

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Інженерно-енергетичний факультет

Кафедра агроінженерії

СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ

Конспект лекцій

для здобувачів першого (бакалаврського) рівня вищої освіти
ОПП «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки»
денної форми здобуття вищої освіти

МИКОЛАЇВ
2025

УДК 004.422.8
С34

Друкується за рішенням науково-методичної комісії інженерно-енергетичного факультету Миколаївського національного аграрного університету від 24.02.2025, протокол № 6.

Укладачі:

О. С. Садовий – канд. техн. наук, доцент кафедри агроінженерії. Миколаївський національний аграрний університет.

Рецензенти:

С. І. Тищенко – канд. пед. наук, доцентка кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій. Миколаївський національний аграрний університет.

Д. Л. Кошкін – канд. техн. наук, доцент кафедри електроенергетики, електротехніки та електромеханіки. Миколаївський національний аграрний університет.

ЗМІСТ

Лекція 1. Введення в автоматизоване проектування.....	3
Лекція 2. Рівні проектування.....	7
Лекція 3. Проектні процедури.....	11
Лекція 4. Технічне забезпечення САПР.....	19
Лекція 5. Пам'ять ЕОМ.....	21
Лекція 6. Типи обчислювальних машин і систем.....	26
Лекція 7. Математичне забезпечення САПР.....	28
Лекція 8. Механічні системи.....	30
Лекція 9. Алгоритми чисельного інтегрування систем диференціальних рівнянь.....	35
Лекція 10. Методи аналізу на мікрорівні.....	43
Лекція 11. Геометричне моделювання й машинна графіка.....	48
Лекція 12. Автоматизовані системи в промисловості.....	53
Лекція 13. Програмування для верстатів з ЧПК	57
Лекція 14. Основні функції САЕ-систем.....	59
Лекція 15. Огляд машинобудівних САПР.....	61
Лекція 16. Методичне й програмне забезпечення автоматизованих систем.....	65
Лекція 17. Специфікації проектів програмних систем.....	68
Лекція 18. Програмне забезпечення CASE-систем.....	69

Лекція 1. Введення в автоматизоване проектування

1.1. Поняття проектування

Проектування технічного об'єкта — створення, перетворення й подання в прийнятій формі образу цього ще не існуючого об'єкта. Образ об'єкта або його складових частин може створюватися в уяві людини в результаті творчого процесу або генеруватися відповідно до деяких алгоритмів у процесі взаємодії людини й ЕОМ. У кожному разі інженерне проектування починається при наявності вираженої потреби суспільства в деяких технічних об'єктах, якими можуть бути об'єкти будівництва, промислові виробы або процеси. Проектування містить у собі розробку технічної пропозиції та (або) *технічного завдання* (ТЗ), що відбивають ці потреби, і реалізацію ТЗ у вигляді проектної документації.

Найчастіше ТЗ представляють у вигляді деяких документів, і воно є вихідним (первинним) описом об'єкта. Результатом проектування, як правило, служить повний комплект документації, що містить достатні відомості для виготовлення об'єкта в заданих умовах. Ця документація і є *проект*, точніше остаточний опис об'єкта. Більш коротко, проектування – процес, що полягає в одержанні й перетворенні вихідного опису об'єкта в остаточний опис на основі виконання комплексу робіт дослідницького, розрахункового й конструкторського характеру.

Перетворення вихідного опису в остаточне породжує ряд проміжних описів, що підводять підсумки рішення деяких задач і використовуваних для обговорення й прийняття рішень для закінчення або продовження проектування. Такі проміжні описи називають *проектними рішеннями*.

Проектування, при якому всі проектні рішення або їхня частина одержуються шляхом взаємодії людини й ЕОМ, називають *автоматизованим проектуванням*, на відміну від ручного (без використання ЕОМ) або автоматичного (без участі людини на проміжних етапах). Система, що реалізує автоматизоване проектування, являє собою *систему автоматизованого проектування* (САПР, в англійській мові написанні CAD System — Computer Aided Design System).

Автоматичне проектування можливо лише в окремих випадках для порівняно нескладних об'єктів. Переважаючим в цей час є автоматизоване проектування.

