ПОБУДОВА СТРУКТУРИ САПР

Загалом, стандартна САПР має три підсистеми: препроцесор, процесор і постпроцесор результатів.

Препроцесор відповідає за визначення моделі задачі. Зазвичай модель задачі включає опис форми об’єкта, впливів зовнішнього середовища (наприклад, прикладених сил, температур, вібрацій). Форму досліджуваного об’єкта описують за допомогою однієї з схем подання. Під схемою подання розуміють семантично та синтаксично коректне відношення між множинами формальних описів і допустимих форм об’єктів. Процесор є ядром підсистеми інженерного аналізу.

Процесор – це множина чисельних методів, які зазвичай реалізуються з використанням високопродуктивних обчислень. Можливо, найбільш поширеним чисельним методом є метод скінченних елементів. У цьому методі неперервна модель замінюється дискретною. Відповідно, перший крок такого моделювання потребує дискретизації неперервної геометричної області на елементи простої форми, наприклад, трикутники, чотирикутники, тетраедри та шестигранники. Поширені альтернативи методу скінченних елементів (наприклад, безсіткові методи) можуть так само бути використані процесором. Ці методи не вимагають зв’язків між вузлами у моделі форми об’єкта. Проте, вузлова дискретна модель все ще необхідна. Отже, генерація дискретних моделей є важливою складовою, яка з’єднує опис форми об’єкта з його дискретною моделлю.

Після застосування чисельного аналізу на основі значень, отриманих у вузлах дискретної моделі, постпроцесор обчислює похідні величини та візуалізує

результати. Тобто, постпроцесор створює подання дискретної моделі. Також значення, отримані у вузлах дискретної моделі, можна використовувати для поліпшення обох вихідної (неперервної) та проміжної (дискретної) моделей. Отже, абстракціями верхнього рівня при розробці підсистем інженерного аналізу є препроцесор, процесор і постпроцесор.

Таким чином, як і будь яку САПР, САПР на основі методу скінченних елементів для розрахунку конструкцій можна представити у вигляді трьох взаємопов'язаних послідовних процесів:

1) підготовка вихідних даних – опис топології конструкції, кінематичних та силових граничних умов, фізико-механічних характеристик композиційного матеріалу, скінченно-елементна дискретизація конструкції та інших;

2) чисельний розрахунок скінченно-елементної моделі – обчислення коефіцієнтів матриці жорсткості скінченних елементів, формування глобальної системи розв'язувальних рівнянь та її розв'язок;

3) обробка результатів розв'язання – обчислення параметрів напружено-деформованого стану конструкції; їх візуальне представлення у вигляді таблиць, графіків, двовимірних або тривимірних зображень.

Ці процеси при програмній реалізації традиційно виконуються трьома підсистемами – *препроцесором*, *процесором* та *постпроцесором*, відповідно.

Структуру стандартної САПР можна представити у вигляді укрупнених основних блоків та їх взаємодії в процесі визначення напружено-деформованого стану конструкцій (рис. 1).

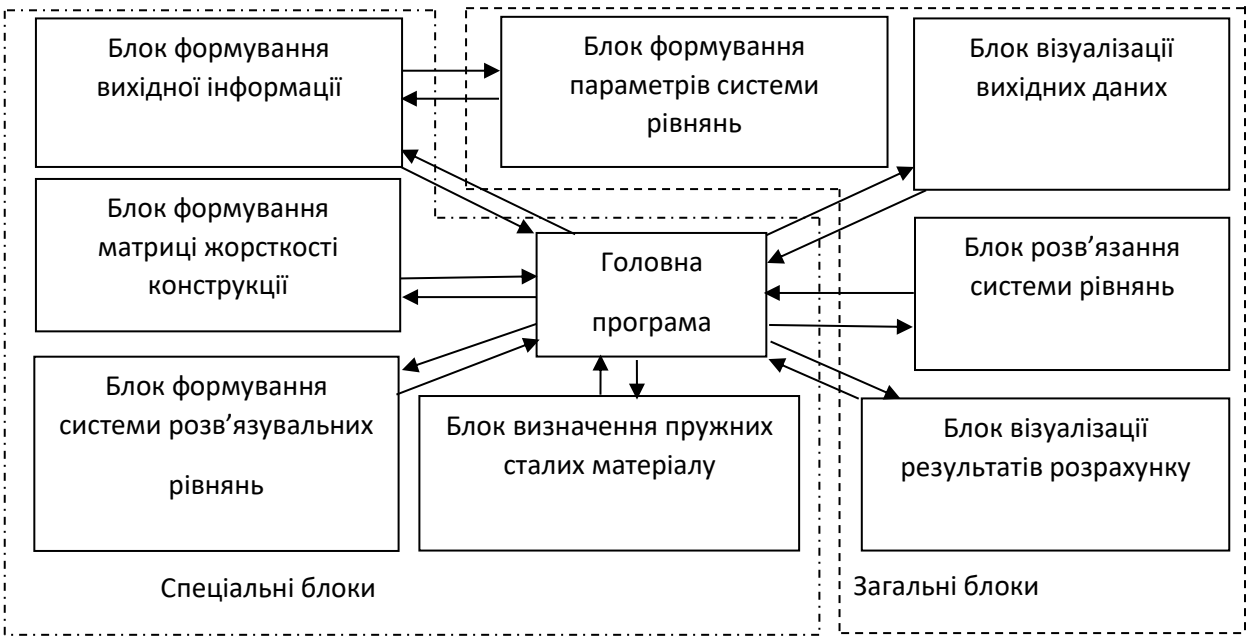

Рисунок 1 – Структура стандартної САПР для визначення напружено-деформованого стану конструкцій

