

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут аерокосмічних технологій
Кафедра авіа- та ракетобудування**

«На правах рукопису»
УДК 629.7.01

До захисту допущено:
В. о. завідувача кафедри
_____ Володимир КАБАНЯЧИЙ
«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Літаки і вертольоти»

зі спеціальності 134 «Авіаційна та ракетно-космічна техніка»

**на тему: «Програмний комплекс для розрахунку аеродинамічних
характеристик літака»**

Виконав (-ла):

студент (-ка) VI курсу, групи АЛ-91мп
Смаглій Антон Сергійович _____

Керівник:

Доцент, к.т.н.
Бондаренко Олександр Миколайович _____

Рецензент:

Професор, к.т.н., професор кафедри
Пономаренко Сергій Олексійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент (-ка) _____

Київ – 2020 року

- 4.2 Вибір методу для програмування.
- 4.3 Формування моделі літака для розрахунку аеродинамічних характеристик і вимоги до вхідних та вихідних даних.
- 4.4 Розробка інтерфейсу програмного комплексу.
- 4.5 Розроблення та тестування програмного модулю для розрахунків.
- 4.6 Аналіз аеродинамічних характеристик літака в розробленому програмному модулі.
- 4.7 Розроблення документації програми.
- 4.8 Розроблення конструкції літака.
- 4.9 Оформлення пояснювальної записки.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

- 5.1. Алгоритм програми – 1 арк. ф. А1.
- 5.2. Конструкція літака – 1 арк. ф. А1.
- 5.3. Інтерфейс та результати розрахунку – 1 арк. ф. А1
- 5.4. Опис стартап-проекту – 1 арк. ф. А1
- 5.5. Аналіз варіантів побудови програмного комплексу – 1 арк.ф. А1

6. Дата видачі завдання « 01 » 09 2020 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз сучасних методів розрахунку аеродинамічних характеристик.	01.09.2020 р.	
2	Вибір методу для програмування.	до 15.09.2020 р.	
3	Формування моделі літака для розрахунку аеродинамічних характеристик і вимоги до вхідних та вихідних даних.	до 29.09.2020 р.	
4	Розробка інтерфейсу програмного комплексу.	до 13.10.2020 р.	
5	Розроблення та тестування програмного модулю для розрахунків.	до 27.10.2020 р.	
6	Аналіз аеродинамічних характеристик літака в розробленому програмному модулі.	до 10.11.2020р	
7	Розроблення документації програми.	до 24.11.2020 р.	
8	Розроблення конструкції літака.	до 01.12.2020 р.	
9	Оформлення пояснювальної записки.	до 10.12.2020 р.	

Студент

_____ (підпис)

_____ Смаглій А.С. (ініціали, прізвище)

Керівник роботи

_____ (підпис)

_____ Бондаренко О.М. (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація:

АЕРОДИНАМІЧНІ ХАРАКТЕРИСТИКИ, КОЕФІЦІЄНТИ, ПІДЙОМНА СИЛА, C#, ОБ'ЄКТНО-ОРІЄНТОВНЕ ПРОГРАМУВАННЯ.

Актуальність роботи. Розрахунок аеродинамічних характеристик літака є дуже важливою частиною проектування літаку. Літальні апарати та програмне забезпечення для розрахунку аеродинамічних характеристик щороку стають більш популярні, оскільки попит на літаки росте. Актуальність роботи полягає в прискоренні розрахунку аеродинамічних характеристик літальних апаратів та підвищенні їх точності.

Мета і завдання дослідження. Мета дослідження - розробка програмного інтерфейсу, який реалізує метод визначення аеродинамічних характеристик ЛА. Комплекс зорієнтований на розрахунок параметрів легких літальних апаратів. Для досягнення мети було вирішено наступні задачі. Було проаналізовано сучасні методи визначення аеродинамічних характеристик літальних апаратів. Розглянуто метод вихорів, метод кінцевих елементів, теоретичні можливості розрахунку із використанням рівняння рівняння Бернуллі та формули Жуковського, а також експериментальний метод перебудови поляр. Обраний був останній метод.

Об'єкт дослідження – методи розрахунку аеродинамічних характеристик, такі як:

- Метод кінцевих елементів,
- Метод дискретних вихорів,
- Рівняння Бернуллі, т.д.

Методи дослідження – аналіз отриманих результатів в сфері розрахунку аеродинамічних характеристик, порівняння переваг та недоліків програми.

Наукова новизна отриманих результатів. Створено вузьконаправлений програмний комплекс, який не містить лишніх деталей та функцій.

Практичне значення одержаних результатів. Запропоновано використовувати розроблений програмний комплекс, що дає змогу більш точно та швидше знаходити аеродинамічні характеристики літальних апаратів.

ABSTRACT

Master's thesis:

AERODYNAMIC CHARACTERISTICS, COEFFICIENTS, LIFTING FORCE, C #, OBJECT-ORIENTED PROGRAMMING.

Relevance of work. Calculating the aerodynamic characteristics of an aircraft is a very important part of aircraft design. Aircraft and software for calculating aerodynamic characteristics are becoming more popular every year as the demand for aircraft grows. The urgency of the work is to accelerate the calculation of aerodynamic characteristics of aircraft and increase their accuracy.

The purpose and objectives of the study. The purpose of the study is to develop a software interface that implements the method of determining the aerodynamic characteristics of the aircraft. The complex is focused on calculating the parameters of light aircraft. To achieve this goal, the following tasks were solved. Modern methods for determining the aerodynamic characteristics of aircraft were analyzed. The vortex method, the finite element method, the theoretical possibilities of calculation using the Bernoulli equation and the Zhukovsky formula, as well as the experimental method of polar rearrangement are considered. The last method was chosen.

The object of study - methods for calculating aerodynamic characteristics, such as:

- Finite element method,
- The method of discrete vortices,
- Bernoulli's equation, etc.

Research methods - analysis of the obtained results in the field of calculation of aerodynamic characteristics, comparison of advantages and disadvantages of the program.

Scientific novelty of the obtained results. A narrowly focused software package has been created, which does not contain unnecessary details and functions.

The practical significance of the obtained results. It is proposed to use the developed software package, which allows to more accurately and quickly find the aerodynamic characteristics of aircraft.

ЗМІСТ

ВСТУП	9
1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРАХУНКУ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК	10
1.1 Метод скінчено-елементної моделі.....	10
1.2 Метод дискретних вихорів.	13
1.3 Рівняння Бернуллі.	17
Висновок до розділу.....	21
2. ВИБІР МЕТОДУ ПРОГРАМУВАННЯ	22
2.1. Simulink.....	22
2.2 Femap	29
Висновки до розділу	32
3. ФОРМУВАННЯ МОДЕЛІ ЛІТАКА ДЛЯ РОЗРАХУНКУ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК І ВИМОГИ ДО ВХІДНИХ ТА ВИХІДНИХ ДАНИХ.	34
Висновки до розділу	39
4. РОЗРОБКА ІНТЕРФЕЙСУ ПРОГРАМНОГО КОМПЛЕКСУ	40
Висновок до розділу.....	44
5. РОЗРОБЛЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО МОДУЛЮ ДЛЯ РОЗРАХУНКІВ.	45
Висновок до розділу.....	55
6. СТАРТАП-ПРОЕКТ	56
Висновок до розділу.....	65

АЛ91МП26.14.00.00.00 ПЗ				
<i>Изм.</i>	<i>Лист</i>	<i>№ докум</i>	<i>Подпись</i>	<i>Дата</i>
<i>Разраб</i>		Смаглій А.С.		
<i>Пров</i>		Кабанячий		
<i>Н. Контр.</i>		Поваров С.А.		
<i>Утв</i>		Сухов В.В.		
Програмний комплекс для розрахунку аеродинамічних характеристик літака			<i>Литера</i>	<i>Лист</i>
			у	8
			КПІ. Ім Ігоря Сікорського 8	
			Кафедра АРБ АЛ-91мп	

ВСТУП

Одним з найважливіших завдань, що вирішують в процесі проектування літаків – є завдання з вибору програмного середовища для виконання розрахунків аеродинамічних характеристик. Існують різні методи розрахунку аеродинамічних характеристик. Серед них найвідоміші такі як метод скінчено-елементної моделі, метод кінцевих вихорів, теоретично можливий метод розрахунку за допомогою рівняння Бернуллі, експериментальний метод перебудови поляр.

У кожного з цих методів є свої переваги та недоліки. Метод скінчено-елементної моделі – є найбільш ефективним методом вирішення завдання визначення аеродинамічних характеристик з високою точністю, але, даний метод потребує великий об'єм оперативної пам'яті на носії програми, адже операції по знаходженню цих коефіцієнтів є дуже об'ємними.

Метод скінчених вихорів – також дуже точний у знаходженні коефіцієнтів, але, на жаль, він є дуже складним до реалізації у програмі.

Метою дипломного проекту було обрано розробку системи розрахунку аеродинамічних характеристик та визначення аеродинамічних характеристик та параметрів літака.

1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ РОЗРАХУНКУ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК

Згідно з технічним завданням необхідно розробити систему розрахунку аеродинамічних характеристик. У даному розділі буде проведено аналіз сучасних методів розрахунку аеродинамічних характеристик, та обраний такий, який задовольняє всі вимоги.

1.1 Метод скінчено-елементної моделі.

Метод скінченних елементів (МСЕ) — числова техніка знаходження розв'язків інтегральних та диференціальних рівнянь у частинних похідних (ДРЧП). Процес розв'язання побудований або на повному усуненні диференціального рівняння для стаціонарних задач, або на розкладі ДРЧП в апроксимуючу систему звичайних диференціальних рівнянь, які потім розв'язуються використанням якої-небудь стандартної техніки, такої як метод Ейлера, Рунге-Кутти тощо.

В даному розділі приводяться вихідні положення МСЕ а також описуються основні задачі з розрахунку конструкцій. Співвідношення МСЕ даються в різних формах, таких як координатна, векторна та матрична.

МСЕ в останні десятиліття отримав дуже широке розповсюдження і став одним з основних методів розрахунку конструкцій. Це обумовлено універсальністю підходу, який лежить в основу МСЕ, який заключається у представленні геометрії будь якого тіла у вигляді сукупності елементів простої форми, таких як трикутники, чотирикутники та інші (рис. 1.1.1).

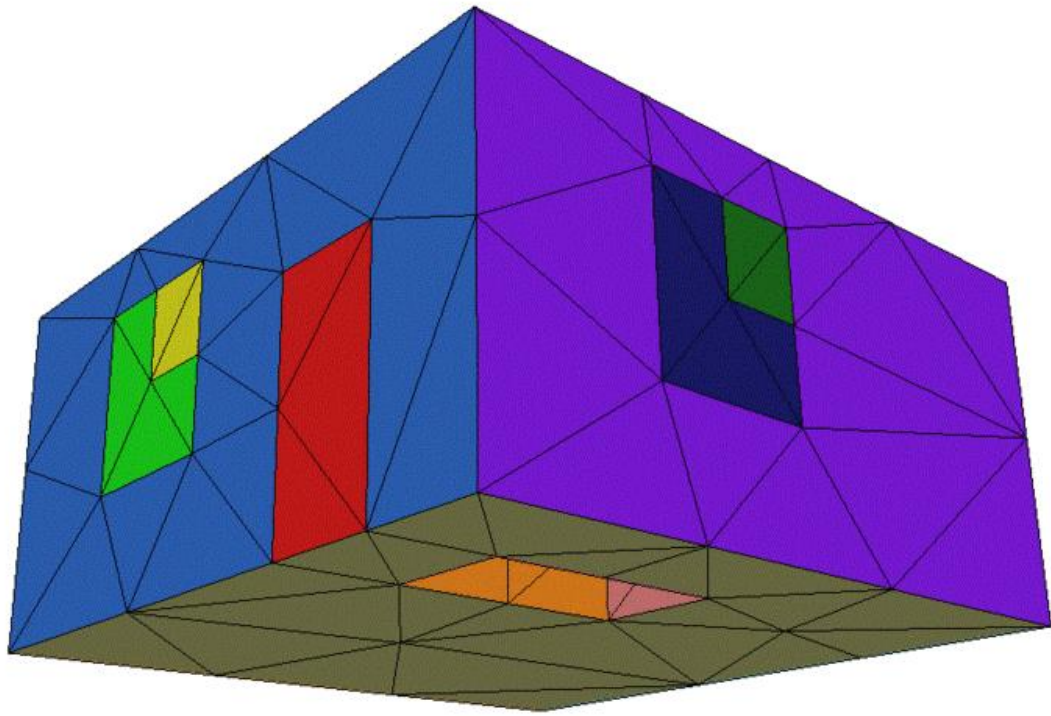


Рис. 1.1.1. Представлення геометрії тіла у вигляді елементів простої форми.

Основна ідея МСЕ складається в тому, що будь-яку неперервну в якій області величину можна апроксимувати дискретною моделлю, яка складається з із множини кусочно-неперервних функцій, які визначені в кінцевому числі подобластей(елементів). Зазвичай такими функціями являються поліноми – лінійні, квадратні, кубічні і т.д. Кусочно-неперервні функції будуються за допомогою значення неперервної величини в точках з'єднання елементів.

Таким чином, щоб визначити невідому неперервну величину, потрібно знайти її значення у вузлах.

Основні етапи створення дискретної моделі невідомої величини наступні:

- У досліджуваній області задається кінцеве число точок.
- Значення неперервної величини в кожній точці вважаються невідомими, але вони мають бути знайдені
- Досліджувана область розбивається на кінцеве число подобластей(елементів), які мають спільні точки.

- Неперервна величина в кожному елементі апроксимується поліномом, який визначає за допомогою вузлових значень цієї величини: для кожного елемента визначається свій поліном, но його коефіцієнти підбираються так, щоб зберіглась неперервність величина на кожній границі елемента.

Елементи бувають одновимірні(рис. 1.1.2.), плоскі(рис. 1.1.3.) і просторові(рис. 1.1.4.), з прямолінійними або криволінійними сторонами. Вздовж них може бути два або більшу вузлів. У всіх вузлах задаються узагальнені координати, які називають вузловим зміщенням, сувокупність яких записуються матрицею

$$\{\lambda\} = \{\lambda_1, \lambda_2, \dots, \lambda_N\}^T,$$



Рис. 1.1.2. Одновимірний елемент з двома вузлами.

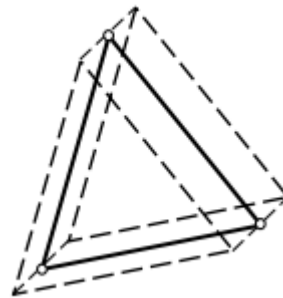


Рис. 1.1.3. Плоский елемент.

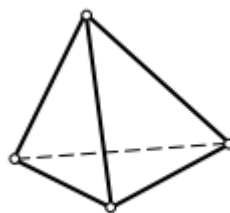


Рис. 1.1.4. Просторовий елемент.

Одна з основних задач МСЕ розрахунок конструкцій на міцність при статичній нарузці. Із вирішення задачі визначають вектор вузлових зміщень і по співвідношенням можна знайти переміщення точок або деформації та напруження.

Основними перевагами МСЕ є:

- Можливість досліджувати тіла (конструкції), складені з декількох матеріалів (так як властивості матеріалів сусідніх елементів можуть бути різними),
- Можливість досліджувати області (конструкції) будь-якої форми (Так як криволінійна область апроксимується прямолінійними елементами або точно описується криволінійними елементами).
- Можливість обліку різних граничних умов: з розривної навантаженням, змішаних.
- Можливість складання загальних методик і програм для вирішення різних з фізики завдань одного певного виду.

1.2 Метод дискретних вихорів.

На етапі концептуального аеродинамічного проектування стоїть завдання оцінки основних аеродинамічних коефіцієнтів. При цьому достатньо, щоб ці залежності відтворювалися якісно з точністю на рівні 10-15%. Якщо несуча поверхня ЛА не дуже товста, то, як випливає з теорії, її підйомна сила залежить практично повністю від кривизни профілю, а його товщину можна ігнорувати.

$$C_y = 2\pi(1 + 0,77c)\sin(\alpha + 2f),$$

де c – відносна товщина профілю, f – відносна кривизна профілю, віднесені до хорди b . Таким чином, підйомна сила може повністю моделюватись системою приєднаних вихорів, які розташовані на безкінечно

тонкій серединній поверхні (рис. 1.2.1), а не на самій поверхні (рис. 1.2.2), при цьому її підйомна сила може буде вирахована по формулі, яка представлена вище.