Проектування складних об'єктів засновано на застосуванні ідей і принципів, викладених у ряді теорій і підходів. Найбільш загальним підходом є системний підхід, ідеями якого пронизані різні методики проектування складних систем.

1.2. Принципи системного підходу

Проектування складних об'єктів засновано на застосуванні ідей і принципів, викладених у ряді теорій і підходів. Найбільш загальним підходом є *системний підхід*, ідеями якого пронизані різні методики проектування складних систем.

Для фахівця в області системотехніки ідеї й принципи системного підходу є очевидними й природними, однак їхнє дотримання й реалізація найчастіше сполучені з певними труднощами, що обумовлюються особливостями проектування. Як і більшість дорослих людей, що правильно використовують рідну мову без залучення правил граматики, інженери використовують системний підхід без звертання до допомоги системного аналізу. Однак інтуїтивний підхід без застосування правил системного аналізу може виявитися недостатнім для рішення все більш ускладнюючихся задач інженерної діяльності.

Основний загальний принцип системного підходу полягає в розгляді частин

досліджуваного явища або складної системи з урахуванням їх взаємодії. Системний підхід містить у собі виявлення структури системи, типізацію зв'язків, визначення атрибутів, аналіз впливу зовнішнього середовища, формування моделі системи, дослідження моделі й, можливо, *оптимізацію* її структури й функціонування.

Системний підхід є базою для узагальнюючої дисципліни "теорія систем" (інша використовувана назва — "системний аналіз"). *Теорія систем* — дисципліна, у якій конкретизуються положення системного підходу; вона присвячена дослідженню й проектуванню складних економічних, соціальних, технічних систем, найчастіше слабкоструктурованих. Характерними прикладами таких систем є виробничі системи. При проектуванні систем мета досягається в багатокрокових процесах прийняття рішень. Методи прийняття рішень часто виділяють у самостійну дисципліну, що має назву "Теорія прийняття рішень".

В техніці дисципліну, у якій досліджуються складні технічні системи, їхнє проектування, і аналогічну теорії систем, частіше називають *системотехнікою*. Предметом системотехніки є, по-перше, організація процесу створення, використання й розвитку технічних систем, по-друге, методи й принципи їхнього проектування й дослідження. У системотехніку важливо вміти сформулювати цілі системи й організувати її розгляд з позицій поставлених цілей. Тоді можна відкинути зайві й малозначимі частини при проектуванні й моделюванні, перейти до постановки оптимізаційних задач.

Системи автоматизованого проектування й керування відносяться до числа найбільш складних сучасних систем. Їхнє проектування й використання неможливі без системного підходу. Тому ідеї й положення системотехніки входять складовою частиною в дисципліни, присвячені вивченню сучасних автоматизованих систем і технологій їхнього створення й застосування.

Інтерпретація й конкретизація системного підходу мають місце в ряді відомих підходів з іншими назвами, які також можна розглядати як компоненти системотехніки. Такими є структурні, блочно-ієрархічні, об'єктно-орієнтовані підходи.

При *структурному* підході, як різновиду системного, потрібно синтезувати варіанти системи з компонентів (блоків) і оцінювати варіанти при їхньому частковому переборі з попереднім прогнозуванням характеристик компонентів.

Блочно-ієрархічний підхід до проектування використовує ідеї декомпозиції складних описів об'єктів і відповідно засобів їхнього створення на ієрархічні рівні й аспекти, вводить поняття стилю проектування (висхідне й спадне), встановлює зв'язок між параметрами сусідніх ієрархічних рівнів.

Ряд важливих структурних принципів, використовуємих при розробці інформаційних систем і, насамперед, їх програмного забезпечення (ПЗ), виражений у підході, що має назву *об'єктно-орієнтованим проектуванням* (ООП). Такий підхід має наступні переваги в рішенні проблем керування складністю й інтеграції ПЗ:

- вносить у моделі програм більшу структурну визначеність, розподіляючи представлені в програмі дані й процедури між класами об'єктів;
- скорочує об'єм специфікацій, завдяки введенню в описи ієрархії об'єктів і відносин наслідування між властивостями об'єктів різних рівнів ієрархії;
- зменшує ймовірність перекручування даних внаслідок помилкових дій за рахунок обмеження доступу до певних категорій даних в об'єктах.

