

Що таке обчислювальна інженерія?

Обчислювальна інженерія пропонує принципово інший підхід до розробки продукту, замінюючи ручне креслення та візуальне моделювання алгоритмічною, багаторазово використовуваною логікою проєктування. Вона відкриває ряд ключових переваг порівняно з традиційними методами конструювання. Замість створення єдиного креслення за допомогою ручного CAD-моделювання інженери пишуть алгоритми, які кодують увесь процес проєктування для певного класу об'єктів. У результаті виходить не просто одна деталь, а система, здатна генерувати безліч допустимих варіантів проєктування, кожен із яких ґрунтується на загальній інженерній логіці.

Кожен об'єкт, створений за допомогою обчислювальної інженерії, робить свій внесок у кодову базу платформи, збагачуючи знання про проєктування для майбутніх ітерацій. Це створює замкнене коло, в якому кожен проєкт підвищує можливості, гнучкість і складність системи. Чим більше об'єктів ви створюєте, тим розумнішою стає ваша платформа проєктування.



LEAP 71 перебуває на передовій цього нового підходу. Її розробки допомогли визначити та просунути сферу обчислювальної інженерії, продемонструвавши, що складні технічні виробни, такі як ракетні двигуни, можна створити безпосередньо програмою, яка кодує фізичні процеси, виробничі обмеження та інженерну логіку.

Для сприяння впровадженню та співпраці LEAP 71 випустила **PicoGK**, базове геометричне ядро, як відкритий початковий код.

Інженер-програміст починає з того, що розбиває проєкт на основні складові, визначаючи, як ці компоненти взаємодіють через чіткі, логічні залежності.

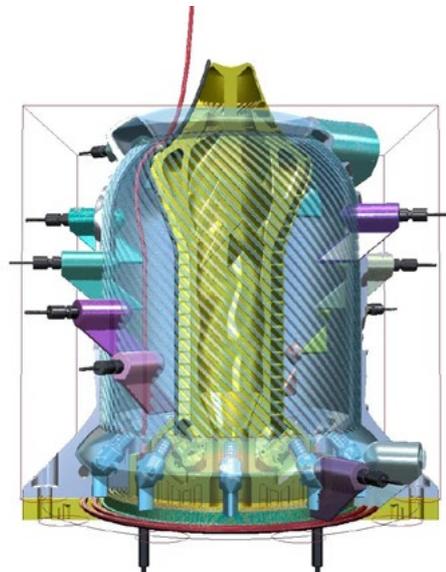
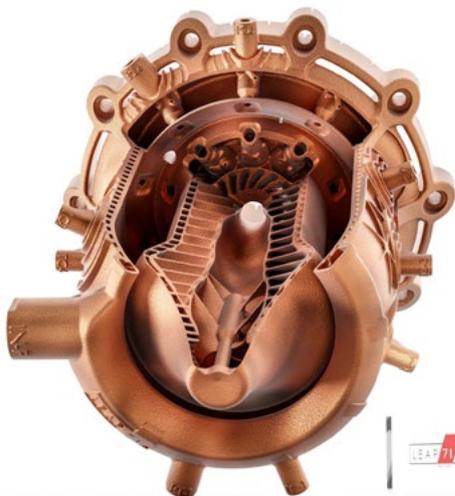
На цьому етапі акцент робиться не на естетиці, а на виявленні основних правил: функціональних вимог, фізичних обмежень, технологічності виготовлення та критеріїв продуктивності.

Наступний крок — кодування логіки побудови, тобто відповідь на питання: враховуючи ці обмеження та вхідні дані, як би експерт у цій галузі підійшов до проєктування? Ця початкова реалізація може створювати просту геометрію-заповнювач, але вона закладає структуру, на основі якої можна будувати більш складні алгоритми.

У процесі ітерацій обчислювальна модель удосконалюється, поступово впроваджуючи вищі рівні інтелектуального проєктування. Інженери перевіряють результати в широкому діапазоні параметрів, використовуючи візуальний огляд, фізичні випробування та, дедалі частіше, автоматизоване чисельне моделювання.

