UEL =
49, PressureLimitPa = 4000 } }, // Приблизні значення
        { ("Бутан", "Повітря"), new ExplosionLimits { LEL = 1.8, UEL =
8.4, PressureLimitPa = 6666 } } // Приблизні значення
    };

```

```

public MainForm()
{
    InitializeComponent();
    calculatorLogic = new DetonationCalculatorLogic();
    InitializeComboBoxes();
    SetInitialState();

    // Обробники подій для динамічного розрахунку rho1
    textBoxRho1.TextChanged += TextBoxRho1_TextChanged;

    comboBoxFuel.SelectedIndexChanged += SharedInputs_Changed;
    comboBoxOxidizer.SelectedIndexChanged +=
SharedInputs_Changed;
    textBoxP1.TextChanged += SharedInputs_Changed;

```

```

        textBoxInitialTemperature.TextChanged += SharedInputs_Changed;

        textBoxMassPercentFuel.TextChanged +=
CompositionInput_TextChanged;
        textBoxVolumePercentFuel.TextChanged +=
CompositionInput_TextChanged;
        textBoxMassFuel.TextChanged += CompositionInput_TextChanged;
        textBoxMassOxidizer.TextChanged +=
CompositionInput_TextChanged;
        textBoxVolumeFuel.TextChanged +=
CompositionInput_TextChanged;
        textBoxVolumeOxidizer.TextChanged +=
CompositionInput_TextChanged;
    }

    private void InitializeComboBoxes()
    {
        comboBoxFuel.Items.AddRange(new object[] { "Водень", "Метан",
"Пропан", "Етан", "Бутан" });
        comboBoxFuel.SelectedIndex = 0; // Водень за замовчуванням
        comboBoxOxidizer.Items.AddRange(new object[] { "Кисень",
"Повітря" });
        comboBoxOxidizer.SelectedIndex = 0; // Кисень за замовчуванням
    }

    private void SetInitialState()
    {
        _isUpdatingRho1FromCode = true; // Запобігаємо спрацьовуванню
подій під час ініціалізації
        _isClearingCompositionFieldsFromCode = true;

        textBoxGamma1.Text = ""; // Типове значення для
повітря/двоатомних газів
        textBoxGamma2.Text = ""; // Типове значення для продуктів
згоряння
        textBoxP1.Text = ""; // Атмосферний тиск в Па
        textBoxInitialTemperature.Text = ""; // Кімнатна температура
(20°C) в К

        // textBoxRho1 буде розраховано або введено користувачем

```


// Спочатку очищаємо, потім можемо спробувати розрахувати на основі значень за замовчуванням

```
textBoxRho1.Text = "";
```

```
radioButtonQUseActualMix.Checked = true;
```

```
textBoxQActualMixValue.Text = "";
```

```
textBoxQStoichRefValue.Text = "";
```

```
textBoxResultP2.Text = "";
```

```
textBoxResultP2Atm.Text = "";
```

```
textBoxResultRho2.Text = "";
```

```
textBoxResultU1.Text = "";
```

```
textBoxResultA1.Text = "";
```

```
textBoxResultRhoRatio.Text = "";
```

```
textBoxResultA2.Text = "";
```

```
textBoxResultM1.Text = "";
```

```
labelError.Text = "";
```

```
labelError.Visible = false;
```

```
radioButtonMixMassPercent.Checked = true;
```

```
ShowInputPanel("MassPercent");
```

// Встановлюємо значення за замовчуванням для масової частки, щоб rho1 розраховувалася

```
textBoxMassPercentFuel.Text = ""; // Наприклад, 10% пального
```

```
_isUpdatingRho1FromCode = false;
```

```
_isClearingCompositionFieldsFromCode = false;
```

AttemptRho1UpdateFromComposition(); // Спробувати розрахувати rho1 на основі значень за замовчуванням

```
}
```

```
private string GetCheckedMixtureDefinition()
```

```
{
```

```
    if (radioButtonMixMassPercent.Checked) return "MassPercent";
```

```
    if (radioButtonMixVolumePercent.Checked) return
```

```
"VolumePercent";
```

```
    if (radioButtonMixMasses.Checked) return "Masses";
```

```
    if (radioButtonMixVolumes.Checked) return "Volumes";
```

```
    return string.Empty;
```

```

    }

    private void ShowInputPanel(string panelName)
    {
        panelMassPercent.Visible = (panelName == "MassPercent");
        panelVolumePercent.Visible = (panelName == "VolumePercent");
        panelMasses.Visible = (panelName == "Masses");
        panelVolumes.Visible = (panelName == "Volumes");

        _isClearingCompositionFieldsFromCode = true;
        textBoxMassPercentFuel.Clear();
        textBoxVolumePercentFuel.Clear();
        textBoxMassFuel.Clear();
        textBoxMassOxidizer.Clear();
        textBoxVolumeFuel.Clear();
        textBoxVolumeOxidizer.Clear();
        _isClearingCompositionFieldsFromCode = false;
    }

    private void RadioButtonMix_CheckedChanged(object sender,
    EventArgs e)
    {
        if (!((RadioButton)sender).Checked) return;

        string panelName = GetCheckedMixtureDefinition();
        if (!string.IsNullOrEmpty(panelName))
        {
            ShowInputPanel(panelName);
            // Після зміни панелі та очищення полів, спробувати оновити
            rho1
            // Користувач повинен буде ввести нові дані для складу, щоб
            rho1 перерахувалася.
            // Або можна встановити якісь значення за замовчуванням для
            нової панелі тут.
            // Наприклад, для MassPercent можна встановити
            textBoxMassPercentFuel.Text = "10";
            // і потім викликати AttemptRho1UpdateFromComposition();
        }
    }
}

```

```

private void SharedInputs_Changed(object sender, EventArgs e)
{
    AttemptRho1UpdateFromComposition();
}

private void CompositionInput_TextChanged(object sender, EventArgs
e)
{
    if (_isClearingCompositionFieldsFromCode) return;

    var textBox = sender as TextBox;
    if (textBox == null) return;

    bool shouldAttemptUpdate = false;
    if (textBox == textBoxMassPercentFuel &&
radioButtonMixMassPercent.Checked) shouldAttemptUpdate = true;
    else if (textBox == textBoxVolumePercentFuel &&
radioButtonMixVolumePercent.Checked) shouldAttemptUpdate = true;
    else if ((textBox == textBoxMassFuel || textBox ==
textBoxMassOxidizer) && radioButtonMixMasses.Checked)
shouldAttemptUpdate = true;
    else if ((textBox == textBoxVolumeFuel || textBox ==
textBoxVolumeOxidizer) && radioButtonMixVolumes.Checked)
shouldAttemptUpdate = true;

    if (shouldAttemptUpdate)
    {
        AttemptRho1UpdateFromComposition();
    }
}

private void TextBoxRho1_TextChanged(object sender, EventArgs e)
{
    if (_isUpdatingRho1FromCode) return;

    _isClearingCompositionFieldsFromCode = true;
    string currentMixDef = GetCheckedMixtureDefinition();
    switch (currentMixDef)
    {

```

```

        case "MassPercent":
            textBoxMassPercentFuel.Clear();
            break;
        case "VolumePercent":
            textBoxVolumePercentFuel.Clear();
            break;
        case "Masses":
            textBoxMassFuel.Clear();
            textBoxMassOxidizer.Clear();
            break;
        case "Volumes":
            textBoxVolumeFuel.Clear();
            textBoxVolumeOxidizer.Clear();
            break;
    }
    _isClearingCompositionFieldsFromCode = false;
}

private void AttemptRho1UpdateFromComposition()
{
    string fuelName = comboBoxFuel.SelectedItem?.ToString();
    string oxidizerName = comboBoxOxidizer.SelectedItem?.ToString();

    if (string.IsNullOrEmpty(fuelName) ||
        string.IsNullOrEmpty(oxidizerName)) return;

    if (!double.TryParse(textBoxP1.Text, NumberStyles.Any,
        CultureInfo.CurrentCulture, out double p1_Pa) || p1_Pa <= 0)
    {
        return;
    }
    if (!double.TryParse(textBoxInitialTemperature.Text,
        NumberStyles.Any, CultureInfo.CurrentCulture, out double temp_K) || temp_K <=
        0)
    {
        return;
    }

    string mixtureDefinition = GetCheckedMixtureDefinition();
    if (string.IsNullOrEmpty(mixtureDefinition)) return;

```

```

        double v1 = 0, v2 = 0;
        bool inputsValid = TryParseCompositionInputs(mixtureDefinition,
out v1, out v2);

        if (!inputsValid)
        {
            // Якщо введення для складу суміші не валідні (наприклад,
порожні), не оновлюємо rho1
            // Можна додати очищення textBoxRho1, якщо це бажано
            // _isUpdatingRho1FromCode = true;
            // textBoxRho1.Clear();
            // _isUpdatingRho1FromCode = false;
            return;
        }

        if (TryCalculateFuelMassFraction(mixtureDefinition, fuelName,
oxidizerName, v1, v2, out double fuelMassFraction))
        {
            double calculatedDensity = GasData.CalculateMixtureDensity(
                fuelName,
                oxidizerName,
                fuelMassFraction,
                p1_Pa,
                temp_K);

            if (!double.IsNaN(calculatedDensity) && calculatedDensity > 0)
            {
                _isUpdatingRho1FromCode = true;
                textBoxRho1.Text = calculatedDensity.ToString("G4",
CultureInfo.CurrentCulture);
                _isUpdatingRho1FromCode = false;
            }
        }
    }

    private bool TryParseCompositionInputs(string mixtureDefinition, out
double v1, out double v2)
    {

```