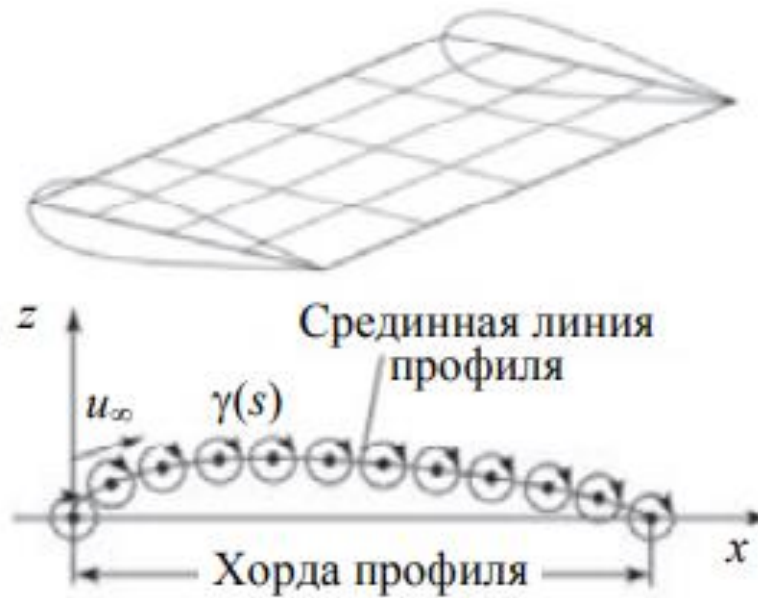


Рис. 1.2.1. Моделювання товстого крила за допомогою системи вихрів, розташованих на тонкій серединній поверхні.

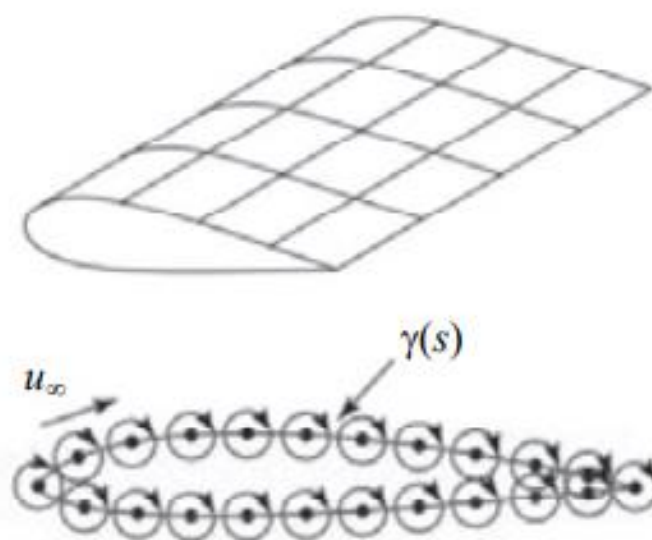


Рис. 1.2.2. Моделювання товстого крила за допомогою системи вихорів, розташованих на його поверхні.

У загальному випадку несуча поверхня має кінцеву товщину і обтікається в'язким потоком, що призводить до утворення прикордонного шару (рис. 1.2.3). Прикордонний шар можна замінити системою дискретних вихорів γ , розподілених по поверхні s , з завихренням, що дорівнює відповідній завихреності прикордонного шару ω (рис. 1.2.4). Кривизну серединної поверхні можна моделювати змінною інтенсивністю вихорів, розташованих уздовж хорди. Таким чином, для тонких тривимірних поверхонь, складених з профілів, що мають невелику товщину і кривизну, допустима їх заміна плоскими поверхнями. При цьому підйомна сила і вихровий опір будуть моделюватися досить точно.

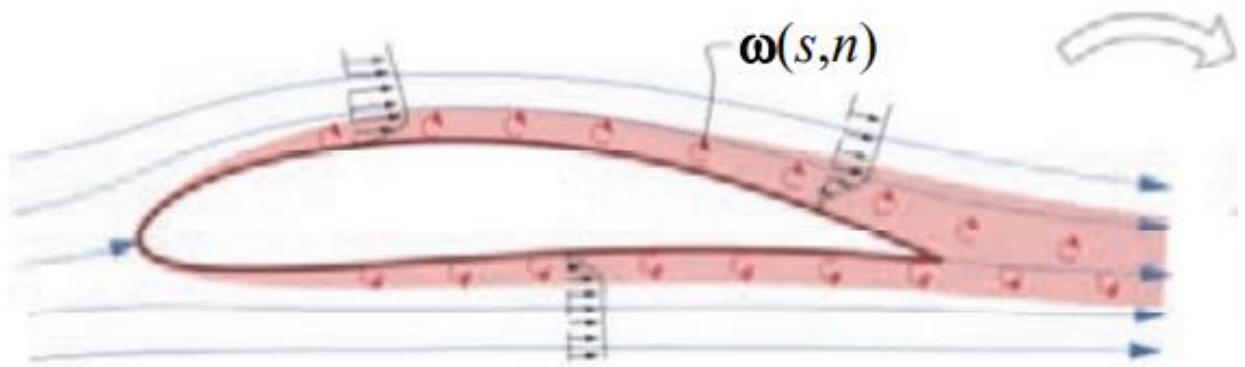


Рис. 1.2.3. Імітація реального профілю крила кінцевої товщини, обтікаємого в'язким потоком з пограничним слоєм з локальною завихреністю $\omega(s, n)$.

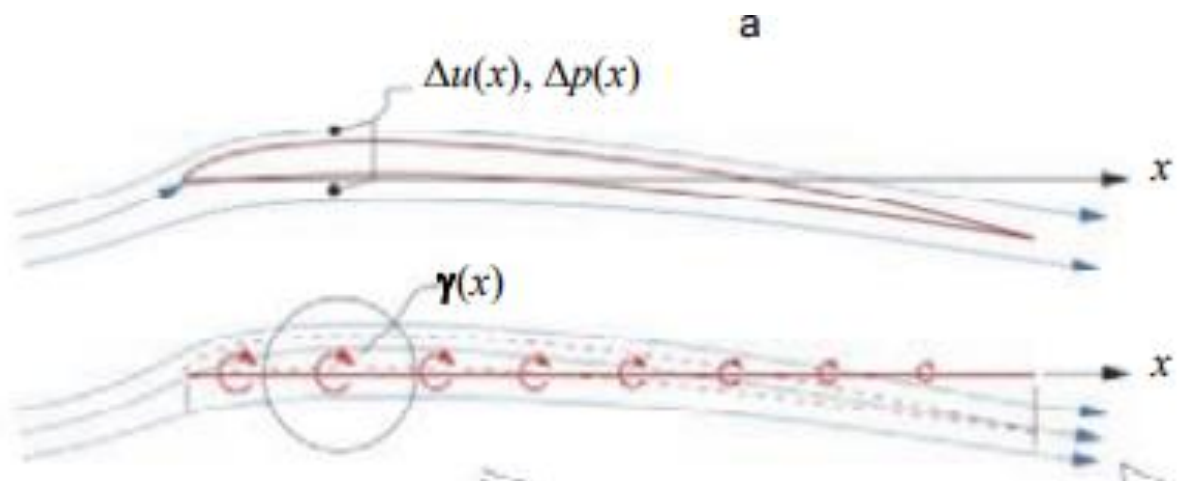


Рис. 1.2.4. Імітація реального профілю крила кінцевої товщини, обтікаємого системою дискретних вихрів $\gamma(s)$.

Описаний метод вихрових ґраток VLM, розроблений Фелкнером, дозволяє розраховувати плоскі тонкі крила і може бути використаний з урахуванням ідей, для оцінки обтікання тонких крил з невеликою кривизною. З деякою натяжкою можна враховувати геометричну крутку крила, задаючи різні кути атаки в різних перетинах крила.

В реальності несуча система ЛА завжди тривимірна. Крило має аеродинамічну і геометричну крутку, тобто в різних перетинах крило може мати різні профілі, встановлені під різними кутами атаки. ЛА має об'ємний фюзеляж, який не можна вважати тонким, хвостове оперення, механізацію крила, яка може відхилятися в польоті. Консолі крила часто мають відігнуті вгору вінглети, самі вони зазвичай встановлюються під деяким кутом до осі фюзеляжу, тобто кінцеві перетини можуть бути розташовані істотно вище або нижче кінцевих перетинів. Все це необхідно якось враховувати на стадії параметричного аналізу. Наступним за методом Фелкнера наближенням є ідея розташувати підковоподібні вихори на поверхні або серединної поверхні крила в кілька рядів (рис. 1.2.5). Очевидно, що це дозволяє врахувати кривизну крила (рис. 1.2.5, а) і моделювати тривимірну поверхню сходять з його задньої кромки вихровий пелени (рис. 1.2.5, б).

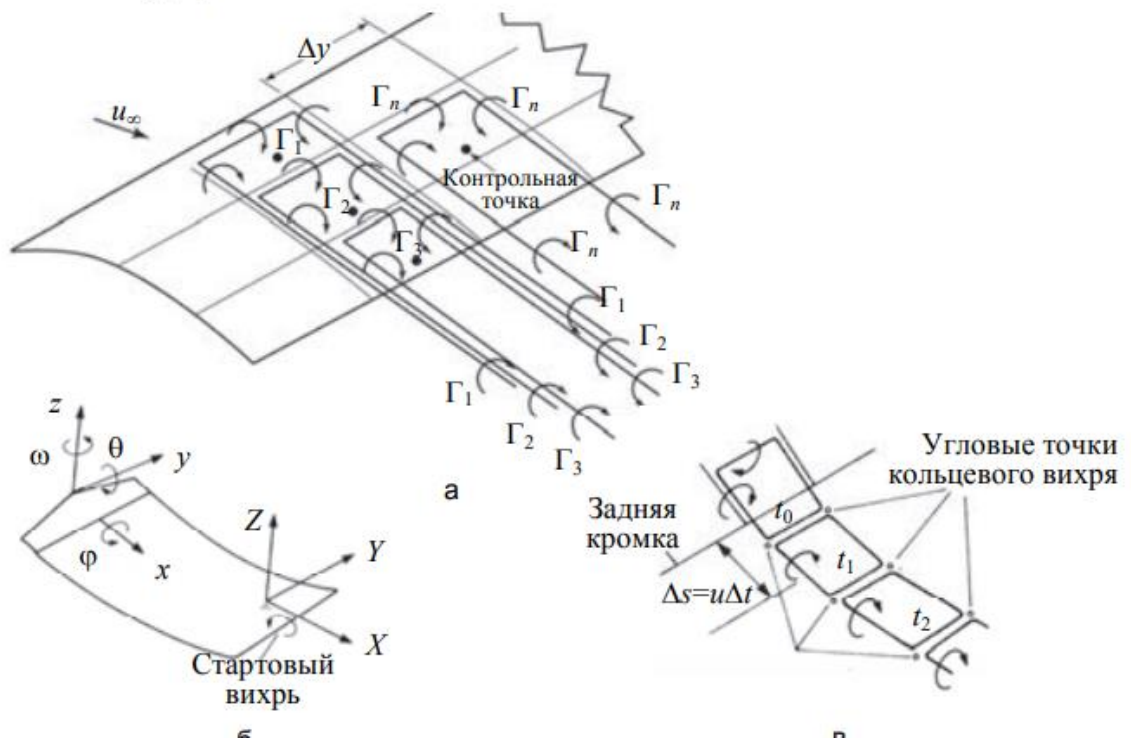


Рис. 1.2.5. Формулювання методу дискретних вихорів, що дозволяє врахувати кривизну несучої поверхні (а) за рахунок розміщення на її серединній поверхні П-образних приєднаних вихорів з циркуляцією Γ_i , де $i = 1..n$, для нестационарного випадку (б) заміна пелени замкнутими вихорами з розміром сторони $\Delta s = u \Delta t$, де Δt - крок за часом, дозволяє врахувати обертання по осях x, y, z і тривимірну форму вихровий пелени (в).

Вихрова пелена може моделюватися замкнутими вихорами (рис. 1.2.5, в), а не напівнескінченної, як показано на рис. 1.2.5, а. Дотична до поверхні пелени паралельна місцевим векторам швидкості на задній кромці. Для врахування впливу пелени на обтікання крила необхідно вводити час t , стартовий вихор і довжину завіси. Зазвичай достатньо довжини в 500 хорд крила.

1.3 Рівняння Бернуллі.

Закон Бернуллі – випливає з закону збереження енергії для потоку ідеальної та нестисливої стаціонарної рідини.

У аеродинаміці повітря розглядається як нестислива рідина. Таке середовище не змінює свою щільність зі зміною тиску. А стаціонарним вважається потік, в якому частки переміщуються по незмінним в часі траєкторіях. Ці траєкторії називаються лініями потоку. У таких потоках не утворюються вихори.

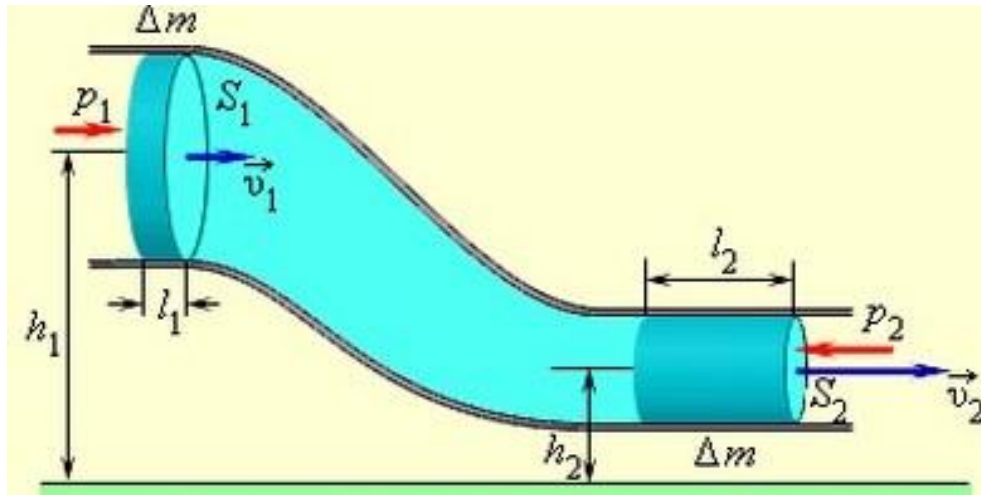


Рис. 1.3.1. Рівняння нерозривності потоку.

Якщо рідина тече по трубі, що має різний поперечний переріз, то тиск в різних місцях труби буде неоднакове.

Подумки виділимо в трубі кілька перетинів, позначивши їх площі S_1 та S_2 . Відповідно, v_1 і v_2 - швидкості течії нестисливої рідини через ці перетину. За час Δt через перетину протекуть рідини, обсяги яких будуть рівні:

$$\Delta V_1 = l_1 * S_1 = v_1 * \Delta t * S_1;$$

$$\Delta V_2 = l_2 * S_2 = v_2 * \Delta t * S_2;$$

Так як ми розглядаємо стаціонарне протягом нестисливої рідини, то згідно із законом збереження маси через будь-який поперечний переріз труби за однаковий проміжок часу проходить однаковий обсяг рідини. Отже, $\Delta V_1 = \Delta V_2$.

$$v_1 * \Delta t * S_1 = v_2 * \Delta t * S_2$$

$$\text{тобто, } v_1 * S_1 = v_2 * S_2$$

$$\text{або } v_1 \cdot \Delta S = \text{const.}$$

Добуток площі поперечного перерізу потоку на його швидкість є незмінною. Це рівняння називають рівнянням нерозривності струменя.