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ САПР

Завдання, з якими зустрічається сучасне програмне забезпечення, все більше ускладнюються. В основу обчислювальних модулів закладаються різні підходи та методи. Як результат суттєво ускладнюються розробка та підтримка. Можливим способом спрощення розробки та підтримки такого програмного забезпечення є не тільки розробка універсальних узагальнених методів, а й типових програмних рішень, які можуть використовуватися повторно.

Шаблони проєктування для об'єктно-орієнтованих САПР

Шаблони проєктування як загальні багаторазово використовувані рішення вперше були розроблені в кінці 1980-х років і з того часу їх активно використовують в інженерії програмного забезпечення. До недавнього часу розробники наукомісткого програмного забезпечення зазвичай уникали об'єктноорієнтованого підходу через його накладні обчислювальні витрати. Проте, програмне забезпечення САПР стає більшим і складнішим. Виникають питання щодо його гнучкості, розширюваності та супроводу. Шаблони програмного забезпечення розв'язують ці задачі шляхом надання загальних об'єктно-орієнтованих рішень. Архітектори, механіки, винахідники та інженери для розв'язання інженерних задач використовують підсистеми інженерного аналізу (Computer-Aided Engineering, CAE) САПР. Підсистеми інженерного аналізу є наукомісткими частинами САПР, які можуть включати модулі дослідження напружено-деформованого стану, поширення температури, динаміки твердих тіл, газів і рідин. Розробка шаблонів проєктування дозволить об'єднати усі абстракції, що виникають при розробці САПР, в єдину об'єктно-орієнтовану модель.

Декомпозиція та ідентифікація

Зазвичай, підсистеми інженерного аналізу САПР та їх базис чисельних методів розробляють з використанням процедурного підходу та таких мов програмування як FORTRAN, C тощо. Процедурний підхід вельми ефективний у реалізації чисельних методів, включаючи метод скінченних елементів. Проте, підтримка САПР включає оновлення чисельних моделей, додавання нової функціональності та підвищення продуктивності програмного коду. Розглянемо підсистему САПР, що ґрунтується на методі скінченних елементів. Можна припустити, що модель кожного об’єкта спочатку описана за допомогою однієї зі схем подання, а також побудована відповідна дискретна модель його форми. Виконаємо декомпозицію САПР з використанням аналізу загальностей та частковостей. Деякі результати ідентифікації за схемою «абстракція (конкретні реалізації) такі: Схема подання (граничне подання; конструктивна блокова геометрія; функціональне подання тощо); Точка (двовимірна точка; тривимірна точка); Вузол (двовимірний вузол; тривимірний вузол); Елемент (сегмент; трикутник; чотирикутник; тетраедр; шестигранник); Дискретна модель (двовимірна дискретна модель; тривимірна дискретна модель) тощо.

Каталог шаблонів проєктування

Шаблон проєктування «UI–Model–Analysis»

Сучасні програмні підсистеми інженерного аналізу САПР мають інтегрований графічний інтерфейс користувача (UI). Використовуючи контролери графічного інтерфейсу користувач вводить модель задачі (Model) та ініціює її аналіз процесором (Analysis). Отже, на верхньому рівні абстракції три основних пакета беруть участь у базовому шаблоні «UI–Model–Analysis». Метою цього шаблона проєктування є декомпозиція підсистеми інженерного аналізу на

високорівневі частини, а також відділення класів, які пов’язані з графічним інтерфейсом від класів моделі та процесора.

Пакет UI (графічний інтерфейс) містить класи, які реалізують пов’язані з графічним інтерфейсом функції програмного забезпечення. Ці класи об’єднують функції пре- і постобробки (наприклад, вводу даних і візуалізацію). Отже, не повинно бути зв’язності між підсистемами аналізу та графічного інтерфейсу. Проте, пакет UI залежить від пакетів Model (препроцесор) і Analysis (процесор). Загалом модель визначають за допомогою деякого проблемно-орієнтованої мови. Проблемно-орієнтована мова дозволяє описати форму об’єкта та параметри задачі (фізико-механічні характеристики, впливи зовнішнього середовища). Форму об’єкта описують на базі певної схеми подання. Абстрактна схема подання (Representation) є інтерфейсом, який дозволяє перевірити належність точки до моделі об’єкта. Отже, схема подання є однією з ключових абстракцій пакета Model. Пакет Analysis (процесор) відповідає за чисельний аналіз. Процедури аналізу, зазвичай, виконують за допомогою метода скінченних елементів. У цьому методі два основних етапи: генерація дискретної моделі та безпосередньо скінченноелементний аналіз. Отже, дискретна модель (Mesh) і процесор задач метода скінченних елементів (FEA Problem) є ключовими абстракціями цього пакета.