```

v1 = 0; v2 = 0;
NumberStyles style = NumberStyles.Any;
CultureInfo culture = CultureInfo.CurrentCulture;

try
{
    switch (mixtureDefinition)
    {
        case "MassPercent":
            return double.TryParse(textBoxMassPercentFuel.Text, style,
culture, out v1) && v1 > 0 && v1 < 100;
        case "VolumePercent":
            return double.TryParse(textBoxVolumePercentFuel.Text,
style, culture, out v1) && v1 > 0 && v1 < 100;
        case "Masses":
            return double.TryParse(textBoxMassFuel.Text, style, culture,
out v1) && v1 > 0 &&
                double.TryParse(textBoxMassOxidizer.Text, style,
culture, out v2) && v2 > 0;
        case "Volumes":
            return double.TryParse(textBoxVolumeFuel.Text, style,
culture, out v1) && v1 > 0 &&
                double.TryParse(textBoxVolumeOxidizer.Text, style,
culture, out v2) && v2 > 0;
        default:
            return false;
    }
}
catch { return false; }
}

private bool TryCalculateFuelMassFraction(
    string mixtureDefinition,
    string fuelName,
    string oxidizerName,
    double val1,
    double val2,
    out double fuelMassFraction)
{
    fuelMassFraction = 0;

```

```

try
{
    switch (mixtureDefinition)
    {
        case "MassPercent":
            fuelMassFraction = val1 / 100.0;
            break;
        case "VolumePercent":
            double fuelVolumePercent = val1;
            if (fuelVolumePercent <= 0 || fuelVolumePercent >= 100)
                return false;
            double volumeRatio = fuelVolumePercent / (100.0 -
fuelVolumePercent);
            double fuelDensity = GasData.GetDensity(fuelName);
            double oxidizerDensity =
GasData.GetDensity(oxidizerName);
            if (fuelDensity <= 0 || oxidizerDensity <= 0) return false;
            double denominatorVolPercent = (fuelDensity * volumeRatio
+ oxidizerDensity);
            if (denominatorVolPercent == 0) return false;
            fuelMassFraction = (fuelDensity * volumeRatio) /
denominatorVolPercent;
            break;
        case "Masses":
            if (val1 <= 0 || val2 <= 0) return false;
            double totalMass = val1 + val2;
            if (totalMass == 0) return false;
            fuelMassFraction = val1 / totalMass;
            break;
        case "Volumes":
            if (val1 <= 0 || val2 <= 0) return false;
            double fuelDens = GasData.GetDensity(fuelName);
            double oxidizerDens = GasData.GetDensity(oxidizerName);
            if (fuelDens <= 0 || oxidizerDens <= 0) return false;
            double massFuel = fuelDens * val1;
            double massOxidizer = oxidizerDens * val2;
            double totalMassFromVolumes = massFuel + massOxidizer;
            if (totalMassFromVolumes == 0) return false;
            fuelMassFraction = massFuel / totalMassFromVolumes;

```

```

        break;
    default:
        return false;
    }
    return fuelMassFraction >= 0 && fuelMassFraction <= 1.0;
}
catch (ArgumentException) { return false; }
catch (Exception) { return false; }
}

private void buttonCalculate_Click(object sender, EventArgs e)
{
    labelError.Visible = false;
    labelError.Text = "";
    textBoxQActualMixValue.Text = "";
    textBoxQStoichRefValue.Text = "";
    calculatedQActualMix_J_kg = 0;
    calculatedQStoichRef_J_kg = 0;
    textBoxResultP2.Text = "";
    textBoxResultP2Atm.Text = "";
    textBoxResultRho2.Text = "";
    textBoxResultU1.Text = "";
    textBoxResultA1.Text = "";
    textBoxResultRhoRatio.Text = "";
    textBoxResultA2.Text = "";
    textBoxResultM1.Text = "";

    if (!double.TryParse(textBoxGamma1.Text, NumberStyles.Any,
        CultureInfo.CurrentCulture, out double gamma1) || gamma1 <= 1) {
        ShowError("Некоректне значення  $\gamma_1$  ( $> 1$ )."); return; }
    if (!double.TryParse(textBoxGamma2.Text, NumberStyles.Any,
        CultureInfo.CurrentCulture, out double gamma2) || gamma2 <= 1) {
        ShowError("Некоректне значення  $\gamma_2$  ( $> 1$ )."); return; }
    if (!double.TryParse(textBoxP1.Text, NumberStyles.Any,
        CultureInfo.CurrentCulture, out double p1) || p1 <= 0) { ShowError("Некоректне
        значення  $p_1$  ( $> 0$ )."); return; }

    // Температура вже використовується для rho1, тут не потрібна
    для основного розрахунку детонації

```



```

        if (!double.TryParse(textBoxInitialTemperature.Text,
        NumberStyles.Any, CultureInfo.CurrentCulture, out double t1) || t1 <= 0) {
        ShowError("Некоректне значення T1 (> 0)."); return; }

```

```

        if (!double.TryParse(textBoxRho1.Text, NumberStyles.Any,
        CultureInfo.CurrentCulture, out double rho1) || rho1 <= 0) {
        ShowError("Некоректне значення  $\rho_1$  (> 0)."); return; }

```

```

        string fuelName = comboBoxFuel.SelectedItem?.ToString();
        string oxidizerName = comboBoxOxidizer.SelectedItem?.ToString();
        if (string.IsNullOrEmpty(fuelName) ||
        string.IsNullOrEmpty(oxidizerName)) { ShowError("Оберіть пальне та
        окислювач."); return; }

```

```

        string mixtureDefinition = GetCheckedMixtureDefinition();
        double value1_for_q = 0, value2_for_q = 0;

```

```

        if (!TryParseCompositionInputs(mixtureDefinition, out
        value1_for_q, out value2_for_q))
        {
            ShowError($"Некоректні або порожні дані для задання суміші
            '{mixtureDefinition}' для розрахунку q.");
            return;
        }

```

```

        // Додаткова перевірка діапазонів для відсотків, якщо TryParse
        пройшов, але значення поза межами

```

```

        if (mixtureDefinition == "MassPercent" && (value1_for_q <= 0 ||
        value1_for_q >= 100))
        { ShowError("Масова частка пального для q має бути в діапазоні
        (0, 100)."); return; }

```

```

        if (mixtureDefinition == "VolumePercent" && (value1_for_q <= 0 ||
        value1_for_q >= 100))
        { ShowError("Об'ємна частка пального для q має бути в діапазоні
        (0, 100)."); return; }

```

```

        if (mixtureDefinition == "VolumePercent" || mixtureDefinition ==
        "Volumes")

```

```

    {
        // Для розрахунку q за об'ємними методами,
        GasData.GetDensity використовується всередині DetonationCalculatorLogic,
        // який використовує густини за нормальних умов.
        // MessageBox.Show("Розрахунок q для об'ємних
        часток/об'ємів використовує густини за нормальних умов.", "Попередження
        q", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    HeatReleaseResult heatResult =
    calculatorLogic.CalculateSpecificHeatReleases(
        fuelName, oxidizerName, mixtureDefinition, value1_for_q,
        value2_for_q);

    if (!string.IsNullOrEmpty(heatResult.ErrorMessage)) {
        ShowError(heatResult.ErrorMessage); return; }

    calculatedQActualMix_J_kg = heatResult.Q_ActualMixture_J_kg;
    calculatedQStoichRef_J_kg =
    heatResult.Q_StoichiometricReference_J_kg;
    textBoxQActualMixValue.Text = (calculatedQActualMix_J_kg /
    1e6).ToString("G4", CultureInfo.CurrentCulture);
    textBoxQStoichRefValue.Text = (calculatedQStoichRef_J_kg /
    1e6).ToString("G4", CultureInfo.CurrentCulture);

    double qToUse;
    if (radioButtonQUseActualMix.Checked) { qToUse =
    calculatedQActualMix_J_kg; }
    else if (radioButtonQUseStoichRef.Checked) { qToUse =
    calculatedQStoichRef_J_kg; }
    else { ShowError("Не обрано тип використовуваного q."); return; }

    if (qToUse < 0) // Додаткова перевірка для q
    {
        ShowError("Розраховане значення q є від'ємним. Розрахунок
        детонації неможливий.");
        return;
    }

```

```

        DetonationResult result =
calculatorLogic.CalculateDetonationParameters(gamma1, gamma2, p1, rho1,
qToUse);

        if (!string.IsNullOrEmpty(result.ErrorMessage))
        {
            ShowError(result.ErrorMessage);
            textBoxResultP2.Text = "Помилка";
            textBoxResultP2Atm.Text = "Помилка";
            textBoxResultRho2.Text = "Помилка";
            textBoxResultU1.Text = "Помилка";
            textBoxResultA1.Text = "Помилка";
            textBoxResultRhoRatio.Text = "Помилка";
            textBoxResultA2.Text = "Помилка";
            textBoxResultM1.Text = "Помилка";
        }
        else
        {
            textBoxResultP2.Text = $"{result.Pressure:N1}";
            textBoxResultP2Atm.Text = $"{(result.Pressure *
PascalsToAtmospheres):F3}";
            textBoxResultRho2.Text = $"{result.Density:N3}";
            textBoxResultU1.Text = $"{result.Velocity:N1}";
            textBoxResultA1.Text = $"{result.a1_Velocity:N1}";
            textBoxResultRhoRatio.Text = $"{result.DensityRatio:N2}";
            textBoxResultA2.Text = double.IsNaN(result.a2_Velocity) ?
"Помилка" : $"{result.a2_Velocity:N1}";
            textBoxResultM1.Text = double.IsNaN(result.MachNumber) ?
"Помилка" : $"{result.MachNumber:N2}";
        }

        if (explosionLimits.TryGetValue((fuelName, oxidizerName), out
ExplosionLimits limits))
        {
            double concentrationToCheck = -1;
            if (mixtureDefinition == "VolumePercent") // Межі вибуховості
зазвичай даються в об'ємних %
            {
                concentrationToCheck = value1_for_q;
            }
        }
    }
}

```