Об'єднавши умова нерозривності рідини і закон збереження енергії, Бернуллі вивів рівняння, згідно з яким з збільшення швидкості потоку зменшується тиск, і навпаки.

З рівняння нерозривності випливає, що $v_1/v_2 = S_2/S_1$. Тобто, швидкості рідин обернено пропорційні площам перетинів. І чим менша площа перетину, тим більша швидкість рідини, що протікає через нього, і навпаки. Рідина, яка надходить з широкою в більш вузьку частину труби, прискорюється. Це означає, що на неї діє сила з боку рідини, що знаходиться в більш широкій частині труби. Для горизонтальної труби причина виникнення цієї сили - різниця тисків в широкому і вузькому ділянках труби. У широкій частині тиск вище, ніж у вузькій, а швидкість нижче. Звідси випливає висновок: «При стаціонарному перебігу рідини тиск більше в тих місцях, де менше швидкість течії, і навпаки».

Рівняння Бернуллі має вигляд:

$$\frac{\rho v^2}{2} + \rho gh + p = \text{const, де}$$

ρ – густина рідини,

v – швидкість потоку,

h – висота, на якій розташовується елемент рідини,

g – прискорення вільного падіння,

p – тиск у точці простору, в якій розташований елемент маси елемента рідини.

Перший доданок рівняння Бернуллі - кінетична енергія потоку, або динамічний тиск. Його створює рух рідини або газу. В авіації його також

називають швидкісним напором. Другий доданок - потенційна енергія, або гідростатичний тиск. Воно створюється вагою стовпа рідини або газу висотою h . І, нарешті, третій доданок, P - це статистичне тиск, який чинять один на одного сусідні шари рідини або газу.

Сума всіх доданків рівняння називається повним тиском.

Для труби, розташованої горизонтально, або горизонтального повітряного потоку рівняння Бернуллі виглядає так:

$$\frac{\rho v^2}{2} + p = const$$

З нього видно, що чим вище швидкість течії рідини (а в аеродинаміці - швидкість повітряного потоку), тим менше тиск, і навпаки.

За допомогою цього закону дуже просто пояснити, як виникає підйомна сила для літального апарату важче повітря. Під час польоту крило літака розрізає повітряний потік на дві частини. Одна частина обтікає верхню поверхню крила, а інша нижню. Форма крила така, що верхній потік повинен подолати більший шлях для того, щоб з'єднатися з нижнім в одній точці. Значить, він рухається з більшою швидкістю. А раз швидкість більше, то і тиск над верхньою поверхнею крила менше, ніж під нижньою. За рахунок різниці цих тисків і виникає підйомна сила крила.

Під час набору літаком висоти зростає різниця тисків, а значить, збільшується і підйомна сила, що дозволяє літаку підніматися вгору.

Відразу зробимо уточнення, що вищеописані закони діють, якщо швидкість руху повітряного потоку не перевищує швидкість звуку (до 340 м / с). Адже ми розглядали повітря як нестисливої рідини. Але виявляється, що при швидкостях вище швидкості звуку повітряний потік поводить по-іншому. Стискальність повітря нехтувати вже не можна. І повітря в цих умовах, як будь-який газ, намагається розширитися і зайняти більший обсяг. З'являються значні перепади тиску або ударні хвилі. А сам повітряний потік

не звужується, а, навпаки, розширюється. Рішенням завдань про рух повітряних потоків зі швидкостями, близькими або перевищують швидкість звуку, займається газова динаміка, що виникла як продовження аеродинаміки.

Використовуючи аеродинамічні закони, теоретична аеродинаміка дозволяє зробити розрахунки аеродинамічних сил, що діють на літальний апарат. А правильність цих розрахунків перевіряють, відчуваючи побудовану модель на спеціальних експериментальних установках, які називаються аеродинамічними трубами. Ці установки дозволяють виміряти величину сил спеціальними приладами.

Крім дослідження сил, що діють на аеродинамічні моделі, за допомогою аеродинамічних вимірювань вивчають розподіл значень швидкості, щільності і температури повітря, що обтікає модель.

Висновок до розділу

В даному розділі було проаналізовано сучасні методи для розрахунку аеродинамічних характеристик, та було встановлено що всі вони мають як свої переваги так і недоліки. Використовуючи ці методи було створено програмний додаток для обрахунку аеродинамічних коефіцієнтів.

2. ВИБІР МЕТОДУ ПРОГРАМУВАННЯ.

2.1. Simulink.

Simulink - середовище динамічного міждисциплінарного моделювання складних технічних систем і основний інструмент для модельно-орієнтованого проектування. Його основним інтерфейсом є графічний інструмент для побудови діаграм і Настроюваний набір бібліотек блоків. Він пропонує тісну інтеграцію з рештою середовищем MATLAB і може або використовувати MATLAB, або створювати сценарії з нього. Simulink широко використовується в автоматичного управління і цифрової обробки сигналів для багатодоменному моделювання та проектування на основі моделей.

У поєднанні з іншими своїми продуктами Simulink може автоматично генерувати вихідний код на мові C для реалізації систем в режимі реального часу. Оскільки ефективність і гнучкість коду поліпшуються, він стає все більш широко поширеним для виробничих систем, на додаток до того, що є інструментом для розробки вбудованих систем через його гнучкості і здатності до швидкої ітерації, Вбудований кодер створює код, досить ефективний для використання у вбудованих системах.

Simulink призначений для моделювання та симуляцій на системному рівні, що дозволяє проводити всебічне дослідження розробляється в єдиному середовищі проектування.

Моделювання та симуляції дозволяють провести перевірку поведінки системи в критичних умовах або аварійних сценаріях. Тим самим відбувається зниження витрат на дорогі фізичні прототипи. Перевірка системи здійснюється за допомогою напівнатурного моделювання і швидкого прототипування.

Моделі Simulink підтримують автоматичну генерацію коду промислового якості на мовах C, C ++ і HDL. Результати роботи

згенерованого коду і моделі ідентичні. Наступним кроком є розгортання коду на цільовому обчислювачі або FPGA / ASIC.

Алгоритми, створені в MATLAB, не потрібно переробляти для повторного використання в Simulink. Код MATLAB додається до блоку Simulink або діаграму Stateflow без змін.

Так само MATLAB використовується для створення наборів вхідних даних для симуляцій систем. Симуляції масштабуються на кластери для прискорення таких інженерних завдань як перебір параметрів або оптимізація, а потім їх результати аналізуються і візуалізуються в MATLAB.

Модельно-орієнтоване проектування (МОП) - це особлива технологія проектування вбудованих систем. Таке проектування виконується певними методами, заснованими на систематичному використанні моделей, які розвиваються, уточнюються і ефективно використовуються на всіх етапах життєвого циклу виробу. Основною метою застосування МОП є гарантія того, що при розробці системи в динаміці враховано взаємний вплив компонентів системи, для системи вибрано оптимально для виконання поставлених завдань, випробування проведені у всіх можливих режимах експлуатації виробу, і компанія розробник вкладалася в бюджет при досягненні потрібної якості виробу в прийнятні терміни. Методологія добре зарекомендувала себе при розробці різних складних систем, зокрема вбудованих систем управління, цифрової обробки сигналів, комп'ютерного зору, радіолокації і зв'язку.

Створення будь-якої системи починається з опису компонентів і визначення зв'язків між ними. Однак продумування архітектури це не простий і ітераційний процес, який ґрунтується на верхнеуровневих вимогах або тактичному технічному завданні. І тут ми стикаємося з необхідністю опрацювання уточнених вимог, які природним чином з'являються в процесі продумування поведінки і сценаріїв роботи системи.

Модельно-орієнтоване проектування надає гнучкі інструменти для створення і проектування архітектури системи у вигляді моделі, описи компонентів, їх властивостей, інтерфейсів між компонентами, а також аналізу властивостей системи.

Ключові особливості:

- Побудова архітектури системи в графічному вигляді,
- Задання властивостей компонентів і інтерфейсів між ними,
- Можливість подивитися на систему в різних видах, наприклад побачити все інтерфейси MIL-STD-1553, або ж тільки механічну підсистему,
- Проведення аналітики і розрахунків властивостей системи за допомогою скриптів.

Модельно-орієнтоване проектування засноване на побудові системної моделі, в якій об'єднані різні компоненти і підсистеми. На такій моделі різні команди розробників спільно перевіряють взаємний вплив підсистем, проводять будь-які віртуальні випробування, в тому числі і безпечно відпрацьовують аварійні або екстремальні режими роботи виробу. Системну модель потрібно використовувати не тільки як віртуальний полігон для відпрацювання різних сценаріїв роботи системи, але і як засіб проектування, тобто швидкого перебору рішень або вибору оптимального компонента для установки в систему. Вбудовані методи математичної оптимізації та калібрування дозволяють вибрати найкращі параметри для досягнення цільових показників поведінки системи.

Ключові особливості:

- Зручне зберігання і відпрацювання різних сценаріїв поведінки системи,
- Можливість командної розробки, в тому числі і з залученням зовнішніх підрядників,

- Вбудовані методи для автоматизованого підбору оптимальних параметрів системи або конкретних компонентів.

Цифрові алгоритми грають істотну роль в образі сучасних комплексних систем. МОП надає середовище для графічного побудови алгоритмів, набори готових бібліотек з реалізованими алгоритмами, а також кошти для автоматичної настройки алгоритмів для задоволення вимог технічного завдання. Зручність полягає в тому, що алгоритми можна описувати у вигляді М-мови, звичному для програмістів або у вигляді графічних діаграм, який легко сприймається інженерами і технологами. На цьому етапі алгоритм абстрагований від обчислювача, на якому буде запущений, що дозволяє сконцентруватися на вирішенні поставленого завдання і перебрати різні варіанти її вирішення, а не на програмуванні та тонкощах реалізації.

Ключові особливості:

- Готові графічні бібліотеки для побудови алгоритмів машинного і глибокого навчання, систем управління, цифрової обробки сигналів і машинного зору,
- Підпрограми для швидкого налаштування і автоматизованого підбору параметрів алгоритмів,
- Гнучкість побудови алгоритмів у вигляді програмного М-коду, графічних блоків або діаграм для опису кінцевих автоматів.

При проектуванні або виборі компонента для використання в системі, вкрай важливим є аналіз взаємовпливу цих підсистем або компонентів один на одного. МОП дозволяє в одній моделі об'єднати всі підсистеми і провести віртуальні комплексні випробування виробу. Це означає, що при розробці систем управління можна відтворити фізичну модель об'єкта управління, що містить механічну, гідравлічну, пневматичну, електричну та інші підсистеми. А при розробці систем ЦГЗ можна описати канал передачі даних, радіочастотний тракт, радіолокаційну обстановку, антенну решітку або навіть поставити перешкоди, що наводяться потенційним противником.

Ключові особливості:

- МОП надає необхідні для розробки алгоритмів інструменти для достовірного опису фізичного світу при розробці алгоритмів,
- Всі підсистеми описуються в єдиному середовищі і взаємно інтегровані,
- Вбудовані методи оптимізації дозволяють підібрати параметри віртуальних компонентів на базі експериментів, проведених на реальному об'єкті.

Модельно-орієнтоване проектування передбачає використання спеціалізованих програмно-апаратних комплексів для перевірки працездатності алгоритмів на реальному об'єкті. Такі комплекси застосовуються, коли цільової обчислювач, на якому будуть реалізовані алгоритми, ще не доступний, або перенесення алгоритмів на цільової обчислювач, а отже і перевірка працездатності, займає надто багато часу.

Застосування комплексів напівнатурного моделювання дозволяє запустити моделі алгоритмів в режимі жорсткого реального часу "натисканням однієї кнопки", а широкий вибір реалізованих інтерфейсів підключити алгоритм до реальної системи.

Застосовуючи цей метод, розробник алгоритмів може переконатися в працездатності та ефективності своїх алгоритмів до того, як передавати їх на реалізацію програмістам на цільовому обчислювачі, економлячи при цьому час і бюджет проекту.

Ключові особливості:

- Виключно просте і зручне використання стенду,
- Швидке прототипування і перевірка алгоритмів в реальному часі,
- Широкий вибір інтерфейсів: ARINC 429, MIL-STD-1553B, цифрові і аналогові вводи / виводи, ШІМ, CAN, SPI, I2C і т.д.,

- Зручні програмні засоби налагодження в реальному часі: візуалізація сигналів, настройка параметрів "на льоту", запис сигналів і створення віртуальних приладів.

Модельно-орієнтоване проектування надає інструменти для автоматичного перекладу моделей алгоритмів в вихідні коди для запуску на цільовому обчислювачі. Широта інструментів дозволяє автоматично генерувати вихідний код для різних мікропроцесорів, операційних систем реального часу, програмованих логічних інтегральних схем (ПЛІС), програмованих логічних контролерів (ПЛК) та вбудованих графічних процесорів (GPU). Такий підхід до розробки вбудованих алгоритмів виключає помилки ручного кодування, гарантує швидке внесення змін до проекту, забезпечує швидке перенесення алгоритмів з однієї елементної бази на іншу і з проекту на проект.

Ключові особливості:

- Автоматична генерація читаного, оптимізованого і однакового коду,
- Підтримка будь вітчизняної і зарубіжної елементної бази: Елвіс, Міландр, Модуль, Texas Instruments, Analog Devices, Xilinx, Intel (Altera) і т.д.,
- Простота повторного використання і внесення змін.

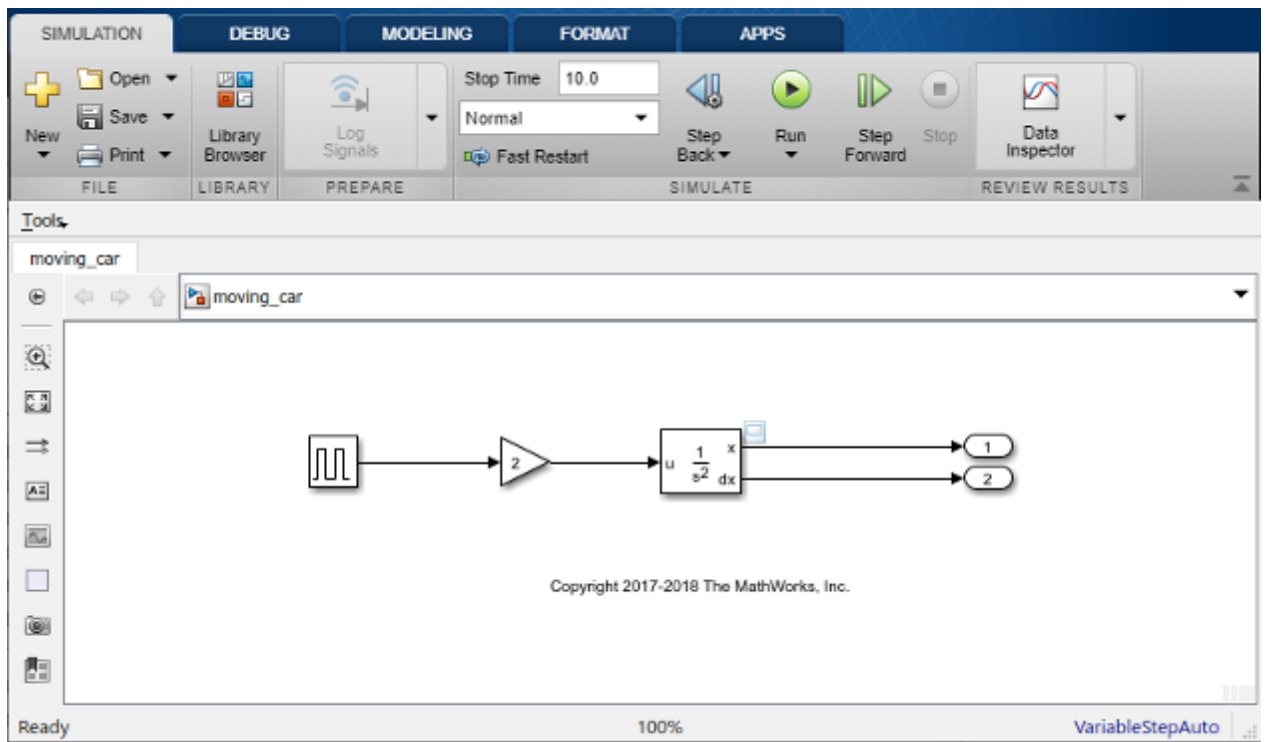


Рис. 2.1.1. Интерфейс программы Simulink.