Опис у кожному класі об'єктів припустимих звертань до них і прийнятих форматів повідомлень полегшує узгодження й інтеграцію ПЗ.

Для всіх підходів до проектування складних систем характерні також наступні

особливості:

1. Структуризація процесу проектування, що виражає декомпозицією проектних задач і документації, виділенням стадій, етапів, проектних процедур. Ця структуризація є сутністю блочно-ієрархічного підходу до проектування.
2. Ітераційний характер проектування.
3. Типізація й уніфікація проектних рішень і засобів проектування.

У теорії систем і системотехніці уведений ряд термінів, серед них до базових потрібно віднести наступні поняття:

- *Система* — множина елементів, що перебувають у відносинах і зв'язках між собою.
- *Елемент* — така частина системи, уявлення про яку недоцільно піддавати при проектуванні подальшому членуванню.
- *Складна система* — система, що характеризується більшим числом елементів і, що найбільш важливо, більшим числом взаємозв'язків елементів. Складність системи визначається також видом взаємозв'язків елементів, властивостями цілеспрямованості, цілісності, членимості, ієрархічності, багатоаспектності. Очевидно, що сучасні автоматизовані інформаційні системи й, зокрема, системи автоматизованого проектування, є складними в силу наявності в них перерахованих властивостей і ознак.
- *Підсистема* — частина системи (підмножина елементів і їхніх взаємозв'язків), що має властивості системи.
- *Надсистема* — система, стосовно якої розглянута система є підсистемою.
- *Структура* — відображення сукупності елементів системи і їхніх взаємозв'язків; поняття структури відрізняється від поняття самої системи також тим, що при описі структури беруть до уваги лише типи елементів і зв'язків без конкретизації значень їхніх параметрів.
- *Параметр* — величина, що виражає властивість або системи, або її частини, або середовища, що впливає на систему. Звичайно в моделях систем як параметри розглядають величини, що не змінюються в процесі дослідження системи. Параметри підрозділяють на зовнішні, внутрішні й вихідні, що виражають властивості елементів системи, самої системи, зовнішнього середовища відповідно. Вектори внутрішніх параметрів, вихідних параметрів і зовнішніх параметрів позначаються $\mathbf{X} = (x_1, x_2, \dots, x_n)$, $\mathbf{Y} = (y_1, y_2, \dots, y_m)$, $\mathbf{Q} = (q_1, q_2, \dots, q_k)$ відповідно.
- *Фазова змінна* — величина, що характеризує енергетичне або інформаційне наповнення елемента або підсистеми.
- *Стан* — сукупність значень фазових змінних, зафіксованих в одній тимчасовій точці процесу функціонування.
- *Поводження (динаміка) системи* — зміна стану системи в процесі функціонування.
- *Система без післядії* — її поведінка при $t > t_0$ визначається завданням стану в момент t_0 і вектором зовнішніх впливів $\mathbf{Q}(t)$. У системах з післядією, крім того, потрібно знати передісторію поведінки, тобто стану системи в моменти, що передують t_0 .
- *Вектор змінних \mathbf{V} , що характеризують стан (вектор змінних стану)* — ненадлишкова множина фазових змінних, завдання значень яких у деякий момент часу повністю визначає поведінку системи надалі (в автономних системах без післядії).
- *Простір станів* — множина можливих значень вектора змінних стану.
- *Фазова траєкторія* — подання процесу (залежності $\mathbf{V}(t)$) у вигляді послідовності точок у просторі станів.

До характеристик складних систем часто відносять наступні поняття:

- Цілеспрямованість — властивість штучної системи, що виражає призначення системи. Ця властивість необхідна для оцінки ефективності варіантів системи.
- Цілісність — властивість системи, що характеризує взаємозв'язок елементів і наявність залежності вихідних параметрів від параметрів елементів, при цьому більшість вихідних параметрів не є простим повторенням або сумою параметрів елементів.
- Ієрархічність — властивість складної системи, що виражає можливість і доцільність її ієрархічного опису, тобто подання у вигляді декількох рівнів, між компонентами яких є відносини ціле—частина.