```

    }
    else if (mixtureDefinition == "MassPercent")
    {
        // Для порівняння з об'ємними межами, потрібно
        конвертувати масову частку в об'ємну
        // Це складніше і вимагає знання густин компонентів за
        умов, для яких дані межі (зазвичай Н.У.)
        // Наразі, якщо межі задані для об'ємних відсотків, а
        введено масові, цю перевірку пропускаємо
        // або виводимо попередження про неможливість точного
        порівняння.
        // Для простоти, поки що перевіряємо тільки якщо введено
        об'ємні відсотки.
    }

    if (concentrationToCheck != -1 && (concentrationToCheck <
    limits.LEL || concentrationToCheck > limits.UEL))
    {
        //MessageBox.Show($"Концентрація пального
        ({concentrationToCheck:F1} об.%) поза межами вибухоздатності
        ({limits.LEL}-{limits.UEL} об.%).\nВибух неможливий.", "Попередження",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        ShowError($"Концентрація ({concentrationToCheck:F1}
        об.%) поза межами вибухоздатності ({limits.LEL}-{limits.UEL} об.%).
        Детонація неможлива.");
        //return; // Можливо, варто зупинити подальшу обробку або
        очистити результати
    }

    if (p1 < limits.PressureLimitPa)
    {
        //MessageBox.Show($"Початковий тиск ({p1:F0} Па) нижче
        порогу для вибуху ({limits.PressureLimitPa:F0} Па).\nВибух неможливий.",
        "Попередження", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        ShowError($"Тиск ({p1:F0} Па) нижче порогу
        ({limits.PressureLimitPa:F0} Па). Детонація неможлива.");
        //return;
    }
}
}

```

```

private void ShowError(string message)
{
    labelError.Text = "Помилка: " + message;
    labelError.Visible = true;
}

private void buttonReset_Click(object sender, EventArgs e)
{
    // SetInitialState вже має прапорці, тому тут вони не потрібні
    SetInitialState();
}
}
}

```

Графічний інтерфейс — MainForm.Designer.cs;

Призначення:

Визначає компонування та оформлення графічного інтерфейсу користувача: поля введення, кнопки, панелі, стилі елементів, їх розташування на формі. Забезпечує зручність взаємодії та читабельність результатів.

```

using System.ComponentModel;

using System.Drawing; // Додано для Color

```

```

namespace WinFormsApp1
{
    partial class MainForm
    {
        /// <summary>

        /// Required designer variable.

        /// </summary>
    }
}

```

```

private System.ComponentModel.IContainer components = null;

/// <summary>
/// Clean up any resources being used.
/// </summary>
/// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{

```

```
labelGamma1 = new Label();  
textBoxGamma1 = new TextBox();  
labelGamma2 = new Label();  
textBoxGamma2 = new TextBox();  
labelP1 = new Label();  
textBoxP1 = new TextBox();  
labelInitialTemperature = new Label();  
textBoxInitialTemperature = new TextBox();  
labelRho1 = new Label();  
textBoxRho1 = new TextBox();  
groupBoxInput = new GroupBox();  
groupBoxMixture = new GroupBox();  
panelVolumes = new Panel();  
textBoxVolumeOxidizer = new TextBox();  
labelVolumeOxidizer = new Label();  
textBoxVolumeFuel = new TextBox();  
labelVolumeFuel = new Label();  
panelMasses = new Panel();  
textBoxMassOxidizer = new TextBox();  
labelMassOxidizer = new Label();  
textBoxMassFuel = new TextBox();  
labelMassFuel = new Label();  
panelVolumePercent = new Panel();
```

```
textBoxVolumePercentFuel = new TextBox();  
  
labelVolumePercentFuel = new Label();  
  
panelMassPercent = new Panel();  
  
textBoxMassPercentFuel = new TextBox();  
  
labelMassPercentFuel = new Label();  
  
radioButtonMixVolumes = new RadioButton();  
  
radioButtonMixMasses = new RadioButton();  
  
radioButtonMixVolumePercent = new RadioButton();  
  
radioButtonMixMassPercent = new RadioButton();  
  
comboBoxOxidizer = new ComboBox();  
  
labelOxidizer = new Label();  
  
comboBoxFuel = new ComboBox();  
  
labelFuel = new Label();  
  
groupBoxQ = new GroupBox();  
  
textBoxQStoichRefValue = new TextBox();  
  
labelQStoichRef = new Label();  
  
textBoxQActualMixValue = new TextBox();  
  
labelQActualMix = new Label();  
  
radioButtonQUseStoichRef = new RadioButton();  
  
radioButtonQUseActualMix = new RadioButton();  
  
buttonCalculate = new Button();  
  
groupBoxResults = new GroupBox();  
  
textBoxResultM1 = new TextBox();
```



```
labelResultM1 = new Label();  
textBoxResultA2 = new TextBox();  
labelResultA2 = new Label();  
textBoxResultRhoRatio = new TextBox();  
labelResultRhoRatio = new Label();  
textBoxResultA1 = new TextBox();  
labelResultA1 = new Label();  
textBoxResultU1 = new TextBox();  
labelResultU1 = new Label();  
textBoxResultRho2 = new TextBox();  
labelResultRho2 = new Label();  
textBoxResultP2Atm = new TextBox();  
labelResultP2Atm = new Label();  
textBoxResultP2 = new TextBox();  
labelResultP2 = new Label();  
labelError = new Label();  
buttonReset = new Button();  
groupBoxInput.SuspendLayout();  
groupBoxMixture.SuspendLayout();  
panelVolumes.SuspendLayout();  
panelMasses.SuspendLayout();  
panelVolumePercent.SuspendLayout();  
panelMassPercent.SuspendLayout();
```

```
groupBoxQ.SuspendLayout();

groupBoxResults.SuspendLayout();

SuspendLayout();

//

// labelGamma1

//

labelGamma1.AutoSize = true;

labelGamma1.Location = new Point(20, 46);

labelGamma1.Margin = new Padding(4, 0, 4, 0);

labelGamma1.Name = "labelGamma1";

labelGamma1.Size = new Size(237, 20);

labelGamma1.TabIndex = 0;

labelGamma1.Text = "Гамма 1 для горючої суміші ( $\gamma_1$ ):";

//

// textBoxGamma1

//

textBoxGamma1.BackColor = Color.LemonChiffon;

textBoxGamma1.ForeColor = Color.Navy;

textBoxGamma1.Location = new Point(287, 42);

textBoxGamma1.Margin = new Padding(4, 5, 4, 5);

textBoxGamma1.Name = "textBoxGamma1";

textBoxGamma1.Size = new Size(132, 27);

textBoxGamma1.TabIndex = 1;
```

```
//  
  
// labelGamma2  
  
//  
  
labelGamma2.AutoSize = true;  
  
labelGamma2.Location = new Point(20, 86);  
  
labelGamma2.Margin = new Padding(4, 0, 4, 0);  
  
labelGamma2.Name = "labelGamma2";  
  
labelGamma2.Size = new Size(267, 20);  
  
labelGamma2.TabIndex = 2;  
  
labelGamma2.Text = "Гамма 2 для продуктів детонації ( $\gamma_2$ ):";  
  
//  
  
// textBoxGamma2  
  
//  
  
textBoxGamma2.BackColor = Color.LemonChiffon;  
  
textBoxGamma2.ForeColor = Color.Black;  
  
textBoxGamma2.Location = new Point(287, 82);  
  
textBoxGamma2.Margin = new Padding(4, 5, 4, 5);  
  
textBoxGamma2.Name = "textBoxGamma2";  
  
textBoxGamma2.Size = new Size(132, 27);  
  
textBoxGamma2.TabIndex = 3;  
  
//  
  
// labelP1  
  
//
```

```
labelP1.AutoSize = true;

labelP1.Location = new Point(20, 126);

labelP1.Margin = new Padding(4, 0, 4, 0);

labelP1.Name = "labelP1";

labelP1.Size = new Size(190, 20);

labelP1.TabIndex = 4;

labelP1.Text = "Тиск до детонації (p1, Па):";

//

// textBoxP1

//

textBoxP1.BackColor = Color.LemonChiffon;

textBoxP1.ForeColor = Color.Black;

textBoxP1.Location = new Point(287, 122);

textBoxP1.Margin = new Padding(4, 5, 4, 5);

textBoxP1.Name = "textBoxP1";

textBoxP1.Size = new Size(132, 27);

textBoxP1.TabIndex = 5;