2.2 Autodesk Simulation CFD

Simulation CFD(рис. 2.2.1) - це САЕ-система, призначена для розрахунків і моделювання руху потоків рідин і газів, а також процесів теплопередачі і тепломасообміну. Програма повністю відповідає національним стандартам ГОСТ, має всі необхідні сертифікати відповідності.

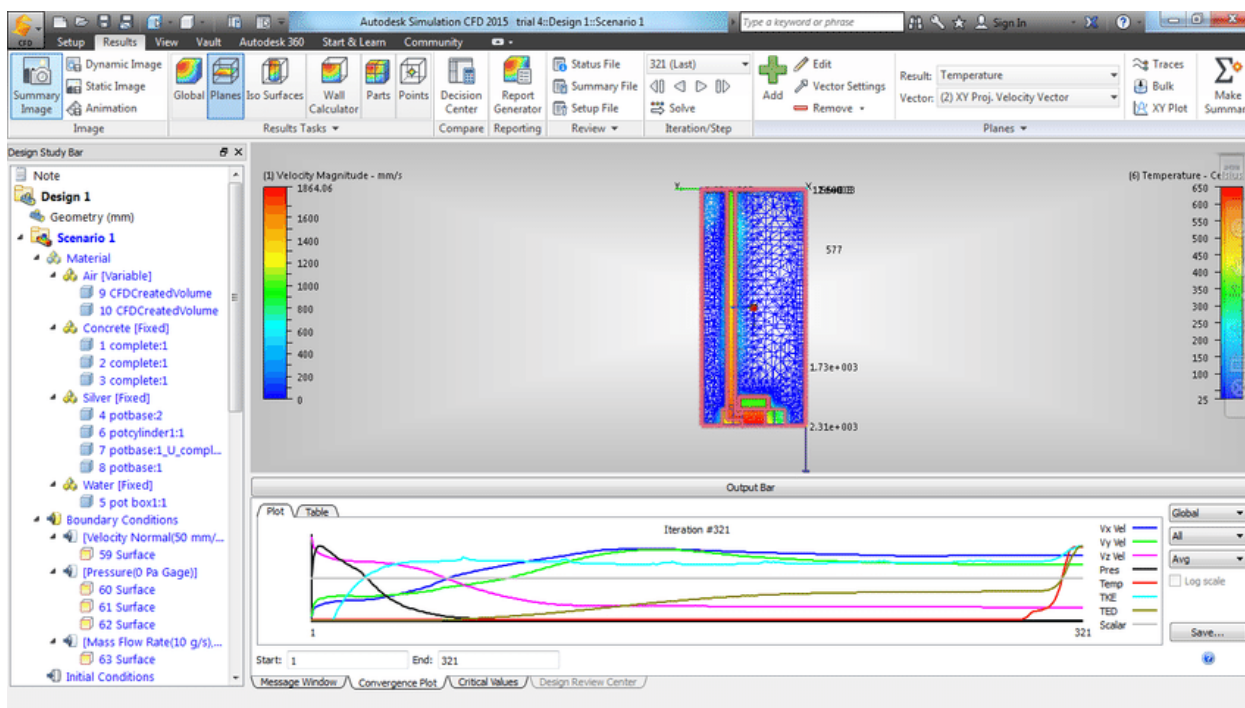


Рис. 2.2.1. Autodesk Simulation CFD.

У Simulation CFD використовуються просунуті механізми безпосереднього імпорту тривимірної геометрії з різних САПР. Маючи тривимірну модель, створену в одній з поширених САД-систем, користувач може відкрити її в Autodesk CFD і використовувати як основу для побудови розрахункової моделі.

Крім того, додаток Autodesk CFD можна інтегрувати безпосередньо в одну з поширених САПР (Inventor, Revit, SpaceClaim, PTC Creo, Solidworks і ін.).

Починаючи з 2019 версії, передплатники Autodesk CFD мають доступ до Fusion 360 Ultimate, який дозволяє спростити геометрію існуючої тривимірної

моделі і зробити її більш дружній до створення розрахункової моделі. Також за допомогою нього можна створювати прості тривимірні об'єкти, аналізувати і редагувати існуючі моделі. За допомогою Fusion 360 можна залити отвори, видалити зайві дрібні деталі з моделі (фаски, скруглення і ін.), які не впливають на результати розрахунку, але їх обробка і розрахунок можуть займати велику кількість часу.

Грамотне застосування Fusion 360 для підготовки моделі сприяє формуванню якісної сітки і скорочення часу розрахунку.

До складу Autodesk CFD входить повноцінна бібліотека матеріалів, речовин і компонентів, які широко використовуються в промисловості. Кожен об'єкт бібліотеки містить в собі опис всіх своїх теплофізичних характеристик, що дозволяє застосовувати їх у формуванні розрахункових завдань. У бібліотеці можна знайти різні рідини і гази, пасивні та активні компоненти електронних приладів, пристрої для створення систем теплообміну і багато іншого.

Бібліотека Autodesk CFD повністю відкрита, тому кожен користувач може створювати необмежену кількість своїх матеріалів і об'єктів або редагувати існуючі.

При розробці Autodesk CFD був реалізований важливий принцип - програма має бути максимально простою і зрозумілою, навіть тим користувачам, які не мають досвіду проведення гідро- і газодинамічних розрахунків в програмних системах. Всі елементи інтерфейсу програми давно стали стандартними і знайомими про інші продукти Autodesk, що дозволяє легко освоїти систему і швидко почати працювати в ній.

У Autodesk CFD можна знайти велику кількість інструментів, що спрощують і прискорюють роботу зі створення завдань, проведення розрахунків та аналізу результатів. Продумані майстра і діалогові вікна, наочні інструменти введення і виведення результатів, засоби визначення навантажень

і коефіцієнтів, покликані прискорити роботу користувачів. Для початкового освоєння програми доступні покрокові текстові відеоінструкції, а повноцінна довідкова система дозволить розібратися у всіх тонкощах роботи.

Автоматична генерація розрахункових сіток. З одного боку, генератор розрахункової сітки, реалізований в Autodesk CFD, здатний позбавити користувача від рутинних процедур, оскільки генерує дуже якісну сітку кінцевих елементів в автоматичному режимі. З іншого боку, досвідчені користувачі можуть самостійно налаштувати параметри генерації сітки і вручну виділити окремі зони, де необхідні особливі параметри сітки. Сітковий генератор здатний автоматично виділити області, в яких необхідно реалізувати різні види сіток. Наприклад, для точного моделювання течії в пограничному шарі система формує в потрібних областях пристінковий шар з призматичних елементів. При формуванні сіток в великих обсягах система самостійно зменшить розмір осередків в тих місцях, де присутні дрібні об'єкти або відбувається зміна форми або напрямку потоку.

В основі Autodesk CFD лежать два унікальних математичних решателя: CFD і CFD2. Вирішувач CFD розроблявся з найперших версій програми і орієнтований на обчислення простих завдань. Більш новий і просунутий вирішувач CFD2 розроблявся спеціально для розрахунку великих завдань, що містять 30-50 мільйонів елементів. Він повністю підтримує багатопроцесорні і багатопотокові технології, автоматично розподіляє обчислення по процесорам, ядрам і потокам.

Вирішувачі налаштовані таким чином, що підходять для роботи з більшістю завдань. Однак досвідчені і просунуті користувачі мають досвід точної настройки алгоритмів. Наприклад, на вибір доступні різні схеми адвекції, моделі турбулентності і багато іншого.

У Autodesk CFD реалізована підтримка більшості реальних фізичних процесів:

- моделювання зовнішніх і внутрішніх нестискуваних і стискаються потоків;
- моделювання нестационарних потоків;
- моделювання процесів теплопередачі;
- робота з періодичними граничними умовами;
- моделювання вільного перебігу рідини методом вільної поверхні (Volume of Fluid - VOF);
- скалярне змішування потоків;
- робота з процесами випаровування та випромінювання;
- прогнозування потенційних місць виникнення кавітації і ерозії;
- моделювання та аналіз сонячного нагріву і нагрівання під дією електричного струму;
- засоби візуалізації диму;
- розрахунок і аналіз теплового комфорту;
- робота з ротаційними машинами (осьові компресори, насоси, вентилятори), теплообмінниками, друкованими платами, термоелектричними компонентами, тепловими радіаторами;
- підтримка проникних матеріалів.

Висновки до розділу

Було проаналізовано два, дуже потужних програмних комплекси, функціонал яких є дуже об'ємний. За допомогою них можна розраховувати, інтегрувати розрахунки, експортувати та імпортувати файли з розрахунками або вхідними даними, що є безпосередньо плюсом. Але наявність багатого

функціоналу є і мінусом, тому що данні програми використовують ресурс комп'ютера не належним чином.

3. ФОРМУВАННЯ МОДЕЛІ ЛІТАКА ДЛЯ РОЗРАХУНКУ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК І ВИМОГИ ДО ВХІДНИХ ТА ВИХІДНИХ ДАНИХ.

Згідно з технічним завданням вхідні данні до роботи наступні(табл. 3.1).

Табл. 3.1.

Найменування	Значення
Пасажиромісткість	2 місця
Дальність польоту	1300 км
Потолок польоту	8500 м
Злітна вага	500 кг
Крейсерська швидкість	175 км/год
Максимальна швидкість	215 км/год
Швидкість посадки	100 км/год
Швидкість зльоту	100 км/год

У якості літака прототипа було обрано літак компанії Аеропракт А-32.

У 2014 році київська компанія "Аеропракт" розробила (головний конструктор літака Юрій Яковлев.) Новий літак і підготувала його до серійного виробництва. На початку 2000-х "Аеропракт" проробляло ідею створення швидкісного двомісного нізкоплана з двигуном Rotax-912 на додаток до серійного "А-22", який був запущений у виробництво в 1996 році. Називався цей проект "Аеропракт-32" (А-32). Планувалося застосувати крило з хордою 1,2 м і профілем НАСА 63 (3) -418. Таке ж застосовувалося в той час на спортивному літаку "Аеропракт-30", ми мали всю необхідну оснастку для виготовлення деталей для нього і складальний стапель.

Поступово прийшло розуміння, що ідея літака "Аеропракт-22" з хорошими злітно-посадковими характеристиками дуже затребувана на ринках різних країн і її варто розвивати. Таким чином, ми прийшли до думки, що новим

літаком під назвою "Аеропракт-32" буде підкисний високоплан з перевіреним і високоефективним крилом "А-22" і поліпшеною аеродинамікою для досягнення більш високих крейсерських швидкостей. Ми поставили скромне завдання отримати надбавку 10 км / год до наявних 185 км / год максимальної швидкості "А-22".

Основною відмінністю нової машини є форма фюзеляжу. При створенні його обводів ставилося завдання не тільки зменшити опір, але і зберегти впізнаваність літака марки "Аеропракт", сімейні риси його "старшого брата" А-22. Після створення комп'ютерної моделі фюзеляжу нової машини за допомогою програмного забезпечення нами були зроблені його "продувки". Одночасно була створена модель фюзеляжу "А-22" і таким же чином "продута" для порівняння. Візуалізація показала, що на великій швидкості нижня панель фюзеляжу А-22, яка знаходиться за балкою шасі, генерує потужний вихор і, відповідно, опір. Фюзеляж нової машини позбавлений цього недоліку.

Зовнішня поверхня фюзеляжу складається з конічних і циліндричних панелей. Важливою особливістю форми є плавне сполучення поверхонь бічній частині фюзеляжу зі сферичною поверхнею скла двері. Конструкція фюзеляжу класична, полумонококовою.

Для обшивок застосовуються листи з американського алюмінієвого сплаву 2024Т3 товщиною від 0,5 до 1,0 мм. З метою поліпшення комфорту в кабіні її довжина була збільшена на 50 мм.

З простору кабіни прибрані поперечна і діагональні труби в районі лобового скла, що поліпшило огляд пілота. Сидіння в польоті можуть пересуватися, як в автомобілі. Спинки сидінь можуть складатися вперед для кращого доступу до багажника. В інтер'єрі кабіни широко використовуються пластикові панелі. З цього ж матеріалу виготовлений корпус приладової дошки. Завдяки зміні конструкції корпусу приладової дошки була зменшена

на 25 мм його висота, що поліпшило огляд пілота. Висота приладової дошки не змінилася, що дає можливість розміщення широкого спектру обладнання.

Певною несподіванкою виявилася розробка каркаса двері. На це ми витратили тривалий час, глибоко пропрацювавши і виготовивши кілька варіантів конструкції двері. Спочатку було вирішено зробити каркас дверей, аналогічний каркасу двері літака А-22, що складається з алюмінієвих трубок, з'єднаних зварними фітингами на односторонніх заклепках. В процесі льотних випробувань з'ясувалося, що при випуску закрилків на граничній для них швидкості в районі передньої частини двері відчинялися щілину до одного сантиметра в результаті утворення області розрядження, яка "відсмоктує" двері від отвору фюзеляжу. Необхідно було збільшити жорсткість каркаса двері, щоб уникнути цієї неприємності. Після цього були розроблені один металевий і два композитних каркаса двері, поки не отримали задовольняє нас результат. В результаті вийшла дуже жорстка двері з композитним каркасом і приклеєним до нього склом на автомобільному герметіку.

Установка двигуна теж істотно відрізняється від А-22. Так як в конструкції фюзеляжу нового літака відсутні діагональні труби, сходяться в одну точку у верхній частині моторного шпангоута, відмовилися від тяги, яка з'єднує цю точку і редуктор двигуна, як це зроблено в А-22. Двигун нової машини встановлений на подмотораме, що складається з чотирьох деталей складної форми, виготовлених з алюмінієвих листів товщиною 8 мм. Складність форми обумовлена бажанням розмістити двигун якнайнижче і розійтися з численними шлангами маслосистеми і системи охолодження двигуна. Двигун разом з подмоторамой кріпиться до звареної ферменної мотораме через шість гумових амортизаторів. На літаку вирішено було застосувати так зване "тунельний" розташування радіаторів, яке ми відчували на нашому літаку А-22 протягом декількох років. Радіатор охолоджуючої рідини і маслорадіатор розміщені один за іншим в тунелі з алюмінієвого листа. На виході з тунелю встановлена керована заслінка, закриваючи яку пілот одночасно відкриває

доступ теплого повітря через канали в кабіну. Довгий час зайняло знаходження гарної форми капота двигуна. Літак почав літати з дуже функціональним і обтічним.

На Рис. 3.1. можна побачити зображення літака Аеропракт А-32.



Рис. 3.1. Загальний вигляд літака Аеропракт А-32.

У табл. 3.2 зображені основні ЛТХ літака.

Найменування параметру	Значення параметру
Розмах крила	9,45 м
Довжика	6,29 м
Висота	2,47 м
Площа крила	12,83 м ²
Маса	600 кг
Пальне	90 л
Тип двигуна	1 ПД Rotax-912S
Максимальна швидкість	215 км/год
Крейсерська швидкість	180 км/год
Максимальна дальність польоту	1100 км

Практичний потолок	8500 м
Екіпаж	2 люд.

Як бачимо данні значення максимально близькі до даних значень у технічному завданні.

На рис. 3.2 зображені проєкції літака у трьох видах.