Складовими частинами системотехніки є наступні основні розділи:

- ієрархічна структура систем, організація їхнього проектування;
- аналіз і моделювання систем;
- синтез і оптимізація систем.

Моделювання має дві чітко помітні задачі:

1. створення моделей складних систем (в англomовному написанні — modeling);
2. аналіз властивостей систем на основі дослідження їхніх моделей (simulation).

Синтез також підрозділяють на дві задачі:

1. синтез структури проєктованих систем (структурний синтез);
2. вибір чисельних значень параметрів елементів систем (параметричний синтез).

Ці задачі відносяться до області прийняття проєктних рішень.

Моделювання й оптимізацію бажано виконувати з урахуванням статистичної природи систем. Детермінованість – лише окремий випадок. При проєктуванні характерна недостача достовірних вихідних даних, невизначеність умов прийняття рішень. Врахування статистичного характеру даних при моделюванні значною мірою заснований на методі статистичних випробувань (методі Монте-Карло), а прийняття рішень – на використанні нечітких множин, експертних систем, еволюційних обчислень.

Приклад 1

Комп'ютер є складною системою в силу наявності в нього великої кількості елементів, різноманітних зв'язків між елементами й підсистемами, властивостей цілеспрямованості, цілісності, ієрархічності. До підсистем комп'ютера відносяться процесор (процесори), оперативна пам'ять, кеш-пам'ять, шини, пристрої вводу-виводу. Як надсистема можуть виступати обчислювальна мережа, автоматизована й (або) організаційна система, до яких належить комп'ютер. Внутрішні параметри – час виконання арифметичних операцій, зчитування (запису) у накопичувачах, пропускна здатність шин та ін. Вихідні параметри – продуктивність комп'ютера, ємність оперативної й зовнішньої пам'яті, собівартість, час наробітку на відмову й ін. Зовнішні параметри – напруга живлення мережі і її стабільність, температура навколишнього середовища й ін.

Приклад 2

Для двигуна внутрішнього згорання підсистемами є колінчатий вал, механізм газорозподілу, поршнева група, система змащення й охолодження. Внутрішні параметри – число циліндрів, об'єм камери згорання й ін. Вихідні параметри – потужність двигуна, ККД,

витрата палива й ін. Зовнішні параметри – характеристики палива, температура повітря, навантаження на вихідному валу.

Лекція 2. Рівні проектування

При використанні блочно-ієрархічного підходу до проектування уявлення про спроектовану систему розчленовують на ієрархічні рівні. На верхньому рівні використовують найменш деталізоване уявлення, що відбиває тільки найбільш загальні риси й особливості спроектованої системи. На наступних рівнях ступінь подробиць опису зростає, при цьому розглядають вже окремі блоки системи, але з урахуванням впливів на кожний його сусідів. Такий підхід дозволяє на кожному ієрархічному рівні формулювати задачі прийнятної складності, що піддаються рішенню за допомогою наявних засобів проектування. Розбивка на рівні повинна бути такою, щоб документація на блок будь-якого рівня була доступна для огляду й сприймана однією людиною.

Інакше кажучи, блочно-ієрархічний підхід є декомпозиційним підходом, що заснований на розбивці складної задачі великої розмірності на послідовно та (або) паралельно розв'язувані групи задач малої розмірності, що істотно скорочує вимоги до використовуваних обчислювальних ресурсів або до часу рішення задач.

Можна говорити не тільки про ієрархічні рівні специфікацій, але й про ієрархічні рівні проектування, розуміючи під кожним з них сукупність специфікацій деякого ієрархічного рівня разом з постановками задач, методами одержання описів і рішення виникаючих проектних задач.

Список ієрархічних рівнів для кожного застосування може бути специфічним, але для більшості застосувань характерно наступне найбільш велике виділення рівнів:

- *системний рівень*, на якому вирішують найбільш загальні задачі проектування систем, машин і процесів; результати проектування представляють у вигляді структурних схем, генеральних планів, схем розміщення устаткування, діаграм потоків даних і т.п.;

- *макрорівень*, на якому проектують окремі пристрої, вузли машин і приладів; результати представляють у вигляді функціональних, принципівих і кінематичних схем, складальних креслень і т.п.;

- *мікрорівень*, на якому проектують окремі деталі й елементи машин і приладів.