//

// labelInitialTemperature

//

labelInitialTemperature.AutoSize = true;

labelInitialTemperature.Location = new Point(20, 166);

labelInitialTemperature.Margin = new Padding(4, 0, 4, 0);
```

```
labelInitialTemperature.Name = "labelInitialTemperature";

labelInitialTemperature.Size = new Size(225, 20);

labelInitialTemperature.TabIndex = 6;

labelInitialTemperature.Text = "Початкова температура (T1, K)";

//

// textBoxInitialTemperature

//

textBoxInitialTemperature.BackColor = Color.LemonChiffon;

textBoxInitialTemperature.ForeColor = Color.Black;

textBoxInitialTemperature.Location = new Point(287, 162);

textBoxInitialTemperature.Margin = new Padding(4, 5, 4, 5);

textBoxInitialTemperature.Name = "textBoxInitialTemperature";

textBoxInitialTemperature.Size = new Size(132, 27);

textBoxInitialTemperature.TabIndex = 7;

//

// labelRho1

//

labelRho1.AutoSize = true;

labelRho1.Location = new Point(20, 206);

labelRho1.Margin = new Padding(4, 0, 4, 0);

labelRho1.Name = "labelRho1";

labelRho1.Size = new Size(229, 20);

labelRho1.TabIndex = 8;
```

```
labelRho1.Text = "Густина до детонації ( $\rho_1$ , кг/м³):";

//

// textBoxRho1

//

textBoxRho1.BackColor = Color.LemonChiffon;

textBoxRho1.ForeColor = Color.Black;

textBoxRho1.Location = new Point(287, 202);

textBoxRho1.Margin = new Padding(4, 5, 4, 5);

textBoxRho1.Name = "textBoxRho1";

textBoxRho1.Size = new Size(132, 27);

textBoxRho1.TabIndex = 9;

//

// groupBoxInput

//

groupBoxInput.Controls.Add(labelGamma1);

groupBoxInput.Controls.Add(textBoxGamma1);

groupBoxInput.Controls.Add(labelGamma2);

groupBoxInput.Controls.Add(textBoxGamma2);

groupBoxInput.Controls.Add(labelP1);

groupBoxInput.Controls.Add(textBoxP1);

groupBoxInput.Controls.Add(labelInitialTemperature);

groupBoxInput.Controls.Add(textBoxInitialTemperature);

groupBoxInput.Controls.Add(labelRho1);
```

```

groupBoxInput.Controls.Add(textBoxRho1);

groupBoxInput.Location = new Point(16, 18);

groupBoxInput.Margin = new Padding(4, 5, 4, 5);

groupBoxInput.Name = "groupBoxInput";

groupBoxInput.Padding = new Padding(4, 5, 4, 5);

groupBoxInput.Size = new Size(440, 248);

groupBoxInput.TabIndex = 0;

groupBoxInput.TabStop = false;

groupBoxInput.Text = "Газодинамічні параметри";

//

// groupBoxMixture

//

groupBoxMixture.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;

groupBoxMixture.Controls.Add(panelVolumes);

groupBoxMixture.Controls.Add(panelMasses);

groupBoxMixture.Controls.Add(panelVolumePercent);

groupBoxMixture.Controls.Add(panelMassPercent);

groupBoxMixture.Controls.Add(radioButtonMixVolumes);

groupBoxMixture.Controls.Add(radioButtonMixMasses);

groupBoxMixture.Controls.Add(radioButtonMixVolumePercent);

groupBoxMixture.Controls.Add(radioButtonMixMassPercent);

groupBoxMixture.Controls.Add(comboBoxOxidizer);

```

```

groupBoxMixture.Controls.Add(labelOxidizer);

groupBoxMixture.Controls.Add(comboBoxFuel);

groupBoxMixture.Controls.Add(labelFuel);

groupBoxMixture.Location = new Point(464, 18);

groupBoxMixture.Margin = new Padding(4, 5, 4, 5);

groupBoxMixture.Name = "groupBoxMixture";

groupBoxMixture.Padding = new Padding(4, 5, 4, 5);

groupBoxMixture.Size = new Size(540, 412);

groupBoxMixture.TabIndex = 1;

groupBoxMixture.TabStop = false;

groupBoxMixture.Text = "Горюча суміш";

//

// panelVolumes

//

panelVolumes.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;

panelVolumes.Controls.Add(textBoxVolumeOxidizer);

panelVolumes.Controls.Add(labelVolumeOxidizer);

panelVolumes.Controls.Add(textBoxVolumeFuel);

panelVolumes.Controls.Add(labelVolumeFuel);

panelVolumes.Location = new Point(25, 240);

panelVolumes.Margin = new Padding(4, 5, 4, 5);

panelVolumes.Name = "panelVolumes";

panelVolumes.Size = new Size(491, 89);

```



```
panelVolumes.TabIndex = 11;

panelVolumes.Visible = false;

//

// textBoxVolumeOxidizer

//

textBoxVolumeOxidizer.Location = new Point(208, 48);
textBoxVolumeOxidizer.Margin = new Padding(4, 5, 4, 5);
textBoxVolumeOxidizer.Name = "textBoxVolumeOxidizer";
textBoxVolumeOxidizer.Size = new Size(147, 27);
textBoxVolumeOxidizer.TabIndex = 3;

//

// labelVolumeOxidizer

//

labelVolumeOxidizer.AutoSize = true;
labelVolumeOxidizer.Location = new Point(11, 52);
labelVolumeOxidizer.Margin = new Padding(4, 0, 4, 0);
labelVolumeOxidizer.Name = "labelVolumeOxidizer";
labelVolumeOxidizer.Size = new Size(155, 20);
labelVolumeOxidizer.TabIndex = 2;
labelVolumeOxidizer.Text = "Об'єм окисл. (V2, м³)";

//

// textBoxVolumeFuel

//
```

```

textBoxVolumeFuel.Location = new Point(208, 8);

textBoxVolumeFuel.Margin = new Padding(4, 5, 4, 5);

textBoxVolumeFuel.Name = "textBoxVolumeFuel";

textBoxVolumeFuel.Size = new Size(147, 27);

textBoxVolumeFuel.TabIndex = 1;

//

// labelVolumeFuel

//

labelVolumeFuel.AutoSize = true;

labelVolumeFuel.Location = new Point(11, 12);

labelVolumeFuel.Margin = new Padding(4, 0, 4, 0);

labelVolumeFuel.Name = "labelVolumeFuel";

labelVolumeFuel.Size = new Size(178, 20);

labelVolumeFuel.TabIndex = 0;

labelVolumeFuel.Text = "Об'єм пального (V1, м³):";

//

// panelMasses

//

panelMasses.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;

panelMasses.Controls.Add(textBoxMassOxidizer);

panelMasses.Controls.Add(labelMassOxidizer);

panelMasses.Controls.Add(textBoxMassFuel);

panelMasses.Controls.Add(labelMassFuel);

```

```
panelMasses.Location = new Point(25, 240);

panelMasses.Margin = new Padding(4, 5, 4, 5);

panelMasses.Name = "panelMasses";

panelMasses.Size = new Size(491, 89);

panelMasses.TabIndex = 10;

panelMasses.Visible = false;

//

// textBoxMassOxidizer

//

textBoxMassOxidizer.Location = new Point(208, 48);

textBoxMassOxidizer.Margin = new Padding(4, 5, 4, 5);

textBoxMassOxidizer.Name = "textBoxMassOxidizer";

textBoxMassOxidizer.Size = new Size(147, 27);

textBoxMassOxidizer.TabIndex = 3;

//

// labelMassOxidizer

//

labelMassOxidizer.AutoSize = true;

labelMassOxidizer.Location = new Point(11, 52);

labelMassOxidizer.Margin = new Padding(4, 0, 4, 0);

labelMassOxidizer.Name = "labelMassOxidizer";

labelMassOxidizer.Size = new Size(150, 20);

labelMassOxidizer.TabIndex = 2;
```

```

labelMassOxidizer.Text = "Маса окисл. (m2, кг):";

//

// textBoxMassFuel

//

textBoxMassFuel.Location = new Point(208, 8);
textBoxMassFuel.Margin = new Padding(4, 5, 4, 5);
textBoxMassFuel.Name = "textBoxMassFuel";
textBoxMassFuel.Size = new Size(147, 27);
textBoxMassFuel.TabIndex = 1;

//

// labelMassFuel

//

labelMassFuel.AutoSize = true;
labelMassFuel.Location = new Point(11, 12);
labelMassFuel.Margin = new Padding(4, 0, 4, 0);
labelMassFuel.Name = "labelMassFuel";
labelMassFuel.Size = new Size(173, 20);
labelMassFuel.TabIndex = 0;
labelMassFuel.Text = "Маса пального (m1, кг):";

//

// panelVolumePercent

//

panelVolumePercent.Anchor = AnchorStyles.Top | AnchorStyles.Left
| AnchorStyles.Right;

```

```
panelVolumePercent.Controls.Add(textBoxVolumePercentFuel);  
panelVolumePercent.Controls.Add(labelVolumePercentFuel);  
panelVolumePercent.Location = new Point(25, 240);  
panelVolumePercent.Margin = new Padding(4, 5, 4, 5);  
panelVolumePercent.Name = "panelVolumePercent";  
panelVolumePercent.Size = new Size(491, 54);  
panelVolumePercent.TabIndex = 9;  
panelVolumePercent.Visible = false;  
  