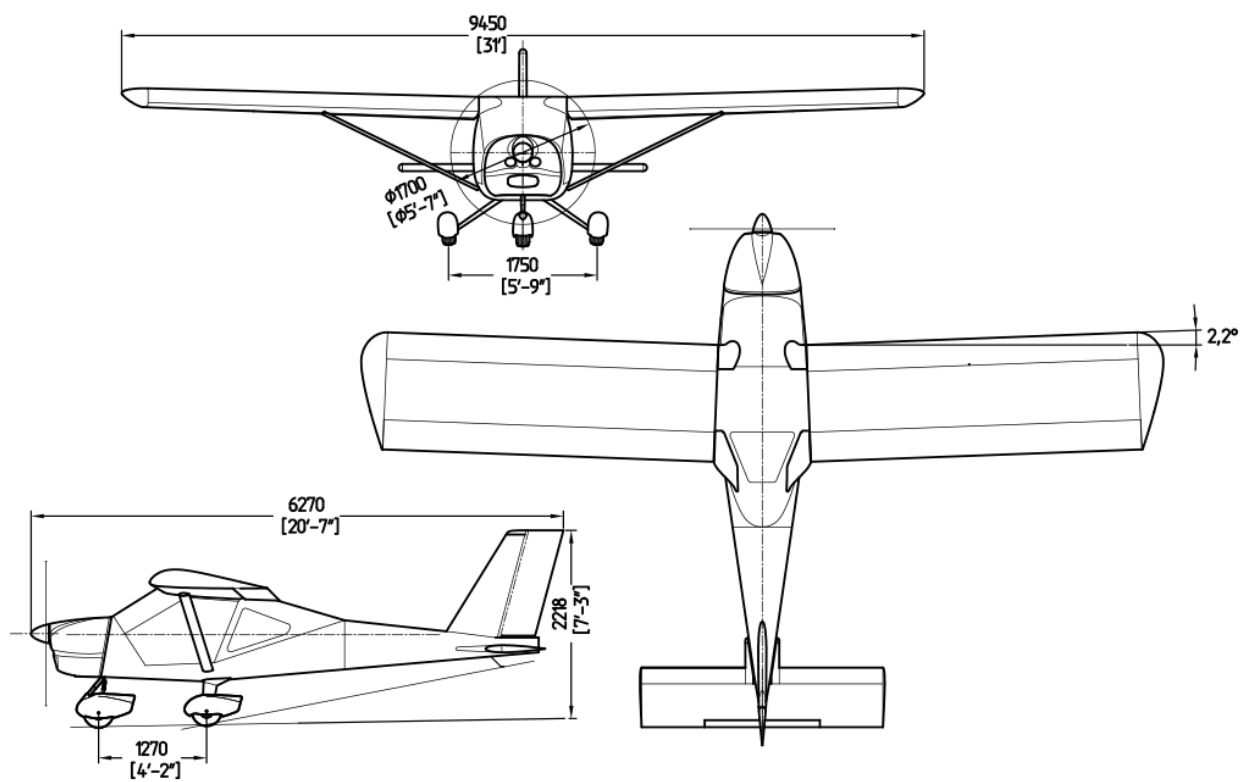


Рис. 3.2. Проєкції літака Аеропракт А-32.

За даними, які вказані вище, за допомогою програмного забезпечення PTC Creo Parametric була створена продувочна модель літака(рис. 3.3.), прототипом якої став літака компанії АЕРОПРАКТ А-32.

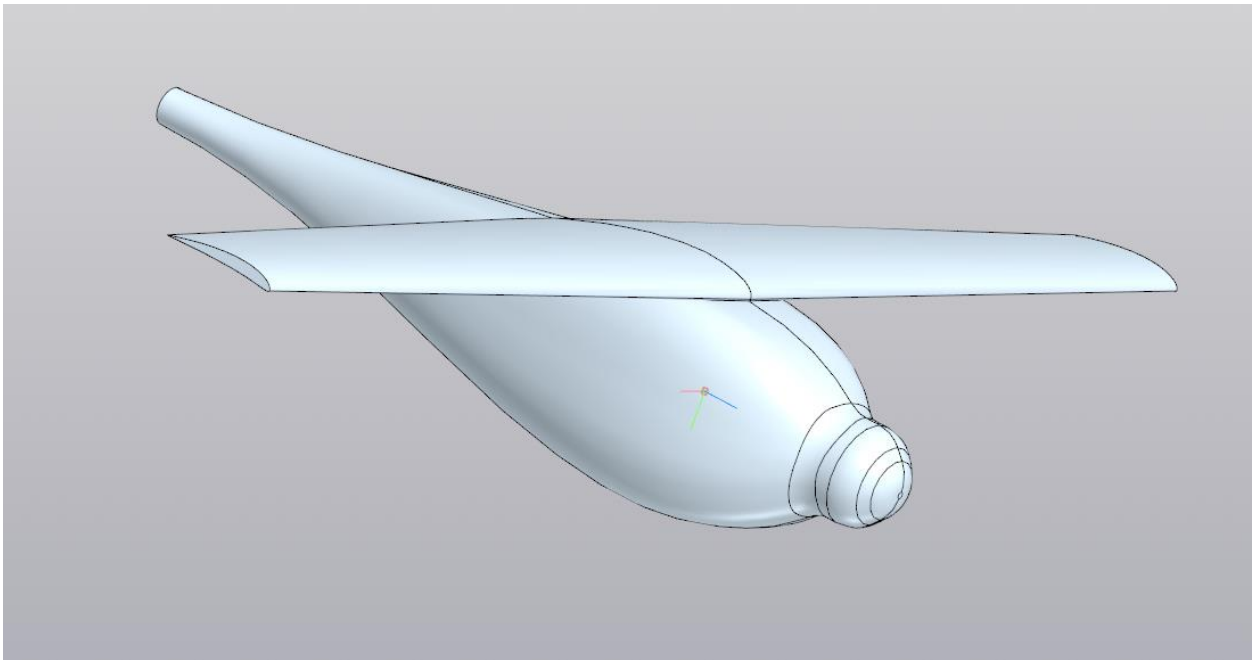


Рис. 3.3. Продувочна модель літака.

Висновки до розділу

Згідно з вихідними даними технічного завдання, було обрано літак-прототип Аеропракт А-32. Це є надлегкий двухмісний літак, за конструкцією – одномоторний поршневий високоплан з тяговим гвинтом та шасі на трьох опорах.

4. РОЗРОБКА ІНТЕРФЕЙСУ ПРОГРАМНОГО КОМПЛЕКСУ

Проектування інтерфейсу програмного комплексу є важливою задачею у процесі розроблення програмного додатку тому, що саме за допомогою інтерфейсу користувач сприймає програму та розуміє як використовувати її функціональність.

Задоволеність користувача у програмному комплексі визначається, в значній мірі, лише інтерфейсом його. Основні фактори, які впливають на задоволеність такі:

- Можливості інтерфейсу,
- Час роботи та відгуку до звичайних дій з інтерфейсом,
- Надійність інтерфейсу
- Зрозумілість інтерфейсу, тощо.

Ці фактори відіграють найбільш важливішу роль у отриманні позитивного UX(user experience – досвід користувача).

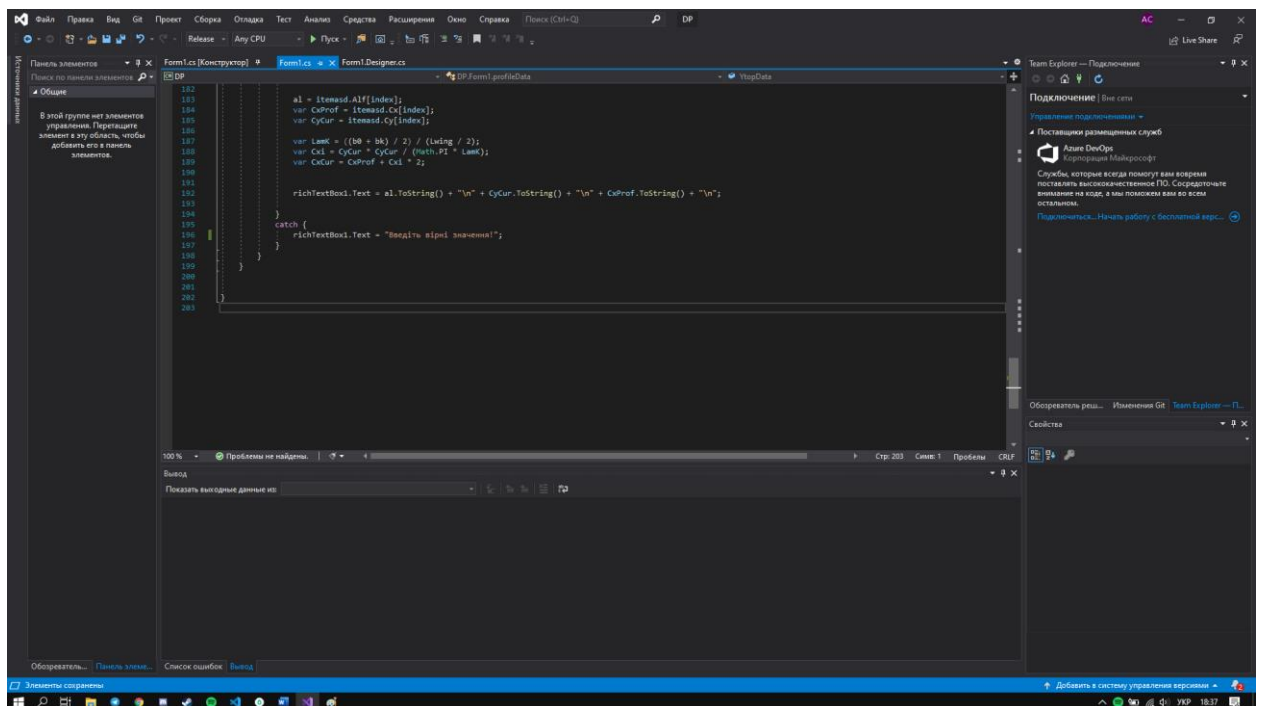
Існує декілька підходів до розробки програмного комплексу. Перший з них – проектування «зовні-всередину», де весь підхід спрямований на інтерфейс та на покращення UX користувача. Другий з них – проектування «зсередини-назовні». Цей підхід базується на тому, що спочатку розробляють функціональну частину програми, а потім, на основі цієї частини – інтерфейс користувача. Третій підхід – еволюційне проектування. Цей підхід базується на зосередженні побудови продукту з покроковим нарощуванням і уточненням можливостей продукту. В цьому проекті був реалізований підхід еволюційного проектування.

Для розробки цього проекту було обрано об'єктно-орієнтовану мову програмування C#. C# (сі-шарп) - був представлений у 2001 році компанією Microsoft. C# має всі переваги сучасного об'єктно-орієнтованої мови програмування:

- Строга типізація даних,
- Поліморфізм
- Перегрузка операторів
- Вказівники
- Атрибути і т.д.

На сьогоднішній момент С# - одна з найбільш потужних та швидко розвиваючих і затребуваних мов програмування. В даний момент на ньому пишуть різні програмні додатки: від невеликих комп'ютерних програм до великих веб-порталів та веб-сервісів, на які заходять мільйони користувачів кожен день.

Для розробки була обрана IDE(integrated development environment – інтегрована середовище розробки) Microsoft Visual Studio 2019(рис 4.1). Ця IDE включає в себе редактор коду, з підтримкою технології IntelliSense(рис. 4.2) – технології автодоповнення. Данна технологія пропонує список методів, або назви змінних і т.д. при введенні початкових букв.



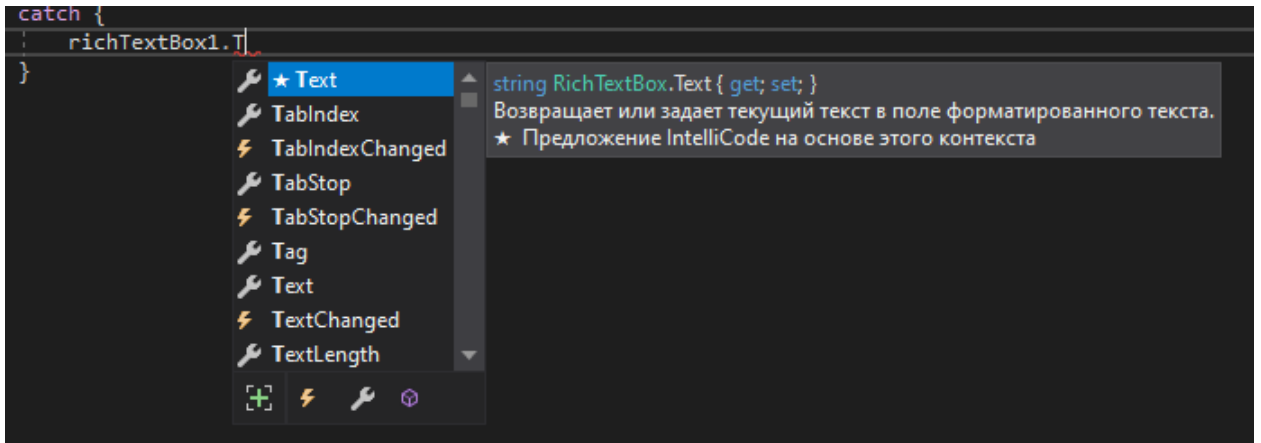


Рис. 4.2. Технологія IntelliSense.

Як вказано вище, інтерфейс виконувався покроково. Першим етапом було – реалізувати функції вибору профілю крила та відмальовування його профілю. На рис. 4.3. можна побачити перший етап побудови інтерфейсу.



Рис. 4.3. Перший етап побудови інтерфейсу.

Другим етапом було реалізувати поля для вводу даних. Усі дані мають вводитись у системі СІ та мають містити у собі тільки цифри. На рис. 4.4. можна побачити другий етап побудови інтерфейсу.

Кінцевая хорда крила	<input type="text"/>
Корнева хорда крила	<input type="text"/>
Размах крила	<input type="text"/>
Кут встановлення	<input type="text"/>
Довжина фюзеляжу	<input type="text"/>
Макс. екв. діаметр	<input type="text"/>
Швидкість набіг. потоку	<input type="text"/>
Кут атаки	<input type="text"/>
Маса літака	<input type="text"/>

Рис. 4.4. Другий етап побудови інтерфейсу.

Третім та останнім етапом було будівництво графіку поляри для обраного крила та додання поля для виводу даних та кнопки. На рис. 4.5. можна побачити третій етап побудови інтерфейсу.



Рис 4.5. Третій етап побудови інтерфейсу.

В результаті вийшов простий та зрозумілий користувачу інтерфейс програмного комплексу (рис. 4.6).

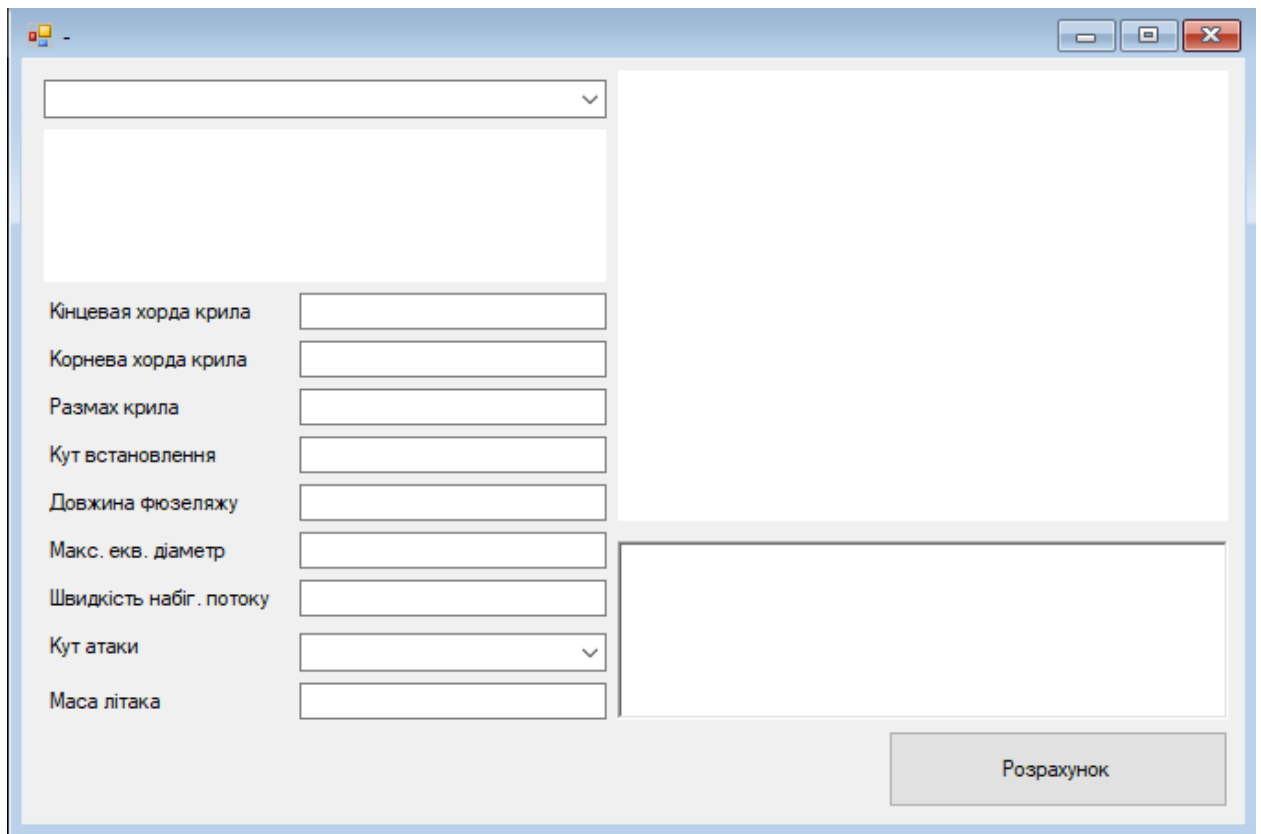


Рис. 4.6. Інтерфейс програмного комплексу.