Для кожного застосування число виділюваних рівнів і їхніх найменувань можуть бути різними. Так, у радіоелектроніці мікрорівень часто називають компонентним, макрорівень — *схемотехнічним рівнем*. Між схемотехнічним і системним рівнями вводять рівень, що маж назву *функціонально-логічного рівня*. В обчислювальній техніці системний рівень підрозділяють на рівні проектування ЕОМ (обчислювальних систем) і обчислювальних мереж. У машинобудуванні є рівні деталей, вузлів, машин, комплексів.

Залежно від послідовності рішення задач ієрархічних рівнів розрізняють спадне, висхідне й змішане проектування (*стили проектування*). Послідовність рішення задач від нижніх рівнів до верхнього характеризує *висхідне проектування*, зворотна послідовність приводить до *спадного проектування*, у *змішаному стилі* є елементи як висхідного, так і спадного проектування. У більшості випадків для складних систем віддають перевагу спадному проектуванню. Відзначимо однак, що при наявності заздалегідь спроектованих складених блоків (пристроїв) можна говорити про змішане проектування.

Невизначеність і нечіткість вихідних даних при спадному проектуванні (ще не спроектовані компоненти) або вихідних вимог при висхідному проектуванні (оскільки ТЗ є на всю систему, а не на її частині) обумовлюють необхідність прогнозування відсутніх даних з

наступним їхнім уточненням, тобто послідовного наближення до остаточного рішення (ітераційність проектування).

Поряд з декомпозицією описів на ієрархічні рівні застосовують поділ уявлень про проєктовані об'єкти на аспекти.

Аспект опису — опис системи або її частин з деякої обговореної точки зору, обумовленої функціональним, фізичними або іншим типом відносин між властивостями й елементами.

Розрізняють аспекти функціональний, інформаційний, структурний і поведінковий (процесний). Функціональний опис відносять до функцій системи й найчастіше представляють його функціональними схемами. Одержання функціональних описів часто називають *функціональним проектуванням*.

Інформаційний опис містить у собі основні поняття предметної області (сутності), словесне пояснення або числові значення характеристик (атрибутів) використовуваних об'єктів, а також опис зв'язків між цими поняттями й характеристиками. Інформаційні моделі можна представляти графічно (графи, діаграми сутність-відношення), у вигляді таблиць або списків. Одержання інформаційних описів часто називають інформаційним проектуванням або стосовно до створення баз даних – інфологічним проектуванням.

Структурний опис ставиться до морфології системи, характеризує складові частини системи і їх з'єднання й може бути представлено структурними схемами, а також різного роду конструкторською документацією. Одержання конструкторської документації, тобто опис геометричних форм виробів, складу компонентів і їхнього просторового розміщення, називають *конструкторським проектуванням*.

Поведінковий опис характеризує процеси функціонування (алгоритми) системи й (або) технологічні процеси створення системи. Розробка алгоритмів і програмного забезпечення систем є предметом *алгоритмічного проектування*, а розробка технологічних процесів виготовлення виробів — предметом *технологічного проектування*.

Іноді аспекти описів зв'язують із підсистемами, функціонування яких засновано на різних фізичних процесах.

Відзначимо, що в загальному випадку виділення аспектів опису може бути неоднозначним. Так, крім зазначеного підходу, очевидна доцільність виділення таких аспектів, як функціональне (розробка принципів дії, структурних, функціональних, принципових схем), конструкторське (визначення форм і просторового розташування компонентів виробів), алгоритмічне (розробка алгоритмів і програмного забезпечення) і технологічне (розробка технологічних процесів) проектування систем. Прикладами аспектів опису у випадку САПР можуть служити також розглянуті далі види забезпечення автоматизованого проектування.