У результаті виходить жива обчислювальна модель: програмний код, який приймає параметри та виробничі обмеження як вхідні дані та видає готові до виготовлення проєкти у вигляді файлів, придатних для прямого друку або механічної обробки.

Ця модель може бути розгорнута у великих масштабах, повторно використана, розширена та поліпшена іншими інженерами. Вона стає не просто проєктом, а масштабованою платформою, що розвивається, та новою формою інтелектуальної власності: динамічною, кодифікованою базою знань для створення фізичних продуктів.



ГНУЧКІСТЬ КОНСТРУКЦІЇ

Обчислювальні моделі використовують усю міць сучасних мов програмування, що дозволяє здійснювати складну параметризацію, значно перевищуючи можливості традиційних систем САПР. Проекти можуть самоадаптуватися до змінюваних вхідних умов — геометрично змінюючи свою форму відповідно до змінюваних вимог без потреби ручного доопрацювання.

Це зміщує роль інженера з трудомісткого створення моделей на дослідження та оптимізацію. Інженери можуть вільно досліджувати нові конфігурації, будучи впевненими, що логіка проєктування впорається з деталями реалізації.



У наведеному на рисунку прикладі — один із варіантів конструкції головки інжектора для камери згоряння ракетного двигуна. Форма автоматично адаптується до конкретних обмежень виробничого процесу, таких як габарити робочої зони, характерні для промислових 3D-принтерів для адитивного виробництва, при збереженні важливих функціональних особливостей.

УДОСКОНАЛЕННЯ МОДЕЛІ ПРОЄКТУВАННЯ

У традиційному машинобудуванні складність і деталізація проєкту безпосередньо залежать від обсягу ручної роботи та часу, доступного для цього проєкту. Часто досягти більш високого рівня деталізації неможливо через обмеженість ресурсів.

Обчислювальна інженерія розриває цю залежність. Замість того, щоб витратити час на проєктування окремих деталей, інженери використовують його для розробки та вдосконалення самої моделі проєктування — багаторазово використовуваної кодової бази, яка з кожною ітерацією генерує все більш складні результати.

Прості алгоритми можуть породжувати надзвичайно складні структури, які було б важко або неможливо створити вручну.

Приклади розробок LEAP 71:

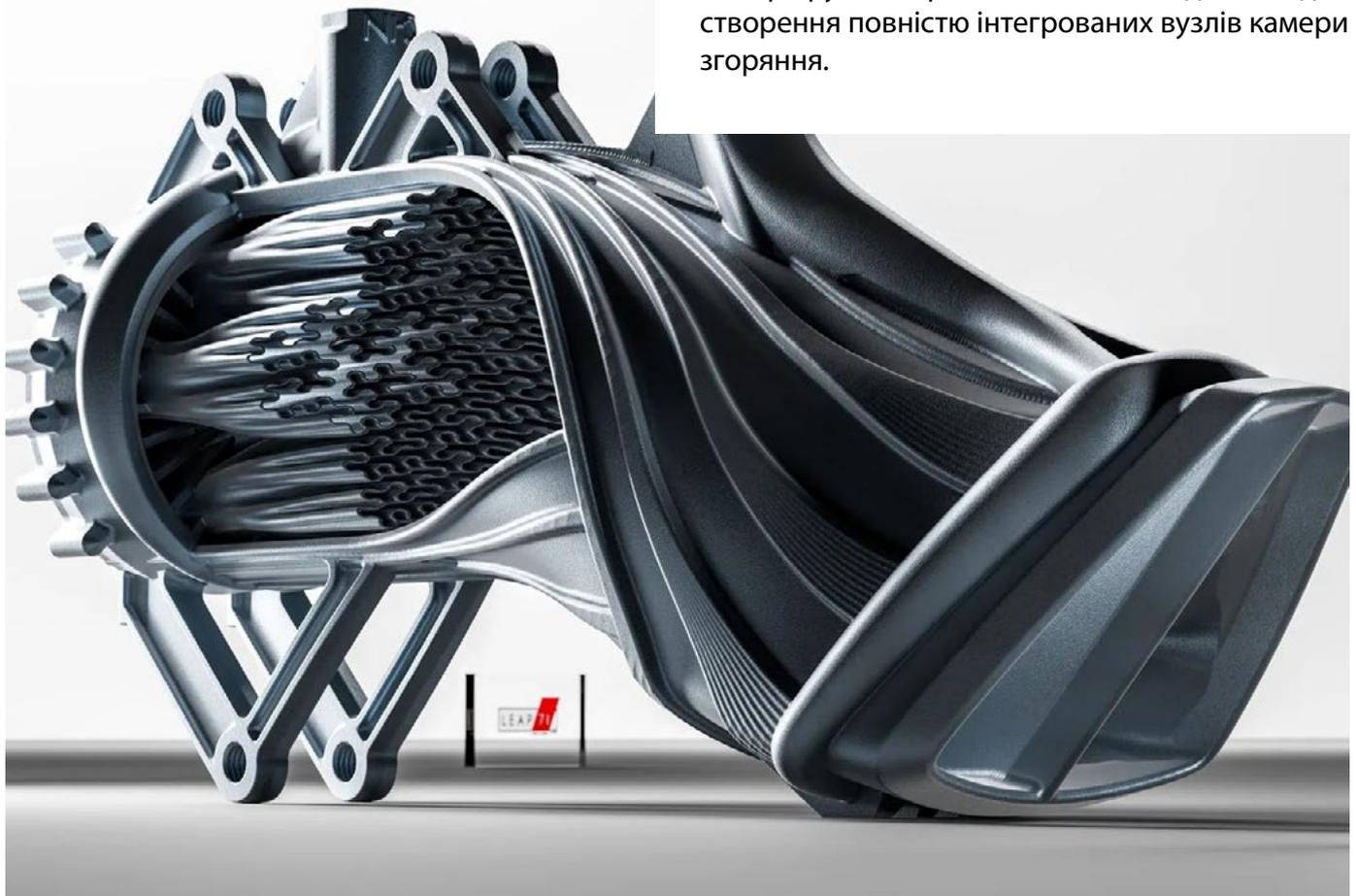
- Автоматизована прокладка каналів регенеративного охолодження.
- Параметрична побудова комплексних різноманіт.



<https://youtu.be/QatsryWonn0>
https://youtu.be/vfq_MurNaXA

- Створення складних текстур поверхні для сучасних теплообмінників.

Завдяки можливості багаторазового використання та компонування цих логічних модулів інженери можуть розвивати наявні рішення — наприклад, комбінуючи колекторний генератор із маршрутизатором каналів охолодження для створення повністю інтегрованих вузлів камери згоряння.



ЕКСПЕРИМЕНТУВАННЯ

Інновації процвітають завдяки експериментам — розширенню меж, перевірці ідей та ітеративному вдосконаленню, поки не з'явиться щось принципово нове. У традиційній інженерії цей процес повільний і дорогий: кожна нова ітерація вимагає ручної переробки, що часто забирає значний час і зусилля. В обчислювальній інженерії експериментування стає обчислювальним процесом. Зміна вхідних параметрів та перерахунок моделі можуть привести до появи нових варіантів конструкції за секунди або хвилини, залежно від складності моделі та використовуваного геометричного ядра. Вартість експериментування зміщується від людських зусиль до обчислювальних потужностей.

У поєднанні з чисельним моделюванням цей підхід відкриває повністю автоматизовані цикли зворотного зв'язку, у межах яких можна генерувати, оцінювати та оптимізувати тисячі варіантів конструкції — без ручного втручання.



Це дозволяє широко досліджувати простір проектування, даючи інженерам можливість знаходити рішення, які були б непрактичними або неможливими при використанні традиційних методів. 