Висновок до розділу

Інтерфейс програми було створено у три кроки за допомогою мови програмування С#. Мова програмування була обрана як сучасна та високофункціональна мова програмування, що є дуже великим плюсом до подальшого розширення функціоналу програми. Програмування інтерфейсу відбувалось в потужному додатку, з технологією автодоповнення IntelliSense - IDE Microsoft Visual Studio.

5. РОЗРОБЛЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО МОДУЛЮ ДЛЯ РОЗРАХУНКІВ.

При розробці програми дуже важливою частиною розробки – є здатність програми до майбутнього розширення функціоналу. При виконванні цього проекту головною ідеєю було майбутнє розширення функціоналу програми, як програмного комплексу.

Розроблення додатку почалось зі створення класу для задання профілю. Клас профілю – `profileData`, складається з 7 полів даних, такі як:

- Назва
- Координати X профілю
- Координати Y верхньої кромки профілю
- Координати Y нижньої кромки профілю
- Значення кута атаки
- Координати коефіцієнту підйомної сили для даного профілю
- Координати коефіцієнту опору для даного профілю.

Всі поля класу мають тип даних – `double[]` – масив чисел з плаваючою точкою, окрім двох полів. Поле `name` – має тип даних `string` – текстовий формат. Поле `alfData` – тип `int[]` – цілочисельний масив. Клас `profileData` містить так звані гетери та сетери – методи для задання та отримання даних з об'єкту класу.

Було створено два об'єкти класу `profileData`(додаток 1) для демонстрації роботи програми.

```
public class profileData
{
    public string name;
    public double[] XData;
    public double[] YtopData;
    public double[] YbotData;
    public int[] alfData;
    public double[] CyData;
    public double[] CxData;
    Ссылка: 2
    public profileData(string profileName, double[] X, double[] Ytop, double[] Ybot, int[] alf, double[] Cy, double[] Cx)
    Ссылка: 2
    public double[] X
    {
        get => XData;
    }
    ссылка: 1
    public double[] Ytop
    {
        get => YtopData;
    }
    ссылка: 1
    public double[] Ybot
    {
        get => YbotData;
    }
    Ссылка: 2
    public string Name
    {
        get => name;
    }
    Ссылка: 2
    public double[] Cy
    {
        get => CyData;
    }
    Ссылка: 2
    public double[] Cx
    {
        get => CxData;
    }
    ссылка: 1
    public int[] Alf
    {
        get => alfData;
    }
}
```

Рис. 5.1. Структура класу profileData.

Наступним кроком є вимальовування профілю крила. Цей етап можна поділити на дві частини. Першою частиною є ініціалізація графіку, та добавлення до нього колекції даних, яка являє собою два об'єкти профілів, які були створені раніше.

Створюється клас Item, який містить два поля. Перше поле містить у собі назву профілю. Друге – містить у собі об'єкт з координатами профілю.

```
public class Item
{
    Ссылка: 2
    public Item() { }
    Ссылка: 4
    public object Value { set; get; }
    Ссылка: 2
    public string Text { set; get; }
}
```

Рис. 5.2. Структура класу Item.

```

public void InitPlot()
{
    List<Item> profiles = new List<Item>();
    profiles.Add(new Item() { Text = prof1.Name, Value = prof1 });
    profiles.Add(new Item() { Text = prof2.Name, Value = prof2 });

    comboBox1.DataSource = profiles;
    comboBox1.DisplayMember = "Text";
    comboBox1.ValueMember = "Value";
}

```

Рис. 5.3. Ініціалізація графіку.

Порядок ініціалізації простий. Спочатку створюється список з типом даних Item, який описаний вище. Наступним кроком до нього записують два профіля. Останнім кроком обирається джерело даних для відображення списку профілів у випадяючому меню comboBox1.

Другою частиною побудови профілю крила – є відмальовування профілю крила.

```

public void CreateProFileChart(profileData obj)
{
    chart1.Series.Clear();

    var topPart = new Series();
    var botPart = new Series();

    topPart.Points.DataBindXY(obj.X, obj.Ytop);
    chart1.Series.Add(topPart);
    topPart.ChartType = SeriesChartType.Spline;
    topPart.Color = Color.Black;

    botPart.Points.DataBindXY(obj.X, obj.Ybot);
    chart1.Series.Add(botPart);
    botPart.ChartType = SeriesChartType.Spline;
    botPart.Color = Color.Black;
}

```

Рис. 5.4. Порядок відмальовування профілю крила.

Першим кроком цього етапу є очищення графіку від попередньо відмальованих профілів. Потім створюється дві змінних topPart та botPart типу Series. Series – об’єкт, який представляє собою ряд даних. До кожної з цих змінних записуються данні про профіль, у topPart – координати точок для верхньої кромки профілю, у botPart – координати для нижньої кромки

профілю. Обирається тип графіку сплайн, чорний колір графіку та відбувається побудова профілю крила.

Наступним кроком є зчитування даних, які ввів користувач та обрахування потрібних нам значень по формулам. Введені значення повинні бути у системі відліку СІ та записані через кому. Поля мають містити в собі тільки цифри та кому, тому проводиться перевірка на правильність вводу даних методом try catch. Даний метод перевірки даних заключається в тому, що програма пробує виконати код, і якщо виникає програмна помилка (у деякому полі число записано через крапку, або не введене значення або в полі літера) то виконання програми припиняється, та з'являється вікно, яке вказує на помилку, та просить ввести коректні данні

```
try
{
    bk = float.Parse(bkTextBox.Text);
    b0 = float.Parse(b0TextBox.Text);
    Lwing = float.Parse(LwingTextBox.Text);
    Lfus = float.Parse(LfusTextBox.Text);
    Dmax = float.Parse(DmaxTextBOx.Text);
    v = float.Parse(VTextBox.Text);
    a1 = float.Parse(lamComboBox.Text);
}
catch
{
    richTextBox1.Text = "Значення введені невірно! \n" + "Введіть вірні значення!";
}
```

Рис. 5.5. Перевірка правильності введення даних.

Далі реалізовується функція зміни профілю(рис. 5.6).

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    Item item = comboBox1.SelectedItem as Item;
    CreateProFileChart(item.Value as profileData);
}
```

Рис. 5.6. Функція зміни профілю.

Наступним кроком знаходяться значення коефіцієнтів підйомної сили та опору для даного кута атаки. Оскільки поляра побудована для видовження крила рівним бескінечності, тому коефіцієнт опору є профільним.


```
//Знаходження Cx та Cy для даного куту атаки
a1 = itemasd.Alf[index];
var CxProf = itemasd.Cx[index];
var CyCur = itemasd.Cy[index];
//-----
```

Рис. 5.7. Знаходження коефіцієнтів підйомної сили та профільного опору.

Вираховується видовження крила, індуктивний опір крила, та коефіцієнт опору крила, який є сумою індуктивного та профільного опорів.

```
//Обрахування видовження крила та коефіцієнту індуктивного опору
var LamK = (Lwing / 2) / ((b0 + bk) / 2);
var Cxi = CyCur * CyCur / (Math.PI * LamK);
var Cxs = CxProf + Cxi * 2;
//-----
```

Рис. 5.8. Обрахування видовження крила, коефіцієнту індуктивного опору та коефіцієнту опору крила.

Проводиться обрахування площі крила.

```
//Знаходження площі крила
var b = Math.Sqrt(Math.Pow((Lwing/2), 2) + Math.Pow((b0 - bk), 2));
var Sk = (b + Lwing / 2) * b0 * 2;
//-----
```

Рис. 5.9. Знаходження площі крила.

Наступним кором знаходиться підйомна сила та сила опору крила.

```
//знаходження підйомної сили та сили опору
var g0 = 1;
var Y = CyCur * g0 * Math.Pow(v, 2) * Sk / 2;
var X = CxCur * g0 * Math.Pow(v, 2) * Sk / 2;
//-----
```

Рис. 5.10. Знаходження підйомної сили та опору крила.

Знаходиться площа повздовжнього перетину фюзеляжу та обраховується опір та підйомна сила фюзеляжу.

```

var LamFus = Lfus / Dmax;
var SD = Lfus * Dmax * Math.PI / 4;
var Fxp = cxf * Sfus * ro * v * v / 2 * Math.Cos(alf);
var Falfa = Math.Abs(cp * Math.Sin(alf) * SD * ro * v * v / 2);
var Fxf = Fxp + Falfa;
var Fyf = cp * Math.Sin(alf) * Math.Cos(alf) * SD * ro * v * v / 2;

```

Рис. 5.11. Знаходження площі повздовжнього перетину фюзеляжу та
обрахування опору та підйомної сили фюзеляжу.

Обраховується повна підйомна сила та сила опору літака.

```

//Знаходження загальної підйомної сили та сили опору літаки
var Yk = Y + Fyf;
var Xk = X + Fxf;
//-----

```

Рис. 5.12. Повна підйомна сила та сила опору літака.

Далі знаходиться потрібна потужність.

```

//Знаходження потрібної потужності
var m = float.Parse(textBox1.Text);
var K = Yk / Xk;
var P = m * 9.81 / K;
var N = P * v;
//-----

```

Рис. 5.13. Знаходження потрібної потужності.

Останнім кроком виводяться дані на екран користувача.