2.1. Стадії проектування

Стадії проектування — найбільш великі частини проектування, як процесу, що розвивається в часі. У загальному випадку виділяють стадії *науково-дослідних робіт* (НДР), *ескізного проекту* або *дослідно-конструкторських робіт* (ДКР), *технічного, робочого проектів, випробувань дослідних зразків або дослідних партій*. Стадію НДР іноді називають *передпроектними дослідженнями* або стадією *технічної пропозиції*. Очевидно, що в міру переходу від стадії до стадії ступінь подробиць й глибина пророблення проекту зростають, і робочий проект вже повинен бути цілком достатнім для виготовлення дослідних або серійних зразків. Близьким до визначення стадії, але менш чітко обговореним поняттям, є поняття *етапу проектування*. Проектування на початкових етапах, у процесі якого приймаються принципові проектні рішення по вигляду й принципам дії проєктованих пристроїв і систем, називають *концептуальним проектуванням*.

Стадії (етапи) проектування підрозділяють на складові частини, що мають назву *проектних процедур*. Прикладами проектних процедур можуть служити підготовка деталіровочних креслень, аналіз кінематики, моделювання перехідного процесу, оптимізація параметрів і інші проектні задачі. У свою чергу, проектні процедури можна розчленувати на більш дрібні компоненти, що мають назву *проектних операцій*, наприклад, при аналізі міцності деталі сітковими методами операціями можуть бути побудова сітки, вибір або розрахунок зовнішніх впливів, моделювання полів напруг і деформацій, подання результатів моделювання в графічній і текстовій формах. Проектування зводиться до виконання деяких послідовностей проектних процедур — *маршрутів проектування*.

Прагнення скоротити тимчасові витрати на проектування привело до розробки методик *паралельного проектування* (сполученого проектування), при якому паралельно в часі вирішуються задачі, зв'язані одна з одною по вхідним і вихідним даним таким чином, що для рішення однієї з них потребується знання результатів рішення іншої задачі. Оскільки ці результати до початку процедури паралельного проектування ще не отримані, у методиці паралельного проектування повинні бути зазначені способи завдання ще не визначених значень параметрів. Іноді розробку технічного завдання на проектування називають *зовнішнім проектуванням*, а реалізацію ТЗ — *внутрішнім проектуванням*.

У ТЗ на проектування об'єкта вказують, принаймні, наступні дані:

1. Призначення об'єкта;
2. Умови експлуатації. Поряд з якісними характеристиками (представленими у вербальній формі) є числові параметри, що мають назву *зовнішніх параметрів*, для яких зазначені області припустимих значень. Приклади зовнішніх параметрів: температура навколишнього середовища, зовнішні сили, електричні напруги, навантаження й т.п.;
3. Вимоги до *вихідних параметрів*, тобто до величин, що характеризують властивості об'єкта, що цікавлять споживача. Ці вимоги виражені у вигляді *умов працездатності*: $y_i \mathop{R} T_i$, де y_i — i -й вихідний параметр, $R \in \{=, <, >, \leq, \geq\}$ — вид відносини; T_i — норма i -го вихідного параметра. У випадку, якщо R — відношення рівності, потрібно задати необхідну точність виконання рівності.

2.1. Моделі і їхні параметри в САПР

В автоматизованих проектних процедурах замість ще неіснуючого спроектованого об'єкта оперують деяким квазіоб'єктом — *моделлю*, що відбиває деякі цікаві для дослідника властивості об'єкта. Модель може бути фізичним об'єктом (макет, стенд) або специфікацією. Серед моделей-специфікацій розрізняють функціональні, поведінкові, інформаційні, структурні моделі. Ці моделі називають *математичними моделями*, якщо вони формалізовані засобами апарата й мови математики.

У свою чергу, математичні моделі можуть бути геометричними, топологічними, динамічними, логічними й т.п., якщо вони відбивають відповідні властивості об'єктів. Поряд з математичними моделями при проектуванні використовують функціональні моделі, інформаційні моделі у вигляді діаграм сутність-відношення, геометричні моделі (креслення). Надалі, якщо немає спеціального застереження, під словом "модель" будемо мати на увазі математичну модель.

Математична функціональна модель у загальному випадку являє собою алгоритм обчислення вектора вихідних параметрів Y при заданих векторах параметрів елементів (*внутрішніх параметрів*) X і *зовнішніх параметрів* Q .