Шаблон проєктування «Representation–Mesh»

Припустимо, що форма об’єкта описана у термінах деякої схеми подання. Найбільш поширеними схемами подання є граничне подання (BRep), конструктивна блокова геометрія (CSG) і функціональне подання (FRep). Спільною властивістю конструктивної блокової геометрії та функціонального подання є можливість легкої перевірки приналежності точки до моделі об’єкта. Водночас граничне подання загалом потребує перевірки парності перетинів променю з точки і межі моделі об’єкта. Тим не менш, спільною поведінкою

відповідних класів можна вважати класифікацію точок щодо межі об’єкта. Для генерації дискретної моделі застосовуються певні методи розбиття частини евклідового простору на елементи простої форми. Можна припустити, що дискретна модель – це абстрактний інтерфейс (Mesh), який дозволяє будувати та ітерувати через колекцію елементів. У такому випадку конкретні класи будуть похідними від абстрактного класу Mesh. Конкретні класи генерують елементи певної форми, використовуючи інтерфейс абстракції схеми подання (Representation) для класифікації точок простору. Обидва класи Representation і Mesh беруть участь у шаблоні проєктуваня «Representation–Mesh». Метою цього шаблону проєктування є розподіл обов’язків між класами подання форм об’єктів і генерації дискретних моделей.

Шаблон проєктування «Element–Node»

Загалом елемент дискретної моделі (Element) – це впорядкована колекція вузлів (Node). У обох (двовимірному та тривимірному) випадках елементи мають ребра. Ребро (Edge) – це відрізок прямої, що з’єднує два вузли. Тривимірні елементи на додаток до ребер мають грані. Грань (Face) – це плоский елемент, обмежений ребрами. Тобто, конкретна грань – це об’єкт класу, який реалізує інтерфейс Element. Отже, Element, Face, Edge і Node – структурні частини шаблону проєктування «Element–Node». Метою цього шаблону проєктування є розробка об’єктної декомпозиції для визначення елементів.

Клас Node – це абстрактний інтерфейс, який визначає спільну для всіх типів вузлів поведінку (наприклад, маніпуляція типами вузлів, управління множиною інцидентних елементів тощо). Можуть бути різні реалізації для конкретних класів вузлів. Наприклад, операції зберігання координат і геометричні операції над ними можна реалізувати безпосередньо в класі. Альтернативно, клас вузла може бути композицією об’єктів. У такому випадку будуть отримані паралельні ієрархії для класів, що реалізують операції над вузлами і векторами. Клас Element

і його похідні (включно грані) можна реалізувати з використанням шаблона проєктування «Ітератор» для організації проходу по колекціям вузлів, ребер і граней. Шаблон «Ітератор» також можна використати для реалізації роботи зі списками інцидентних елементів.

Шаблон проєктування «FEA Problem»

Спочатку метод скінченних елементів був розроблений для дослідження напружено-деформованого стану, а потім застосований у розв'язанні інших задач, що ґрунтуються на рівняннях у частинних похідних. У інженерних задачах, зазвичай, необхідно визначити розподіл значень деякої невідомої величини (наприклад, переміщень у задачах дослідження напружено-деформованого стану). У методі скінченних елементів розподіл значень невідомої величини апроксимують кусково-лінійними функціями на кожному елементі. Як результат отримують систему лінійних алгебраїчних рівнянь, яка дозволяє обчислити значення невідомої величини. З математичної точки зору кожен елемент можна подати матрицями та векторами. Операції ансамблювання дозволяють трансформувати значення з локальних матриць і векторів у глобальні. Вихідною інформацією у процесі анасамблювання є номери вузлів і зв'язки між ними. Перед розв'язанням системи лінійних алгебраїчних рівнянь повинні бути враховані крайові умови. Вони спрощують глобальні матрицю і вектор шляхом виключення деяких невідомих. Якщо крайова умова застосована до деякого вузла, то виключаються рядок і стовпець глобальної матриці, які визначені глобальним номером вузла і напрямком крайової умови. Локальні та глобальні вектори обчислюються при врахуванні сил, що діють в вузлах елементів. Аналогічно крайовим умовам, конкретна позиція вузлового навантаження обчислюється на базі глобального номера вузла та напрямку, в якому це навантаження діє.

Метою цього шаблону проєктування є визначення структури алгоритму скінченно-елементного аналізу та об’єктної композиції для крайових умов та сил, які беруть участь в задачі. Сили та крайові умови реалізують інтерфейс, що дозволяє отримати напрям дії та значення в довільній точці. Крайові умови та сили зазвичай визначаються користувачем з використанням графічного інтерфейсу або проблемно-орієнтованої мови.

Розроблені вище шаблони основних логічних елементів програмної підсистеми інженерного аналізу САПР пропонують схеми об’єктної декомпозиції та розподілу обов’язків без обмежень на реалізацію. Розподіл обов’язків між відносно незалежними блоками дозволяє спростити колективну розробку та підтримку САПР як складного програмного продукту.