```

richTextBox1.Text = "Підйомна сила Y = " + Math.Round(Yk, 2).ToString() + "\n";
richTextBox1.Text += "Сила опору X = " + Math.Round(Xk, 2).ToString() + "\n";
richTextBox1.Text += "Потрібна потужність N = " + Math.Round(N, 2).ToString() + "\n";

```

Рис. 5.14. Вивід отриманих даних.

Залежність коефіцієнту підйомної сили від кута атаки для обраного профілю та поляра крила представляються у вигляді системи точок, поміщених в таблицю.

Розрахунок підйомної сили F_y крила здійснювався на основі добутку коефіцієнту підйомної сили c_y та площі S_w крила в плані:

$$F_y := c_y \cdot S_w \cdot \rho \cdot \frac{V^2}{2}$$

де ρ – густина повітря, V - повітряна швидкість.

За однотипним виразом розраховується сила опору крила F_x

$$F_x := c_x \cdot S_w \cdot \rho \cdot \frac{V^2}{2}$$

де c_x – коефіцієнт аеродинамічного опору крила, який складається із профільного c_{x0} для крила нескінченної довжини та індуктивного c_{xi} опору. Коефіцієнт профільного опору визначається за полярою, а індуктивного - розраховується за коефіцієнтом підйомної сили [2].

Профільний аеродинамічний опір фюзеляжу визначався як пропорційний площі поперечного перетину фюзеляжу S_F (перпендикулярної до повздожньої вісі фюзеляжу):

$$F_{xp} := c_{xf} \cdot S_F \cdot \rho \cdot \frac{V^2}{2} \cdot \cos(\alpha)$$

де c_{xf} - коефіцієнт опору тіла обертання еквівалентного поперечного перетину S_F та рівного видовження λ до фюзеляжу, α – кут атаки літака.

Підйомна сила фюзеляжу F_{yf} з'являлась при наявності кута атаки і визначалась площею повздожнього перетину S_D фюзеляжу:

$$F_{yf} := c_p \cdot \sin(\alpha) \cdot \cos(\alpha) \cdot S_D \cdot \rho \cdot \frac{V^2}{2}$$

Добуток поправочного коефіцієнту c_p та синусу кута атаки α можна вважати коефіцієнтом підйомної сили фюзеляжу

$$c_y := c_p \cdot \sin(\alpha) \cdot \cos(\alpha)$$

При наявності кута атаки аеродинамічний опір фюзеляжу також збільшуватиметься

$$F_{xf} := F_{xp} + F_{alfa}$$

за рахунок попадання повздовжньої проекції фюзеляжу в набігаючий потік під зазначеним кутом $alfa$

$$F_{alfa} := c_p \cdot \sin(alfa) \cdot SD \cdot \rho \cdot \frac{V^2}{2}$$

Коефіцієнт аеродинамічної якості визначався як відношення сумарної підйомної сили крил та фюзеляжу до сумарного опору фюзеляжу та крил.

$$K = \frac{F_y + F_{yf}}{F_x + F_{xf}}$$

Необхідна потужність двигуна визначається як добуток ваги $G=mg$ та швидкості літака V , поділені на аеродинамічну якість K та коефіцієнт корисної дії гвинта:

$$P = G \cdot V / (K \eta)$$

Не дивлячись на те, що всі коефіцієнти та параметри в цій системі формул є теоретично визначеними, проте залишають можливість для їх уточнення за результатами експериментів.

Тестування програмного модулю проводилось для ЛТХ літака схожого з ЛТХ А-32 від АЕРОПРАКТ. На рис. 5.15. можна бачити інтерфейс програми з введеними даними для обрахування та отримані результати, де Y – значення підйомної сили, X – значення сили опору, а N – значена потрібної потужності.

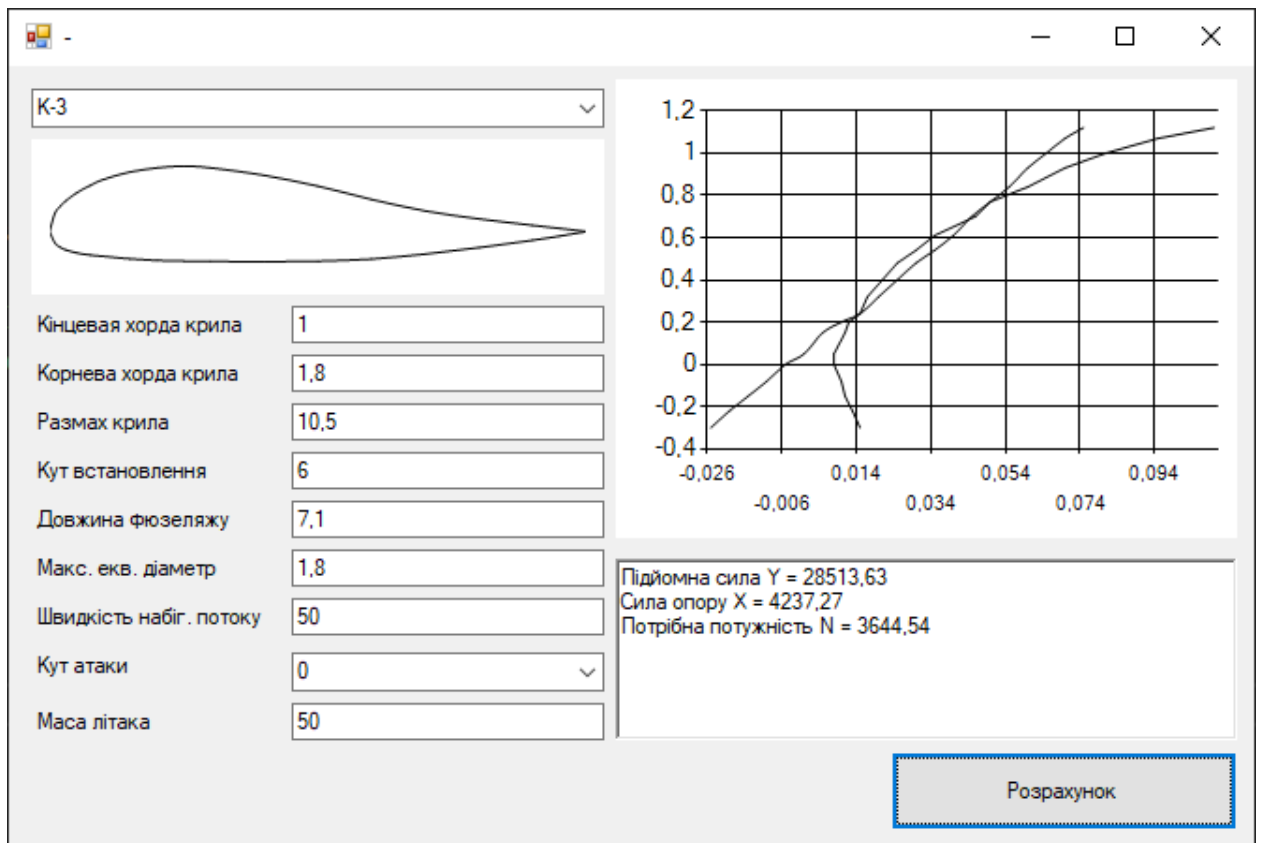


Рис. 5.15. Результати розрахунку.

За допомогою програмного забезпечення PTC Creo Parametric була створена продувочна модель літака.

За допомогою програмного забезпечення Autodesk Simulation CFD була проаналізована аеродинаміка створеної моделі(рис. 5.16). Порівняння результатів розрахунків аеродинамічних навантажень можна побачити на табл. 5.1.

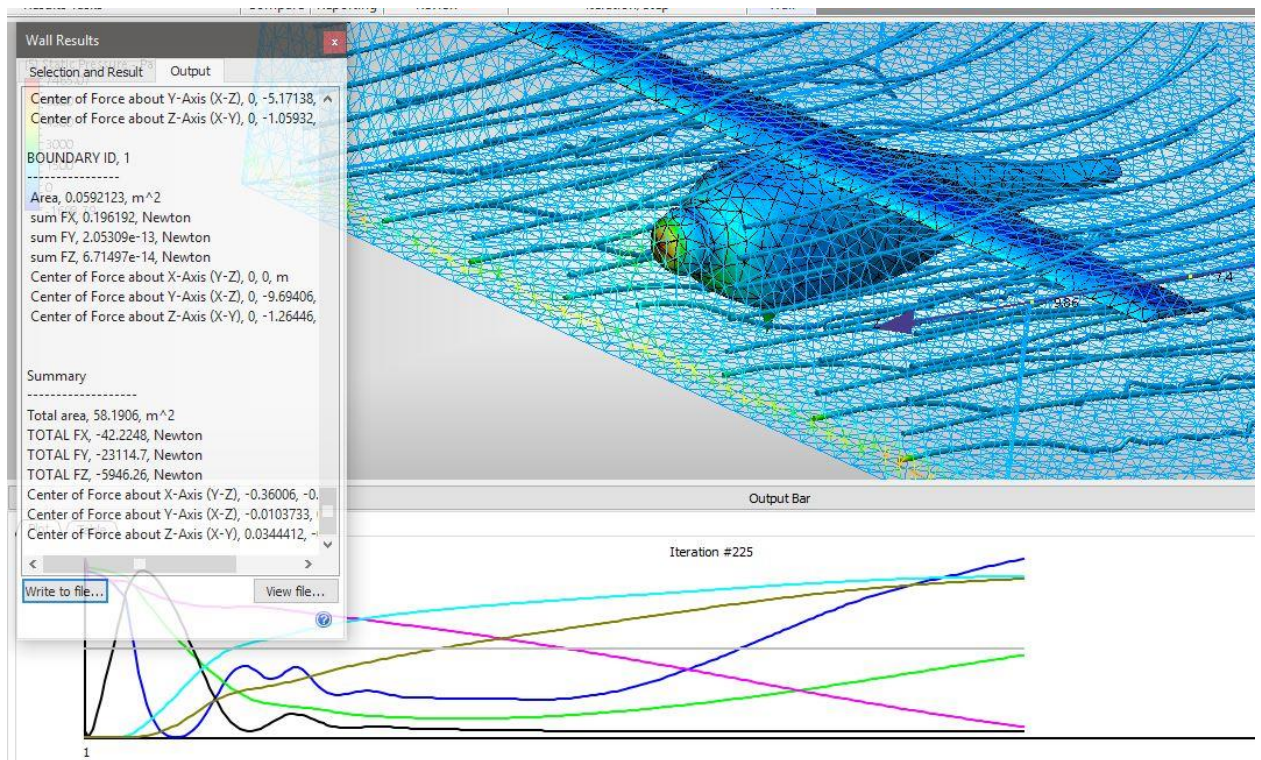


Рис. 5.16. Результати аналізу аеродинаміки за допомогою програми Autodesk Simulation CFD.

Таблиця 5.1.
Порівняння результатів розрахунків аеродинамічних навантажень

Кут атаки, град.	Графічний інтерфейс користувача		Метод кінцевих елементів	
	Підйомна сила, Н	Сила опору, Н	Підйомна сила, Н	Сила опору, Н
0	28500	4200	27400	6100
2	37600	6900	37460	7400
4	48400	10000	48030	9874
6	59300	13200	57670	12900

Як можна бачити, отримані результати підйомної сили у програмі CFD – 27ю.4 кН, у створеній програмі – 28.5кН для кута атаки рівним нуль градусів. Отримані результати сили опору у програмі CFD – 5.9 кН, у

створеній програмі – 4.2 кН для кута атаки нуль градусів. Максимальна похибка вимірювань склала – 18%, що є у допустимих нормах.

Висновок до розділу

В даному розділі описано процес розробки програмного забезпечення. Розробка проходила у декілька етапів, головними з яких були:

- Створення класу для задання профілів,
- Вивід профілю за екран користувача,
- Вивід поляри на екран користувача,
- Розрахунок значень та вивід їх на екран.

Було проведено тестування програми, вхідні данні було обрано згідно з технічним завданням. Помилка розрахунків була не більше ніж 20%.

6. СТАРТАП-ПРОЕКТ

Сутність стартап-проекту. Досліджуючи ринок програмного забезпечення, спрямованого на проектування та конструювання літаків було виявлено можливість зробити стартап-проект. Зміст ідеї стартапу та визначення її характеристик наведено в табл. 6.1 та табл. 6.2.

Таблиця 5.1 – Зміст ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Запропонувати систему розрахунку для аеродинамічних характеристик.	1. Продаж додатку	Пришвидшення розрахунків
	2. Продаж розрахунків	Нема необхідності купувати цілий додаток
	3. Бізнес	Автоматизація виробництва

Таблиця 5.2 – Визначення характеристик ідеї стартап-проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропонований метод	Загальноживаний метод			
1.	Продаж програмного забезпечення	Дає змогу	Дає змогу	Може бути економічно затратним	Цінова політика може не задовільнити кінцевого споживача	Можливість зайняти відносно вільну нішу, продукт цікавий для бізнесу та виробництва

2.	Створення нового програмного забезпечення	Дає змогу	Не дає змогу	Потребує детального опрацювання аспектів захисту	Підтримка користувача, оновлення ПО	Можливість комерційної та дотаційної реалізації
----	---	-----------	--------------	--	-------------------------------------	---

У таблиці 5.3 оцінено можливість технологічної реалізації ідеї стартапу та показано технології, які можна застосувати для реалізації проекту.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення нового ПО для надання можливостей розрахунків аеродинамічних характеристик	Спеціалізоване обладнання	Присутня	Доступна
2		Використання існуючих систем	Присутня	Доступна
3		Розробка власних апаратно-програмних рішень	Відсутні на ринку в Україні	Доступна в випадку достатнього бюджету

Обрана технологія реалізації проекту: розробка та продаж програмного забезпечення для обрахунку аеродинамічних характеристик.

У таблиці 5.4 показано попередню характеристику потенційного ринку стартап-проекту.

Таблиця 5.4. Попередня характеристика потенційного ринку стартапу

№ п/п	Показники ринку (найменування)	Характеристика
1	Кількість основних гравців, од	3
2	Обсяг продажів, грн/ум.од	3000
3	Тенденції ринку (якісна оцінка)	Зростає
4	Обмежень для входу (вказати характер обмежень)	Залучення потенційних клієнтів

5	Специфічні вимоги стандартизування та сертифікування	Ліцензія
6	Середня норма рентабельності в даній галузі, %	$3000/2100 = 143\%$

У таблиці 5.5 показано характеристику потенційних клієнтів стартап-проекту

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності поведінки потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Стабільна робота програми	Бізнес, виробництво	Необхідний рівень якості отримуваних даних	Результат повинен відповідати найвищим стандартам якості

У табл. 5.6 наведено основні загрози реалізації стартап-проекту.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Опис загрози	Планове реагування компанії
1	Недостатній інтерес клієнтів	В випадку невдалого маркетингу клієнта можуть не зацікавити запропоновані послуги	Забезпечення додаткових сервісних послуг
2	Втрата конкурентних позицій	Втрата статусу надійного постачальника	Якісний та кількісний приріст інтенсивності та виважена цінова політика

У табл.5.7 наведено основні можливості під час реалізації стартап-проекту.

Таблиця 5.7. Основні можливості

№ п/п	Фактор	Опис можливості	Планове реагування компанії
1	Лідерські позиції на ринку	Стрімке зростання попиту	Якісне та кількісне збільшення продукту, якісна підтримка користувача, постійні оновлення
2	Впровадження запропонованих технологій в уже існуючі системи	Збільшення об'ємів закупівель	Якісне та кількісне збільшення обсягів продукту

У таблиці 5.8 наведено ступеневий аналіз та вплив конкурентного середовища на впровадження проекту.

Таблиця 5.8. Аналіз конкуренції

Особливості конкурентного середовища	Прояв даної характеристика	Вплив на діяльність підприємства (планові дії компанії для забезпечення конкурентоспроможності)
1.Конкуренція	Застосування вже існуючих технологій	Правильний маркетинг
2.Локальний	Відсутність єдиного постачальника послуг	Індивідуальний підхід до кожного клієнта
3.Міжгалузєва	Відсутня	Відсутня
4.Товарно-видова	Використання технологій	Застосування загальноживаних апаратних засобів, за необхідності
5.Цінова	Використання високовартісних спеціалізованих комплексів	Можливість заощадити шляхом застосування

		загальноживаних апаратних засобів
6.Марочна	Кожна ітерація програми повинна бути достатньо точна	Здобуття переваги на ринку

У таблиці 5.9 проаналізовано конкуренцію проекту в галузі за ринком Іспанії

Таблиця 5.9. Аналіз конкуренції ринку

Складові аналізу	Прямі конкуренти	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Програмні постачальники	Потреба пошуку постачальників	Залучення відомих та перевірених постачальників	Самостійність у прийнятті клієнтських рішень	Надання переваги авторитетнішим апаратним рішенням
Висновки:	Висока	Є можливість виходу на ринок	Постачальники встановлюють цінову політику на програмне забезпечення	Клієнти встановлюють вимоги до якості	Обмеження є тільки в випадку відмови

У табл. 5.10 наведено та обґрунтовано фактори конкурентноспроможності.

Таблиця 5.10. Обґрунтування факторів конкурентноспроможності

№ п/п	Фактор конкурентноспроможності	Обґрунтування (чинники, що роблять фактор порівняння конкурентних проектів значущим)
1	Раціональніша цінова політика	Можливість раціональнішого використання ресурсів

У табл. 5.11 перелічено сильні та слабкі сторони проекту.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Порівняння рейтингу товарів- конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Раціональніша цінова політика	15	+						
2	Послуги обслуговування	7					+		
4	Потреба в залученні висококваліфікованих кадрів	5						+	

У табл.5.12 представлений SWOT-аналіз стартап-проекту.

Таблиця 5.12. SWOT-аналіз стартап-проекту

Сильні сторони: раціональна цінова політика	Слабкі сторони: послуги обслуговування, потреба в залученні висококваліфікованих кадрів
Можливості: Ексклюзивне використання програмного забезпечення	Загрози: низька зацікавленість клієнтів, втрата монополії

Альтернативи ринкового впровадження стартапу показані в табл.5.13.

Таблиця 5.13. Альтернативи ринкового впровадження проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність залучення ресурсів	Терміни реалізації
1	Складання договорів з компаніями та оперативне захоплення ринку при використанні нового рішення	висока	короткі

Обґрунтування вибору цільових груп потенційних споживачів показано в табл. 5.14.

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Загальний профіль цільової групи потенційних клієнтів	Готовність сприйняття продукту споживачами	Орієнтовний попит цільової групи (сегменту)	Напруженість конкуренції в сегменті	Складність входу у сегмент
1	Бізнес.	Середня	Високий	Висока	Середня
2	Приватні мережі	Середня	Середній	Середня	Низька

Визначення базової стратегії розвитку наведено у табл. 5.15.

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Основні конкурентоспроможні позиції згідно з обраною альтернативою	Базова стратегія розвитку*
1	Застосування альтернативних способів розрахунку	Впровадження нового стандарту якості	Залучення ключових гравців у галузі	Стратегія диференціації
2	Бюджетність проекту	Оптимізованіші затрати на послуги	Використання загальноживаних апаратних рішень замість спеціалізованих комплексів	Стратегія лідерства по витратах

Визначення основної стратегії конкурентної поведінки показано в табл. 5.16.

Таблиця 5.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект унікальним на ринку?	Чи необхідно буде компанії шукати нових споживачів, чи опрацювати існуючих у конкурентів?	Чи необхідно компанії копіювати основні характеристики товару конкурента?	Стратегія конкурентної поведінки
1	Так	Опрацювати існуючих та шукати нових	Немає необхідності	Стратегія виклику лідера

Визначення стратегії позиціонування показано в табл. 5.17.

Таблиця 5.17. Визначення стратегії позиціонування

№ п/п	Вимоги цільової аудиторії до товару	Основна стратегія розвитку	Основні конкурентоспроможні позиції стартап-проекту	Визначення асоціацій, які сформулюють комплексну позицію стартап-проекту (три основних)
1	Належна висока якість послуг	Стратегія диференціації	Новизна, гарант якості, точність дослідження	Якість, підтримка, надійність
2	Невисокі витрати	Стратегія лідерства по витратах	Універсальність запропонованого рішення	Універсальність, економічна доцільність

Основні переваги концепції потенційного товару показано в табл. 5.18.

Таблиця 5.18. Визначення основних переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Основні переваги перед конкурентами (існуючі або потенційні)
1	Якість	Належна висока якість, надійність	Постійна підтримка користувача, оффлайн підтримка.
2	Невисока вартість	Оптимальне використання коштів	Невисока вартість

Виявлено три рівні моделі товару. Зміст та складові рівнів товару показано в табл. 5.19.

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Зміст та складові		
I. Товар за задумом	Якісний товар та послуги, стандартизована якість послуг та обладнання		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1)Вартість обслуговування, 2)Кількість комплектів програми 3)Технологічна собівартість товару	1) М 2) М 4) 3)М	1)Е 2) Пр 3)Тх

	Якість: міжнародні стандарти, постійне обслуговування та підтримка
	Встановлення і налаштування
	Марка: ДП
III. Товар із підкріпленням	До продажу – програмне забезпечення
	Після продажу – обслуговування та сервісна підтримка

Визначення цінової політики на послугу показано в табл. 5.20.

Таблиця 5.20. Визначення меж встановлення ціни

№ п/п	Цінова політика товарів-замінників	Цінова політика на товари-аналоги	Рівень купівельної спроможності цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	5000 у.о./од. (стандартна методика)	8000 у.о./од.	Високий	Н.4000 у.о. – В.4500 у.о. (Товар) Н.100 у.о. – В.500 у.о. (Послуга)

Створення системи збуту послуги вказано у табл. 5.21.

Таблиця 5.21. Створення системи збуту

№ п/п	Закупівельна поведінка цільових клієнтів	Функції збуту, що повинен забезпечувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Орієнтована на максимальний дохід	Поставки якісного, точного та надійного товару	Значна	Договірна система збуту

Концепції маркетингових комунікацій показано в табл. 5.22.

Таблиця 5.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій цільових клієнтів	Основні методи позиціонування	Завдання рекламного звернення	Концепція рекламного звернення
1	Зацікавленість в точному та якісному продукті з раціональним використанням ресурсів	Мережні ресурси	Гарантія якості та стандартизація, сервісна політика	Привернути увагу до покращень, пов'язаних із зростаючою популярністю послуг	Позиціонування центру синхронізації відправною точкою на шляху до над якісного контенту
2	Зацікавленість у великих об'ємах продукції із дотриманням умов якості	Мережні ресурси	Глибина каналу постачальників, гарантія якості	Привернути увагу до переваг	Позиціонування послуг центру синхронізації єдиним раціональним шляхом забезпечення

Висновок до розділу

Встановлено, що комерціалізація цього проекту як етапу щодо розвитку та застосування програмного комплексу для розрахування аеродинамічних коефіцієнтів можна вважати доцільною. На ринку послуг у світі існує суттєвий попит на данну пропозицію, який зараз задовольняють більш дорогі рішення.

Можливість виходу на ринок є дуже високою, адже користувачам вигідно отримувати максимум доходу від придбаного програмного

забезпечення, і водночас ця сфера є дуже перспективною та високодохідною в майбутньому.

Обрана альтернатива впровадження – пошук альтернативних технологій та алгоритмів розрахунку аеродинамічних систем.

ВИСНОВКИ

Розрахунки аеродинамічних навантажень завжди мають наближений характер. Однак вони дозволяють здійснити якісну оцінку компонування літального апарату та оптимізувати його з метою досягнення найменших витрат палива, забезпечити стійкість та безпечність польоту. Подальше уточнення аеродинамічних коефіцієнтів літального апарату обраного компонування здійснюється на основі експериментів із його напівнатурною моделлю. Однак вже накопичена велика кількість експериментальних даних для стандартних профілів крил літаків. Ці експериментальні дані дозволяють вверіфікувати широкорозповсюджені програми для розрахунків аеродинамічних навантажень. Зазначені програми базуються на методі кінцевих елементів, панельно вихоровому методі, тощо. Експериментальні дані зазвичай представлені у вигляді графіків залежності коефіцієнтів підйомної сили від кута атаки та поляри при визначених числах Рейнольдса.

В ході даної магістерської дисертації було розроблено графічний інтерфейс користувача, та набір формул, які за допомогою вхідних геометричних та фізичних даних розраховують аеродинамічні характеристики літака. З метою уточнення та перевірки отриманих даних було здійснено моделювання літака типу А-32 компанії AEROPRAKT в програмі ProE WF5. За допомогою програмного забезпечення Simulation CFD були розраховані аеродинамічні навантаження, та порівнянні з отриманими у створеному програмному комплексі. Для кута атаки нуль градусів, значення підйомної сили розраховане за допомогою створеної програми – 28,5 кН, для програми Simulation CFD – 27,4 кН. Помилка розрахунків склала не більше ніж 20%.

Встановлено, що комерціалізація цього проекту як стратапу щодо розвитку та застосування програмного комплексу для розрахування аеродинамічних коефіцієнтів можна вважати доцільною. На ринку послуг у світі існує суттєвий попит на данну пропозицію, який зараз задовольняють більш дорогі рішення.

Можливість виходу на ринок є дуже високою, адже користувачам вигідно отримувати максимум доходу від придбаного програмного забезпечення, і водночас ця сфера є дуже перспективною та високодохідною в майбутньому.

Використання розробленого програмного комплексу разом із програмами кінцево-елементних розрахунків дозволяє швидко і з достатньою точністю проводити розрахунки літальних апаратів будь якої аеродинамічної форми. Запропонований метод побудови графічного інтерфейсу може бути використаний розробниками літальних апаратів для верифікації своїх розрахунків та швидкого підбору компоновки і геометричних параметрів літака. Оскільки графічний інтерфейс користувача написаний у відкритому коді Сі#, то дозволяє швидко його модифікувати та наповнювати статистичними даними інших профілів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. С. Т. Кашафутдинов В.Н. Лушин. Атлас аэродинамических характеристик крыловых профилей / Сибирский научно-исследовательский институт авиации им. С. А. Чаплыгина, 1994. – 76с.
2. Прицкер Д.М., Сахаров Г.И. Аэродинамика. - М.: Машиностроение, 1968. – 310 с.
3. Джон Скит. С#. Программирование для профессионалов. / Издательский дом «Вильямс», 2014 – 598 с.
4. Авиационная энциклопедия: Уголок Неба. Аэропракт А-32. [Электронный ресурс]. – Режим доступа: <http://www.airwar.ru/enc/la/a32.html> - 10.07.2018.
5. AEROPRAKT. Development and serial production of ultra-light aircraft of various modifications. А-32 – AEROPRAKT. [Электронный ресурс]. – Режим доступа: <https://aeroprakt.kiev.ua/aircraft/a-32/>
6. Simulink – Simulation and Model-Based Design. MATLAB & Simulink. [Электронный ресурс]. – Режим доступа: <https://uk.mathworks.com/products/simulink.html>
7. Siemens Digital Industries Software. Simcenter Femap. [Электронный ресурс]. – Режим доступа: <https://www.plm.automation.siemens.com/global/ru/products/simcenter/femap.html>
8. Научно-технический вестник информационных технологий, механики и оптики, том 18, № 2. 2018. – 169с. – 190с.
9. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / за заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 29 с.
10. Расчет конструкций в MSC/NASTRAN for Windows. Шимкович Д.Г. – Москва, 2003. – 444с.
11. Autodesk CFD [Электронный ресурс] : Autodesk CFD – функционал, полное описание. – Режим доступа: <https://www.pointcad.ru/product/autodesk-simulation-cfd/funkczional-autodesk-cfd>

ДОДАТОК 1

Лістинг програми

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace DP
{
    public partial class Form1 : Form{
        public Form1()
        {
            InitializeComponent();
            InitPlot();
            CreateProFileChart(prof1);
            lamComboBox.SelectedIndex = 0;
        }

        public profileData prof1 = new profileData("K-3" , 0.164,
            new[] { 0, 1, 2, 3, 5, 7.5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 95, 100},
            new[] { 0, 3.17, 4.37, 5.28, 6.66, 7.94, 9.00, 10.31, 11.03, 11.32, 11.03,
                9.72, 7.86, 5.56, 3.57, 2.18, 1.08, 0.54, 0},
```

```

        new[] { 0, -1.98, -2.75, -3.17, -3.69, -4.06, -4.29, -4.72, -5.02, -5.12, -5.16,
-5.24, -5.13, -4.76, -3.89, -2.78, -1.51, -0.79, 0},
        new[] { -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15},
        new[] { -0.3, -0.22, -0.15, -0.08, 0, 0.05, 0.15, 0.2, 0.24, 0.32, .4, .48, .54,
.61, .7, .77, .84, .93, 1, 1.07, 1.12},
        new[] { 0.015, .013, 0.011, 0.01, .008, .008, .011, 0.012, 0.015, 0.017,
0.021, 0.025, 0.03, 0.035, 0.046, 0.05, 0.06, 0.07, 0.081, 0.095, 0.11}
    );