Математичні моделі можуть бути символічними й чисельними. При використанні *символьних* моделей оперують не значеннями величин, а їхніми символічними позначеннями

(ідентифікаторами). Чисельні моделі можуть бути *аналітичними моделями*, тобто їх можна представити у вигляді явно виражених залежностей вихідних параметрів Y від параметрів внутрішніми X й зовнішніх Q , або *алгоритмічними моделями*, у яких зв'язок Y , X і Q заданий неявно у вигляді алгоритму *модельовання*. Найважливіший окремий випадок алгоритмічних моделей — *імітаційні моделі*, вони відображають процеси в системі при наявності зовнішніх впливів на систему. Інакше кажучи, імітаційна модель – це алгоритмічна поведінкова модель.

Класифікацію математичних моделей виконують також по ряду інших ознак.

Так, залежно від приналежності до того або іншого ієрархічного рівня виділяють моделі рівнів системного, функціонально-логічного, макрорівня (зосередженого) і мікрорівня (розподіленого).

По характеру використовуваного для опису математичного апарата розрізняють моделі лінгвістичні, теоретико-множинні, абстрактно-алгебраїчні, нечіткі, автоматні й т.п.

Наприклад, на системному рівні переважно застосовують моделі систем масового обслуговування й мережі Петрі, на функціонально-логічному рівні – автоматні моделі на основі апарата передатних функцій або кінцевих автоматів, на макрорівні – системи алгебро-диференціальних рівнянь, на мікрорівні – диференціальні рівняння в частинних похідних. Особливе місце займають геометричні моделі, що використовуються в системах конструювання.

Крім того, введені поняття *повних моделей* і *макромоделей*, моделей статичних і динамічних, детермінованих і стохастичних, аналогових і дискретних, символічних і чисельних.

Повна модель об'єкта на відміну від макромоделі описує не тільки процеси на зовнішніх виводах моделюємого об'єкта, але й внутрішні для об'єкта процеси.

Статичні моделі описують статичні стани, у них не є присутнім час у якості незалежної змінної. Динамічні моделі відбивають поведження системи, тобто в них обов'язково використовується час.

Стохастичні й детерміновані моделі розрізняються залежно від врахування або неврахування випадкових факторів.

Інформаційні моделі використають насамперед при інфологічному проектуванні баз даних (БД) для опису зв'язків між одиницями інформації.

Найбільші труднощі виникають при створенні моделей слабкоструктурованих систем, що характерно насамперед для системного рівня проектування. Тут значна увага приділяється експертним методам. У теорії систем сформульовані загальні рекомендації з підбора експертів при розробці моделі, організації експертизи, по обробці отриманих результатів. Досить загальний підхід до побудови моделей складних слабкоструктурованих систем виражений у методиках IDEF.

Звичайно в імітаційних моделях фігурують величини, що характеризують стан моделюємої системи й мають назву *фазових змінних*. Так, на макрорівні імітаційні моделі являють собою системи алгебро-диференціальних рівнянь

$$\Phi \left(\frac{d\mathbf{V}}{dt}, \mathbf{V}, t \right) = 0, \text{ при } t = 0, \mathbf{V} = \mathbf{V}_0, \quad (1.1)$$

де \mathbf{V} — вектор фазових змінних; t — час; \mathbf{V}_0 — вектор початкових умов. До фазового змінного можна віднести струми й напруги в електричних системах, сили й швидкості – у механічних, тиски й витрати – у гідравлічних.

В *аналогових моделях* фазові змінні — безперервні величини, у *дискретних моделях* — дискретні, в окремому випадку дискретні моделі є логічними (булевськими), у них стан системи і її елементів описується булевськими величинами. У ряді випадків корисне застосування *змішаних моделей*, у яких одна частина підсистем характеризується аналоговими

моделями, інша — логічними.

Вихідні параметри систем можуть бути двох типів. По-перше, це параметри-функціонали, тобто функціонали залежностей $V(t)$ у випадку використання (1.1). Приклади таких параметрів: амплітуди сигналів, тимчасові затримки, потужності розсіювання й т.п. По-друге, це параметри, що характеризують здатність проєктованого об'єкта працювати при певних зовнішніх умовах. Ці вихідні параметри є граничними значеннями діапазонів зовнішніх змінних, у яких зберігається працездатність об'єкта.