//  
  
// textBoxVolumePercentFuel  
  
//  
  
textBoxVolumePercentFuel.Location = new Point(208, 8);  
textBoxVolumePercentFuel.Margin = new Padding(4, 5, 4, 5);  
textBoxVolumePercentFuel.Name = "textBoxVolumePercentFuel";  
textBoxVolumePercentFuel.Size = new Size(147, 27);  
textBoxVolumePercentFuel.TabIndex = 1;  
  
//  
  
// labelVolumePercentFuel  
  
//  
  
labelVolumePercentFuel.AutoSize = true;  
labelVolumePercentFuel.Location = new Point(11, 12);  
labelVolumePercentFuel.Margin = new Padding(4, 0, 4, 0);  
labelVolumePercentFuel.Name = "labelVolumePercentFuel";
```

```

labelVolumePercentFuel.Size = new Size(188, 20);

labelVolumePercentFuel.TabIndex = 0;

labelVolumePercentFuel.Text = "% Пального (за об'ємом)";

//

// panelMassPercent

//

panelMassPercent.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;

panelMassPercent.Controls.Add(textBoxMassPercentFuel);

panelMassPercent.Controls.Add(labelMassPercentFuel);

panelMassPercent.Location = new Point(25, 240);

panelMassPercent.Margin = new Padding(4, 5, 4, 5);

panelMassPercent.Name = "panelMassPercent";

panelMassPercent.Size = new Size(491, 54);

panelMassPercent.TabIndex = 8;

//

// textBoxMassPercentFuel

//

textBoxMassPercentFuel.Location = new Point(208, 8);

textBoxMassPercentFuel.Margin = new Padding(4, 5, 4, 5);

textBoxMassPercentFuel.Name = "textBoxMassPercentFuel";

textBoxMassPercentFuel.Size = new Size(147, 27);

textBoxMassPercentFuel.TabIndex = 1;

//

```

```

// labelMassPercentFuel

//

labelMassPercentFuel.AutoSize = true;

labelMassPercentFuel.Location = new Point(11, 12);

labelMassPercentFuel.Margin = new Padding(4, 0, 4, 0);

labelMassPercentFuel.Name = "labelMassPercentFuel";

labelMassPercentFuel.Size = new Size(176, 20);

labelMassPercentFuel.TabIndex = 0;

labelMassPercentFuel.Text = "% Пального (за масою)";

//

// radioButtonMixVolumes

//

radioButtonMixVolumes.AutoSize = true;

radioButtonMixVolumes.Location = new Point(278, 183);

radioButtonMixVolumes.Margin = new Padding(4, 5, 4, 5);

radioButtonMixVolumes.Name = "radioButtonMixVolumes";

radioButtonMixVolumes.Size = new Size(150, 24);

radioButtonMixVolumes.TabIndex = 7;

radioButtonMixVolumes.Text = "Задана об'ємами";

radioButtonMixVolumes.UseVisualStyleBackColor = true;

radioButtonMixVolumes.CheckedChanged
RadioButtonMix_CheckedChanged;

//

```

+=

```

// radioButtonMixMasses

//

radioButtonMixMasses.AutoSize = true;

radioButtonMixMasses.Location = new Point(25, 183);

radioButtonMixMasses.Margin = new Padding(4, 5, 4, 5);

radioButtonMixMasses.Name = "radioButtonMixMasses";

radioButtonMixMasses.Size = new Size(137, 24);

radioButtonMixMasses.TabIndex = 6;

radioButtonMixMasses.Text = "Задана масами";

radioButtonMixMasses.UseVisualStyleBackColor = true;

radioButtonMixMasses.CheckedChanged +=
RadioButtonMix_CheckedChanged;

//

// radioButtonMixVolumePercent

//

radioButtonMixVolumePercent.AutoSize = true;

radioButtonMixVolumePercent.Location = new Point(278, 149);

radioButtonMixVolumePercent.Margin = new Padding(4, 5, 4, 5);

radioButtonMixVolumePercent.Name =

"radioButtonMixVolumePercent";

radioButtonMixVolumePercent.Size = new Size(254, 24);

radioButtonMixVolumePercent.TabIndex = 5;

radioButtonMixVolumePercent.Text = "Задана об'ємними частками
(%)";

```



```

        radioButtonMixVolumePercent.UseVisualStyleBackColor = true;

        radioButtonMixVolumePercent.CheckedChanged +=
RadioButtonMix_CheckedChanged;

        //

        // radioButtonMixMassPercent

        //

        radioButtonMixMassPercent.AutoSize = true;

        radioButtonMixMassPercent.Checked = true;

        radioButtonMixMassPercent.Location = new Point(21, 149);

        radioButtonMixMassPercent.Margin = new Padding(4, 5, 4, 5);

        radioButtonMixMassPercent.Name =
"radioButtonMixMassPercent";

        radioButtonMixMassPercent.Size = new Size(249, 24);

        radioButtonMixMassPercent.TabIndex = 4;

        radioButtonMixMassPercent.TabStop = true;

        radioButtonMixMassPercent.Text = "Задана масовими частками
(%)";

        radioButtonMixMassPercent.UseVisualStyleBackColor = true;

        radioButtonMixMassPercent.CheckedChanged +=
RadioButtonMix_CheckedChanged;

        //

        // comboBoxOxidizer

        //

```

```
comboBoxOxidizer.DropDownStyle
ComboBoxStyle.DropDownList;

comboBoxOxidizer.FormattingEnabled = true;

comboBoxOxidizer.Location = new Point(136, 80);

comboBoxOxidizer.Margin = new Padding(4, 5, 4, 5);

comboBoxOxidizer.Name = "comboBoxOxidizer";

comboBoxOxidizer.Size = new Size(211, 28);

comboBoxOxidizer.TabIndex = 3;

//

// labelOxidizer

//

labelOxidizer.AutoSize = true;

labelOxidizer.Location = new Point(21, 86);

labelOxidizer.Margin = new Padding(4, 0, 4, 0);

labelOxidizer.Name = "labelOxidizer";

labelOxidizer.Size = new Size(90, 20);

labelOxidizer.TabIndex = 2;

labelOxidizer.Text = "Окислювач:";

//

// comboBoxFuel

//

comboBoxFuel.DropDownStyle = ComboBoxStyle.DropDownList;

comboBoxFuel.FormattingEnabled = true;

comboBoxFuel.Location = new Point(136, 40);
```

```
comboBoxFuel.Margin = new Padding(4, 5, 4, 5);

comboBoxFuel.Name = "comboBoxFuel";

comboBoxFuel.Size = new Size(211, 28);

comboBoxFuel.TabIndex = 1;

//

// labelFuel

//

labelFuel.AutoSize = true;

labelFuel.Location = new Point(21, 46);

labelFuel.Margin = new Padding(4, 0, 4, 0);

labelFuel.Name = "labelFuel";

labelFuel.Size = new Size(64, 20);

labelFuel.TabIndex = 0;

labelFuel.Text = "Пальне:";

//

// groupBoxQ

//

groupBoxQ.Controls.Add(textBoxQStoichRefValue);

groupBoxQ.Controls.Add(labelQStoichRef);

groupBoxQ.Controls.Add(textBoxQActualMixValue);

groupBoxQ.Controls.Add(labelQActualMix);

groupBoxQ.Controls.Add(radioButtonQUseStoichRef);

groupBoxQ.Controls.Add(radioButtonQUseActualMix);
```

```
groupBoxQ.Location = new Point(16, 276);

groupBoxQ.Margin = new Padding(4, 5, 4, 5);

groupBoxQ.Name = "groupBoxQ";

groupBoxQ.Padding = new Padding(4, 5, 4, 5);

groupBoxQ.Size = new Size(440, 185);

groupBoxQ.TabIndex = 2;

groupBoxQ.TabStop = false;

groupBoxQ.Text = "Вибір питомого тепловиділення (q)";

//

// textBoxQStoichRefValue

//

textBoxQStoichRefValue.Location = new Point(287, 135);

textBoxQStoichRefValue.Margin = new Padding(4, 5, 4, 5);

textBoxQStoichRefValue.Name = "textBoxQStoichRefValue";

textBoxQStoichRefValue.ReadOnly = true;

textBoxQStoichRefValue.Size = new Size(132, 27);

textBoxQStoichRefValue.TabIndex = 5;

textBoxQStoichRefValue.TabStop = false;

//

// labelQStoichRef

//

labelQStoichRef.AutoSize = true;

labelQStoichRef.Location = new Point(20, 140);
```

```

labelQStoichRef.Margin = new Padding(4, 0, 4, 0);

labelQStoichRef.Name = "labelQStoichRef";

labelQStoichRef.Size = new Size(147, 20);

labelQStoichRef.TabIndex = 4;

labelQStoichRef.Text = "q стexiom. (МДж/кг)";

//

// textBoxQActualMixValue

//

textBoxQActualMixValue.Location = new Point(287, 65);

textBoxQActualMixValue.Margin = new Padding(4, 5, 4, 5);

textBoxQActualMixValue.Name = "textBoxQActualMixValue";

textBoxQActualMixValue.ReadOnly = true;

textBoxQActualMixValue.Size = new Size(132, 27);

textBoxQActualMixValue.TabIndex = 2;

textBoxQActualMixValue.TabStop = false;

//

// labelQActualMix

//

labelQActualMix.AutoSize = true;

labelQActualMix.Location = new Point(20, 69);

labelQActualMix.Margin = new Padding(4, 0, 4, 0);

labelQActualMix.Name = "labelQActualMix";

labelQActualMix.Size = new Size(165, 20);

```

```

labelQActualMix.TabIndex = 1;

labelQActualMix.Text = "q для суміші (МДж/кг):";