```

```

    public profileData prof2 = new profileData("NASA 43012A", 0.122,
        new[] { 0, 1.25, 2.50, 5.00, 7.50, 10.00, 15.00, 20.00, 25.00, 30.00, 40.00,
50.00, 60.00, 70.00, 80.00, 90.00, 95.00, 100.00},
        new[] { 0, 3.87, 5.19, 7.02, 8.17, 8.96, 9.62, 9.65, 9.43, 9.14, 8.47, 7.52,
6.35, 5.00, 3.52, 1.90, 1.04, 0},
        new[] { 0, -0.81, -1.09, -1.44, -1.71, -1.92, -2.28, -2.57, -2.79, -2.93, -3.15,
-3.07, -2.77, -2.31, -1.72, -1.00, -0.59, 0},
        new[] { -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 },
        new[] { -0.15, -0.09, -0.05, 0.03, 0.09, 0.13, 0.22, 0.28, 0.35, 0.41, 0.5, 0.56,
0.63, 0.7, 0.8, 0.85, 0.91, 0.96, 1.04, 1.1, 1.15},
        new[] { 0.03, 0.017, 0.015, 0.011, 0.01, 0.011, 0.012, 0.014, 0.017, 0.021,
0.025, 0.03, 0.037, 0.044, 0.055, 0.062, 0.071, 0.076, 0.083, 0.1, 0.11}
    );

```

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    Item item = comboBox1.SelectedItem as Item;
    CreateProFileChart(item.Value as profileData);
}

```

```

public class profileData
{
    public string name;

```

```

public double[] XData;
public double[] YtopData;
public double[] YbotData;
public int[] alfData;
public double[] CyData;
public double[] CxData;
public double cv;

public profileData(string profileName, double C, double[] X, double[]
Ytop, double[] Ybot, int[] alf, double[] Cy, double[] Cx)
{
    this.name = profileName;
    XData = X;
    YtopData = Ytop;
    YbotData = Ybot;
    alfData = alf;
    CyData = Cy;
    CxData = Cx;
    cv = C;
}

public double[] X
{
    get => XData;
}

public double C
{
    get => cv;
}

public double[] Ytop
{

```



```

        get => YtopData;
    }
    public double[] Ybot
    {
        get => YbotData;
    }
    public string Name
    {
        get => name;
    }
    public double[] Cy
    {
        get => CyData;
    }
    public double[] Cx
    {
        get => CxData;
    }
    public int[] Alf
    {
        get => alfData;
    }
}

```

```

public float bk, b0, Lwing, LAMwing, Lfus, Smid, Dmax, LAMfus, v, al, Vst;

```

```

private void label5_Click(object sender, EventArgs e)
{

```

```
}
```

```
public class Item
```

```
{
```

```
    public Item() { }
```

```
    public object Value { set; get; }
```

```
    public string Text { set; get; }
```

```
}
```

```
public void InitPlot()
```

```
{
```

```
    List<Item> profiles = new List<Item>();
```

```
    profiles.Add(new Item() { Text = prof1.Name, Value = prof1 });
```

```
    profiles.Add(new Item() { Text = prof2.Name, Value = prof2 });
```

```
    comboBox1.DataSource = profiles;
```

```
    comboBox1.DisplayMember = "Text";
```

```
    comboBox1.ValueMember = "Value";
```

```
}
```

```
public void CreateProFileChart(profileData obj)
```

```
{
```

```
    chart1.Series.Clear();
```

```
    var topPart = new Series();
```

```
    var botPart = new Series();
```

```
    topPart.Points.DataBindXY(obj.X, obj.Ytop);
```

```
    chart1.Series.Add(topPart);
```

```
    topPart.ChartType = SeriesChartType.Spline;
```

```
topPart.Color = Color.Black;
```

```
botPart.Points.DataBindXY(obj.X, obj.Ybot);
```

```
chart1.Series.Add(botPart);
```

```
botPart.ChartType = SeriesChartType.Spline;
```

```
botPart.Color = Color.Black;
```

```
chart2.Series.Clear();
```

```
var pol = new Series();
```

```
pol.Points.DataBindXY(obj.Cx, obj.Cy);
```

```
chart2.Series.Add(pol);
```

```
pol.ChartType = SeriesChartType.FastLine;
```

```
pol.Color = Color.Black;
```

```
double[] alfP = new double[] { -0.025, -0.02, -0.015, -0.01, -0.005, 0,  
0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05, 0.055, 0.06, 0.065,  
0.07, 0.075};
```

```
var zav = new Series();
```

```
zav.Points.DataBindXY(alfP, obj.Cy);
```

```
chart2.Series.Add(zav);
```

```
zav.ChartType = SeriesChartType.Spline;
```

```
zav.Color = Color.Black;
```

```
}
```

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

}
```

```
private void label1_Click(object sender, EventArgs e)
{

}
```

```
private void label2_Click(object sender, EventArgs e)
{

}
```

```
private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{

}
```

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        bk = float.Parse(bkTextBox.Text);
        b0 = float.Parse(b0TextBox.Text);
        Lwing = float.Parse(LwingTextBox.Text);
        Vst = float.Parse(textBox2.Text);
    }
}
```

```
Lfus = float.Parse(LfusTextBox.Text);
Dmax = float.Parse(DmaxTextBOx.Text);
v = float.Parse(VTextBox.Text);
al = float.Parse(lamComboBox.Text);
```

```
Item prof = comboBox1.SelectedItem as Item;
profileData itemasd = prof.Value as profileData;
```

```
int index = lamComboBox.SelectedIndex;
```

```
//Знаходження Cx та Cy для даного куту атаки
```

```
al = itemasd.Alf[index];
var CxProf = itemasd.Cx[index];
var CyCur = itemasd.Cy[index];
```

```
//-----
```

```
//Обрахування видовження крила та коефіцієнту індуктивного  
опору
```

```
var LamK = (Lwing / 2) / ((b0 + bk) / 2);
var Cxi = CyCur * CyCur / (Math.PI * LamK);
var Cxs = CxProf + Cxi;
```

```
//-----
```

```
//Знаходження площі крила
```

```
var b = Math.Sqrt(Math.Pow((Lwing/2), 2) + Math.Pow((b0 - bk), 2));
var Sk = (b + Lwing / 2) * b0 * 2;
```

```
//-----
```

//знаходження площі поперечного перерізу фюзеляжу та площа поперечного перерізу крила

```
var Sfus = Math.PI * Dmax * Dmax / 4;
```

```
var c = itemasd.C;
```

```
var Skp = c * (b0 + bk) * Lwing / 2;
```

```
//-----
```

//знаходження коефіцієнту впливу площі фюзеляжу та отримання коефіцієнту повного опору крила

```
//var k = (Skp + Sfus) / Skp;
```

```
var CxCur = Cxs;
```

```
//-----
```

//знаходження підйомної сили та сили опору

```
var ro = 1.25;
```

```
var Y = CyCur * ro * Math.Pow(v, 2) * Sk / 2;
```

```
var X = CxCur * ro * Math.Pow(v, 2) * Sk / 2;
```

```
//-----
```

```
var alf = (al - Vst) * Math.PI / 180;
```

```
var cxf = 0.4;
```

```
var cp = 2.5;
```

//Знаходження площі повздовжнього перетину фюзеляжу та опору та підйомної сили фюзеляжу

```
var LamFus = Lfus / Dmax;
```

```

var SD = Lfus * Dmax * Math.PI / 4;
var Fxp = cxf * Sfus * ro * v * v / 2 * Math.Cos(alf);
var Falfa = Math.Abs(cp * Math.Sin(alf) * SD * ro * v * v / 2);
var Fxf = Fxp + Falfa;
var Fyf = cp * Math.Sin(alf) * Math.Cos(alf) * SD * ro * v * v / 2;
//-----

```

```

//Знаходження загальної підйомної сили та сили опору літаки
var Yk = Y + Fyf;
var Xk = X + Fxf;
//-----

```

```

//Знаходження потрібної потужності
var m = float.Parse(textBox1.Text);
var K = Yk / Xk;
var P = m * 9.81 / K;
var N = P * v;
//-----

```

```

richTextBox1.Text = "Підйомна сила Y = " + Math.Round(Yk,
2).ToString() + "\n";
richTextBox1.Text += "Сила опору X = " + Math.Round(Xk,
2).ToString() + "\n";
richTextBox1.Text += "Потрібна потужність N = " + Math.Round(N,
2).ToString() + "\n";

```

```

}

```

```
        catch {  
            richTextBox1.Text = "Значення введені невірно! \n" + "Введіть вірні  
значення!";  
        }  
    }  
}
```