//

// radioButtonQUseStoichRef

//

radioButtonQUseStoichRef.AutoSize = true;

radioButtonQUseStoichRef.Location = new Point(24, 106);

radioButtonQUseStoichRef.Margin = new Padding(4, 5, 4, 5);

radioButtonQUseStoichRef.Name = "radioButtonQUseStoichRef";

radioButtonQUseStoichRef.Size = new Size(309, 24);

radioButtonQUseStoichRef.TabIndex = 3;

radioButtonQUseStoichRef.Text      =      "Використання      q
стехіометричної суміші";

radioButtonQUseStoichRef.UseVisualStyleBackColor = true;

//

// radioButtonQUseActualMix

//

radioButtonQUseActualMix.AutoSize = true;

radioButtonQUseActualMix.Checked = true;

radioButtonQUseActualMix.Location = new Point(24, 29);

radioButtonQUseActualMix.Margin = new Padding(4, 5, 4, 5);

radioButtonQUseActualMix.Name = "radioButtonQUseActualMix";

radioButtonQUseActualMix.Size = new Size(276, 24);

radioButtonQUseActualMix.TabIndex = 0;

```

```

radioButtonQUseActualMix.TabStop = true;

radioButtonQUseActualMix.Text = "Використання q для заданої
суміші";

radioButtonQUseActualMix.UseVisualStyleBackColor = true;

//

// buttonCalculate

//

buttonCalculate.BackColor = Color.LightGreen;

buttonCalculate.ForeColor = Color.DarkGreen;

buttonCalculate.Location = new Point(16, 471);

buttonCalculate.Margin = new Padding(4, 5, 4, 5);

buttonCalculate.Name = "buttonCalculate";

buttonCalculate.Size = new Size(175, 62);

buttonCalculate.TabIndex = 3;

buttonCalculate.Text = "Розрахувати";

buttonCalculate.UseVisualStyleBackColor = false;

buttonCalculate.Click += buttonCalculate_Click;

//

// groupBoxResults

//

groupBoxResults.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;

groupBoxResults.Controls.Add(textBoxResultM1);

```

```
groupBoxResults.Controls.Add(labelResultM1);
groupBoxResults.Controls.Add(textBoxResultA2);
groupBoxResults.Controls.Add(labelResultA2);
groupBoxResults.Controls.Add(textBoxResultRhoRatio);
groupBoxResults.Controls.Add(labelResultRhoRatio);
groupBoxResults.Controls.Add(textBoxResultA1);
groupBoxResults.Controls.Add(labelResultA1);
groupBoxResults.Controls.Add(textBoxResultU1);
groupBoxResults.Controls.Add(labelResultU1);
groupBoxResults.Controls.Add(textBoxResultRho2);
groupBoxResults.Controls.Add(labelResultRho2);
groupBoxResults.Controls.Add(textBoxResultP2Atm);
groupBoxResults.Controls.Add(labelResultP2Atm);
groupBoxResults.Controls.Add(textBoxResultP2);
groupBoxResults.Controls.Add(labelResultP2);
groupBoxResults.Location = new Point(16, 551);
groupBoxResults.Margin = new Padding(4, 5, 4, 5);
groupBoxResults.Name = "groupBoxResults";
groupBoxResults.Padding = new Padding(4, 5, 4, 5);
groupBoxResults.Size = new Size(988, 208);
groupBoxResults.TabIndex = 5;
groupBoxResults.TabStop = false;

groupBoxResults.Text = "Результати розрахунку (Параметри за  
фронтом детонації)";
```



```
//  
  
// textBoxResultM1  
  
//  
  
textBoxResultM1.BackColor = Color.AliceBlue;  
textBoxResultM1.ForeColor = Color.Tomato;  
textBoxResultM1.Location = new Point(817, 128);  
textBoxResultM1.Margin = new Padding(4, 5, 4, 5);  
textBoxResultM1.Name = "textBoxResultM1";  
textBoxResultM1.ReadOnly = true;  
textBoxResultM1.Size = new Size(119, 27);  
textBoxResultM1.TabIndex = 15;  
textBoxResultM1.TabStop = false;  
  
//  
  
// labelResultM1  
  
//  
  
labelResultM1.AutoSize = true;  
labelResultM1.Location = new Point(553, 164);  
labelResultM1.Margin = new Padding(4, 0, 4, 0);  
labelResultM1.Name = "labelResultM1";  
labelResultM1.Size = new Size(130, 20);  
labelResultM1.TabIndex = 14;  
labelResultM1.Text = "Число Маха (M1):";  
  
//
```

```
// textBoxResultA2

//

textBoxResultA2.BackColor = Color.AliceBlue;

textBoxResultA2.ForeColor = Color.Tomato;

textBoxResultA2.Location = new Point(817, 164);

textBoxResultA2.Margin = new Padding(4, 5, 4, 5);

textBoxResultA2.Name = "textBoxResultA2";

textBoxResultA2.ReadOnly = true;

textBoxResultA2.Size = new Size(119, 27);

textBoxResultA2.TabIndex = 13;

textBoxResultA2.TabStop = false;

//

// labelResultA2

//

labelResultA2.AutoSize = true;

labelResultA2.Location = new Point(553, 126);

labelResultA2.Margin = new Padding(4, 0, 4, 0);

labelResultA2.Name = "labelResultA2";

labelResultA2.Size = new Size(184, 20);

labelResultA2.TabIndex = 12;

labelResultA2.Text = "Швидкість звуку (a2, м/с):";

//

// textBoxResultRhoRatio
```

```
//

textBoxResultRhoRatio.BackColor = Color.AliceBlue;

textBoxResultRhoRatio.ForeColor = Color.Tomato;

textBoxResultRhoRatio.Location = new Point(367, 165);

textBoxResultRhoRatio.Margin = new Padding(4, 5, 4, 5);

textBoxResultRhoRatio.Name = "textBoxResultRhoRatio";

textBoxResultRhoRatio.ReadOnly = true;

textBoxResultRhoRatio.Size = new Size(119, 27);

textBoxResultRhoRatio.TabIndex = 11;

textBoxResultRhoRatio.TabStop = false;

//

// labelResultRhoRatio

//

labelResultRhoRatio.AutoSize = true;

labelResultRhoRatio.Location = new Point(20, 171);

labelResultRhoRatio.Margin = new Padding(4, 0, 4, 0);

labelResultRhoRatio.Name = "labelResultRhoRatio";

labelResultRhoRatio.Size = new Size(189, 20);

labelResultRhoRatio.TabIndex = 10;

labelResultRhoRatio.Text = "Відношення густин  $\rho_2/\rho_1$ ";

//

// textBoxResultA1

//
```

```
textBoxResultA1.BackColor = Color.AliceBlue;

textBoxResultA1.ForeColor = Color.Tomato;

textBoxResultA1.Location = new Point(817, 83);

textBoxResultA1.Margin = new Padding(4, 5, 4, 5);

textBoxResultA1.Name = "textBoxResultA1";

textBoxResultA1.ReadOnly = true;

textBoxResultA1.Size = new Size(119, 27);

textBoxResultA1.TabIndex = 9;

textBoxResultA1.TabStop = false;

//

// labelResultA1

//

labelResultA1.AutoSize = true;

labelResultA1.Location = new Point(553, 86);

labelResultA1.Margin = new Padding(4, 0, 4, 0);

labelResultA1.Name = "labelResultA1";

labelResultA1.Size = new Size(184, 20);

labelResultA1.TabIndex = 8;

labelResultA1.Text = "Швидкість звуку (a1, м/с):";

//

// textBoxResultU1

//

textBoxResultU1.BackColor = Color.AliceBlue;
```

```
textBoxResultU1.ForeColor = Color.Tomato;

textBoxResultU1.Location = new Point(817, 44);

textBoxResultU1.Margin = new Padding(4, 5, 4, 5);

textBoxResultU1.Name = "textBoxResultU1";

textBoxResultU1.ReadOnly = true;

textBoxResultU1.Size = new Size(119, 27);

textBoxResultU1.TabIndex = 5;

textBoxResultU1.TabStop = false;

//

// labelResultU1

//

labelResultU1.AutoSize = true;

labelResultU1.Location = new Point(553, 48);

labelResultU1.Margin = new Padding(4, 0, 4, 0);

labelResultU1.Name = "labelResultU1";

labelResultU1.Size = new Size(213, 20);

labelResultU1.TabIndex = 4;

labelResultU1.Text = "Швидкість детонації (u1, м/с):";

//

// textBoxResultRho2

//

textBoxResultRho2.BackColor = Color.AliceBlue;

textBoxResultRho2.ForeColor = Color.Tomato;
```

```

textBoxResultRho2.Location = new Point(367, 128);

textBoxResultRho2.Margin = new Padding(4, 5, 4, 5);

textBoxResultRho2.Name = "textBoxResultRho2";

textBoxResultRho2.ReadOnly = true;

textBoxResultRho2.Size = new Size(119, 27);

textBoxResultRho2.TabIndex = 3;

textBoxResultRho2.TabStop = false;

//

// labelResultRho2

//

labelResultRho2.AutoSize = true;

labelResultRho2.Location = new Point(20, 131);

labelResultRho2.Margin = new Padding(4, 0, 4, 0);

labelResultRho2.Name = "labelResultRho2";

labelResultRho2.Size = new Size(327, 20);

labelResultRho2.TabIndex = 2;

labelResultRho2.Text = "Щільність вихідної горючої суміші ( $\rho_2$ ,
кг/м³):";

//

// textBoxResultP2Atm

//

textBoxResultP2Atm.BackColor = Color.AliceBlue;

textBoxResultP2Atm.ForeColor = Color.Tomato;

textBoxResultP2Atm.Location = new Point(367, 86);

```

```

textBoxResultP2Atm.Margin = new Padding(4, 5, 4, 5);
textBoxResultP2Atm.Name = "textBoxResultP2Atm";
textBoxResultP2Atm.ReadOnly = true;
textBoxResultP2Atm.Size = new Size(119, 27);
textBoxResultP2Atm.TabIndex = 17;
textBoxResultP2Atm.TabStop = false;

//

// labelResultP2Atm

//

labelResultP2Atm.AutoSize = true;
labelResultP2Atm.Location = new Point(20, 91);
labelResultP2Atm.Margin = new Padding(4, 0, 4, 0);
labelResultP2Atm.Name = "labelResultP2Atm";
labelResultP2Atm.Size = new Size(289, 20);
labelResultP2Atm.TabIndex = 16;
labelResultP2Atm.Text = "Тиск у вихідній горючій суміші (p2,
атм):";

//

// textBoxResultP2

//

textBoxResultP2.BackColor = Color.AliceBlue;
textBoxResultP2.ForeColor = Color.Tomato;
textBoxResultP2.Location = new Point(367, 45);

```

```

textBoxResultP2.Margin = new Padding(4, 5, 4, 5);

textBoxResultP2.Name = "textBoxResultP2";

textBoxResultP2.ReadOnly = true;

textBoxResultP2.Size = new Size(119, 27);

textBoxResultP2.TabIndex = 1;

textBoxResultP2.TabStop = false;

//

// labelResultP2

//

labelResultP2.AutoSize = true;

labelResultP2.Location = new Point(20, 51);

labelResultP2.Margin = new Padding(4, 0, 4, 0);

labelResultP2.Name = "labelResultP2";

labelResultP2.Size = new Size(283, 20);

labelResultP2.TabIndex = 0;

labelResultP2.Text = "Тиск у вихідній горючій суміші (p2, Па):";

//

// labelError

//

labelError.Anchor = AnchorStyles.Top | AnchorStyles.Left |
AnchorStyles.Right;

labelError.ForeColor = Color.Red;

labelError.Location = new Point(16, 777);

labelError.Margin = new Padding(4, 0, 4, 0);

```



```
labelError.Name = "labelError";

labelError.Size = new Size(988, 20);

labelError.TabIndex = 6;

labelError.Text = "Текст помилки";

labelError.Visible = false;

//

// buttonReset

//

buttonReset.BackColor = Color.IndianRed;

buttonReset.ForeColor = Color.White;

buttonReset.Location = new Point(219, 483);

buttonReset.Margin = new Padding(4, 5, 4, 5);

buttonReset.Name = "buttonReset";

buttonReset.Size = new Size(175, 35);

buttonReset.TabIndex = 4;

buttonReset.Text = "Скинути";

buttonReset.UseVisualStyleBackColor = false;

buttonReset.Click += buttonReset_Click;

//

// MainForm

//

AutoScaleDimensions = new.SizeF(8F, 20F);

AutoScaleMode = AutoScaleMode.Font;
```

```
BackColor = Color.LightSteelBlue;

ClientSize = new Size(1020, 815);

Controls.Add(buttonReset);

Controls.Add(labelError);

Controls.Add(groupBoxResults);

Controls.Add(buttonCalculate);

Controls.Add(groupBoxQ);

Controls.Add(groupBoxMixture);

Controls.Add(groupBoxInput);

Margin = new Padding(4, 5, 4, 5);

MinimumSize = new Size(1035, 860);

Name = "MainForm";

Text = "Калькулятор параметрів детонації";

groupBoxInput.ResumeLayout(false);

groupBoxInput.PerformLayout();

groupBoxMixture.ResumeLayout(false);

groupBoxMixture.PerformLayout();

panelVolumes.ResumeLayout(false);

panelVolumes.PerformLayout();

panelMasses.ResumeLayout(false);

panelMasses.PerformLayout();

panelVolumePercent.ResumeLayout(false);

panelVolumePercent.PerformLayout();
```

```

panelMassPercent.ResumeLayout(false);

panelMassPercent.PerformLayout();

groupBoxQ.ResumeLayout(false);

groupBoxQ.PerformLayout();

groupBoxResults.ResumeLayout(false);

groupBoxResults.PerformLayout();

ResumeLayout(false);

}

```

```
#endregion
```

```

private System.Windows.Forms.Label labelGamma1;

private System.Windows.Forms.TextBox textBoxGamma1;

private System.Windows.Forms.Label labelGamma2;

private System.Windows.Forms.TextBox textBoxGamma2;

private System.Windows.Forms.Label labelP1;

private System.Windows.Forms.TextBox textBoxP1;

private System.Windows.Forms.Label labelInitialTemperature; //

```

Новий елемент

```
private System.Windows.Forms.TextBox textBoxInitialTemperature; //
```

Новий елемент

```
private System.Windows.Forms.Label labelRho1;
```

```

private System.Windows.Forms.TextBox textBoxRho1;

private System.Windows.Forms.GroupBox groupBoxInput;

private System.Windows.Forms.GroupBox groupBoxMixture;

private System.Windows.Forms.Panel panelVolumes;

private System.Windows.Forms.TextBox textBoxVolumeOxidizer;

private System.Windows.Forms.Label labelVolumeOxidizer;

private System.Windows.Forms.TextBox textBoxVolumeFuel;

private System.Windows.Forms.Label labelVolumeFuel;

private System.Windows.Forms.Panel panelMasses;

private System.Windows.Forms.TextBox textBoxMassOxidizer;

private System.Windows.Forms.Label labelMassOxidizer;

private System.Windows.Forms.TextBox textBoxMassFuel;

private System.Windows.Forms.Label labelMassFuel;

private System.Windows.Forms.Panel panelVolumePercent;

private System.Windows.Forms.TextBox textBoxVolumePercentFuel;

private System.Windows.Forms.Label labelVolumePercentFuel;

private System.Windows.Forms.Panel panelMassPercent;

private System.Windows.Forms.TextBox textBoxMassPercentFuel;

private System.Windows.Forms.Label labelMassPercentFuel;

private System.Windows.Forms.RadioButton radioButtonMixVolumes;

private System.Windows.Forms.RadioButton radioButtonMixMasses;

private System.Windows.Forms.RadioButton radioButtonMixVolumePercent;

```

```

        private System.Windows.Forms.RadioButton
radioButtonMixMassPercent;

        private System.Windows.Forms.ComboBox comboBoxOxidizer;

        private System.Windows.Forms.Label labelOxidizer;

        private System.Windows.Forms.ComboBox comboBoxFuel;

        private System.Windows.Forms.Label labelFuel;

        private System.Windows.Forms.GroupBox groupBoxQ;

        private System.Windows.Forms.TextBox textBoxQStoichRefValue;

        private System.Windows.Forms.Label labelQStoichRef;

        private System.Windows.Forms.TextBox textBoxQActualMixValue;

        private System.Windows.Forms.Label labelQActualMix;

        private System.Windows.Forms.RadioButton
radioButtonQUseStoichRef;

        private System.Windows.Forms.RadioButton
radioButtonQUseActualMix;

        private System.Windows.Forms.Button buttonCalculate;

        private System.Windows.Forms.GroupBox groupBoxResults;

        private System.Windows.Forms.TextBox textBoxResultM1;

        private System.Windows.Forms.Label labelResultM1;

        private System.Windows.Forms.TextBox textBoxResultA2;

        private System.Windows.Forms.Label labelResultA2;

        private System.Windows.Forms.TextBox textBoxResultRhoRatio;

        private System.Windows.Forms.Label labelResultRhoRatio;

```

```

private System.Windows.Forms.TextBox textBoxResultA1;

private System.Windows.Forms.Label labelResultA1;

private System.Windows.Forms.TextBox textBoxResultU1;

private System.Windows.Forms.Label labelResultU1;

private System.Windows.Forms.TextBox textBoxResultRho2;

private System.Windows.Forms.Label labelResultRho2;

private System.Windows.Forms.TextBox textBoxResultP2;

private System.Windows.Forms.Label labelResultP2;

private System.Windows.Forms.Label labelError;

private System.Windows.Forms.Button buttonReset;

private System.Windows.Forms.TextBox textBoxResultP2Atm;

private System.Windows.Forms.Label labelResultP2Atm;

}

}

```

Класи результатів — HeatReleaseResult.cs, DetonationResult.cs;

Призначення:

Описують структури для збереження результатів розрахунків теплоти згоряння (q) та параметрів детонації. Використовуються для передачі даних між обчислювальними модулями та інтерфейсом.

```

namespace WinFormsApp1;

/// <summary>
/// Допоміжний клас для зберігання розрахункових значень  $q$ 
/// </summary>
public class HeatReleaseResult
{

```

```

        public double Q_ActualMixture_J_kg { get; set; } // q для фактичної
        суміші (Дж/кг)
        public double Q_StoichiometricReference_J_kg { get; set; } // q для
        стехіометричної суміші (Дж/кг)
        public string ErrorMessage { get; set; } = null;
    }

```

Довідковий модуль — GasData.cs;

Призначення: Цей клас забезпечує централізоване зберігання та доступ до фізичних властивостей газів, необхідних для всіх розрахунків.

```
namespace WinFormsApp1;
```

```

/// <summary>
/// Статичний клас для зберігання даних про горючі гази та окислювачі.
/// </summary>
public static class GasData
{
    // Густини за нормальних умов (кг/м^3)
    public const double DensityHydrogen = 0.0899;
    public const double DensityMethane = 0.7168;
    public const double DensityPropane = 1.9659;
    public const double DensityEthane = 1.342;
    public const double DensityButane = 2.703;
    public const double DensityOxygen = 1.429;
    public const double DensityAir = 1.2928; // Наближено

    // Молярні маси (кг/моль)
    public const double MolarMassH2 = 0.002016;
    public const double MolarMassCH4 = 0.01604;
    public const double MolarMassC3H8 = 0.0441;
    public const double MolarMassC2H6 = 0.03007;
    public const double MolarMassC4H10 = 0.05812;
    public const double MolarMassO2 = 0.031998;
    public const double MolarMassN2 = 0.028014; // Для повітря
    public const double MolarMassAir = 0.02897; // Наближено (середня)

    // Універсальна газова стала (Дж/(моль*K))
    public const double UniversalGasConstant = 8.314462;

```

```

    /// <summary>
    /// Standard temperature often used for "normal conditions" in Kelvin (0
degrees Celsius).
    /// </summary>
    public const double StandardTemperatureK = 273.15;

    /// <summary>
    /// Питома теплота згоряння стехіометричної суміші з КИСНЕМ
(МДж/кг суміші)
    /// </summary>
    public static readonly Dictionary<string, double>
StoichiometricQ_Oxygen = new Dictionary<string, double>
    {
        { "Водень", 15.5 }, { "Метан", 10.0 }, { "Пропан", 9.99 }, { "Етан",
10.04 }, { "Бутан", 10.36 }
    };

    /// <summary>
    /// Питома теплота згоряння стехіометричної суміші з ПОВІТРЯМ
(МДж/кг суміші)
    /// </summary>
    public static readonly Dictionary<string, double> StoichiometricQ_Air =
new Dictionary<string, double>
    {
        { "Водень", 3.91 }, { "Метан", 2.72 }, { "Пропан", 2.75 }, { "Етан",
2.76 }, { "Бутан", 2.86 }
    };

    /// <summary>
    /// Масова частка пального в стехіометричній суміші з КИСНЕМ (%)
    /// </summary>
    public static readonly Dictionary<string, double>
StoichiometricMassPercentFuel_Oxygen = new Dictionary<string, double>
    {
        { "Водень", (2 * MolarMassH2) / (2 * MolarMassH2 + MolarMassO2)
* 100.0 },
        { "Метан", MolarMassCH4 / (MolarMassCH4 + 2 * MolarMassO2) *
100.0 },
        { "Пропан", MolarMassC3H8 / (MolarMassC3H8 + 5 *
MolarMassO2) * 100.0 },
        { "Етан", (2 * MolarMassC2H6) / (2 * MolarMassC2H6 + 7 *
MolarMassO2) * 100.0 },
        { "Бутан", (2 * MolarMassC4H10) / (2 * MolarMassC4H10 + 13 *
MolarMassO2) * 100.0 }
    }

```



```

};

/// <summary>
/// Масова частка пального в стехіометричній суміші з ПОВІТРЯМ
(%)
/// </summary>
public static readonly Dictionary<string, double>
StoichiometricMassPercentFuel_Air = new Dictionary<string, double>
{
    { "Водень", (2 * MolarMassH2) / (2 * MolarMassH2 + MolarMassO2
/ 0.23) * 100.0 },
    { "Метан", MolarMassCH4 / (MolarMassCH4 + (2 * MolarMassO2) /
0.23) * 100.0 },
    { "Пропан", MolarMassC3H8 / (MolarMassC3H8 + (5 *
MolarMassO2) / 0.23) * 100.0 },
    { "Етан", (2 * MolarMassC2H6) / ((2 * MolarMassC2H6) + (7 *
MolarMassO2) / 0.23) * 100.0 },
    { "Бутан", (2 * MolarMassC4H10) / ((2 * MolarMassC4H10) + (13 *
MolarMassO2) / 0.23) * 100.0 }
};

/// <summary>
/// Повертає густину газу за назвою.
/// </summary>
public static double GetDensity(string gasName)
{
    switch (gasName)
    {
        case "Водень": return DensityHydrogen;
        case "Метан": return DensityMethane;
        case "Пропан": return DensityPropane;
        case "Етан": return DensityEthane;
        case "Бутан": return DensityButane;
        case "Кисень": return DensityOxygen;
        case "Повітря": return DensityAir;
        default: throw new ArgumentException("Невідомий газ: " +
gasName);
    }
}

/// <summary>
/// Повертає молярну масу газу за назвою.
/// </summary>
public static double GetMolarMass(string gasName)

```

```

{
    switch (gasName)
    {
        case "Водень": return MolarMassH2;
        case "Метан": return MolarMassCH4;
        case "Пропан": return MolarMassC3H8;
        case "Етан": return MolarMassC2H6;
        case "Бутан": return MolarMassC4H10;
        case "Кисень": return MolarMassO2;
        case "Повітря": return MolarMassAir;
        default: throw new ArgumentException("Невідомий газ для
молярної маси: " + gasName);
    }
}

/// <summary>
/// Calculates the density of a gas mixture based on mass fractions,
pressure, and temperature.
///  $\rho = P / (R_{\text{mix}} * T)$  where  $R_{\text{mix}} = R_{\text{universal}} / M_{\text{mix}}$ .
///  $M_{\text{mix}} = 1 / ( (\text{mass\_frac\_fuel} / M_{\text{fuel}}) + (\text{mass\_frac\_oxidizer} / M_{\text{oxidizer}}) )$ 
/// </summary>
/// <param name="fuelName">Name of the fuel gas.</param>
/// <param name="oxidizerName">Name of the oxidizer gas.</param>
/// <param name="fuelMassFraction">Mass fraction of the fuel (e.g., 0.1
for 10%).</param>
/// <param name="pressurePa">Pressure of the mixture in
Pascals.</param>
/// <param name="temperatureK">Temperature of the mixture in
Kelvin.</param>
/// <returns>Density of the mixture in kg/m3, or double.NaN if an error
occurs.</returns>
public static double CalculateMixtureDensity(
    string fuelName,
    string oxidizerName,
    double fuelMassFraction,
    double pressurePa,
    double temperatureK)
{
    if (fuelMassFraction < 0.0 || fuelMassFraction > 1.0)
    {
        return double.NaN; // Invalid mass fraction
    }
    if (pressurePa <= 0 || temperatureK <= 0)

```

```

    {
        return double.NaN; // Invalid pressure or temperature
    }

    try
    {
        double molarMassFuel = GetMolarMass(fuelName);
        double molarMassOxidizer = GetMolarMass(oxidizerName);
        double oxidizerMassFraction = 1.0 - fuelMassFraction;

        if (molarMassFuel == 0 || molarMassOxidizer == 0) return
double.NaN;

        // Effective molar mass of the mixture
        double effectiveMolarMassMixtureDenom = (fuelMassFraction /
molarMassFuel) + (oxidizerMassFraction / molarMassOxidizer);
        if (effectiveMolarMassMixtureDenom == 0) return double.NaN;
        double effectiveMolarMassMixture = 1.0 /
effectiveMolarMassMixtureDenom;

        if (double.IsNaN(effectiveMolarMassMixture) ||
effectiveMolarMassMixture <= 0 || double.IsInfinity(effectiveMolarMassMixture))
        {
            return double.NaN;
        }

        // Specific gas constant for the mixture
        double specificGasConstantMixture = UniversalGasConstant /
effectiveMolarMassMixture;

        if (double.IsNaN(specificGasConstantMixture) ||
specificGasConstantMixture <= 0 ||
double.IsInfinity(specificGasConstantMixture))
        {
            return double.NaN;
        }

        // Density using ideal gas law: rho = P / (R_mix * T)
        double density = pressurePa / (specificGasConstantMixture *
temperatureK);
        return density;
    }
    catch (ArgumentException)

```

```

    {
        // Unknown gas name from GetMolarMass
        return double.NaN;
    }
}

```

Клас результатів детонації — DetonationResult.cs;

Призначення:

Клас DetonationResult описує набір параметрів, які повертаються після розрахунку детонації за допомогою моделі Чепмена–Жуге. Цей клас є контейнером даних, який забезпечує структурований обмін результатами між розрахунковим модулем та інтерфейсом користувача.

```
namespace WinFormsApp1;
```

```

/// <summary>
/// Результати розрахунку параметрів детонації.
/// </summary>
public class DetonationResult
{
    public double Pressure { get; set; } // p2, Па
    public double Density { get; set; } // rho2, кг/м^3
    public double Velocity { get; set; } // u1, м/с
    public double A_Value { get; set; } // Розрахована величина A
    public double a1_Velocity { get; set; } // Швидкість звуку a1, м/с
    public double DensityRatio { get; set; } // Відношення rho2/rho1
    public double a2_Velocity { get; set; } // Швидкість звуку a2, м/с
    public double MachNumber { get; set; } // Число Маха M1 = u1/a1
    public string ErrorMessage { get; set; } = null;
}

```