Лекція 3. Проектні процедури

Створити проєкт об'єкта (виробу або процесу) означає вибрати структуру об'єкта, визначити значення всіх його параметрів і представити результати у встановленій формі. Результати (проєктна документація) можуть бути виражені у вигляді креслень, схем, пояснювальних записок, програм для програмно-керованого технологічного устаткування й інших документів на папері або на машинних носіях інформації.

Розробка (або вибір) структури об'єкта є проєктна процедура, що має назву *структурного синтезу*, а розрахунок (або вибір) значень параметрів елементів X — процедура *параметричного синтезу*.

Задача структурного синтезу формулюється в системотехніці як задача прийняття рішень (ЗПР). Її суть полягає у визначенні мети, множини можливих рішень і обмежуючих умов.

Класифікацію ЗПР здійснюють по ряду ознак. По числу критеріїв розрізняють задачі одно- і многокритеріальні. По ступені невизначеності розрізняють ЗПР детерміновані, ЗПР в умовах ризику – при наявності у формулюванні задачі випадкових параметрів, ЗПР в умовах невизначеності, тобто при неповноті або невірогідності вихідної інформації.

Реальні задачі проєктування, як правило, є багатокритеріальними. Одна з основних проблем постановки багатокритеріальних задач – встановлення правил переваги варіантів. Способи зведення багатокритеріальних задач до однокритеріальних і наступні шляхи рішення вивчаються в дисциплінах, присвячених методам оптимізації й математичному програмуванню.

Наявність випадкових факторів ускладнює рішення ЗПР. Основні підходи до рішення ЗПР в умовах ризику полягають або в рішенні "для найгіршого випадку", або в урахуванні в цільовій функції математичного очікування й дисперсії вихідних параметрів. У першому випадку задачу вирішують як детерміновану при завищених вимогах до якості рішення, що є головним недоліком підходу. У другому випадку вірогідність результатів рішення набагато вище, але виникають труднощі з оцінкою цільової функції. Застосування методу Монте-Карло у випадку алгоритмічних моделей стає єдиною альтернативою й, отже, для рішення потрібні значні обчислювальні ресурси.

Існують дві групи ЗПР в умовах невизначеності. Одна з них вирішується при наявності протидії розумного супротивника. Такі задачі вивчаються в теорії ігор, для задач проєктування в техніці вони не характерні. У другій групі досягненню мети протидію створюють сили природи. Для їхнього рішення корисно використати теорію й методи нечітких множин.

При синтезі структури автоматизованої системи постановка задачі повинна включати в якості вихідних даних наступні відомості:

- множина функцій, що виконуються системою (інакше кажучи, множина робіт, кожна з яких може складатися з однієї або більше операцій); можливо, що в цій множині є часткова впорядкованість робіт, що може бути представлене у вигляді орієнтованого графа, у якому вершини відповідають роботам, а дуги - відносинам порядку;
- типи припустимих для використання серверів (машин), що виконують функції системи;

- множина зовнішніх джерел і споживачів інформації;
- у багатьох випадках задається також деяка вихідна структура системи у вигляді взаємозалежної сукупності серверів певних типів; ця структура може розглядатися як узагальнена надлишкова або як варіант першого наближення;
- різного роду обмеження, зокрема, обмеження на витрати матеріальних ресурсів і (або) на часи виконання функцій системи.

Задача полягає в синтезі (або корекції) структури, визначенні типів серверів (програмно-апаратних засобів), розподілі функцій по серверах таким чином, щоб досягався екстремум цільової функції при виконанні заданих обмежень.

Конструювання, розробка технологічних процесів, оформлення проектної документації – окремі випадки структурного синтезу.

Задачу параметричного синтезу називають параметричною оптимізацією (або оптимізацією), якщо її вирішують як задачу математичного програмування, тобто

$$\text{extr } F(\mathbf{X}), \mathbf{X} \in \mathbf{D}_x,$$

де $F(\mathbf{X})$ — цільова функція; \mathbf{X} — вектор керованих параметрів (що мають назву також проектними або вар'юєними); $\mathbf{D}_x = \{ \mathbf{X} \mid \varphi(\mathbf{X}) < 0, \psi(\mathbf{X}) = 0 \}$ — припустима область; $\varphi(\mathbf{X})$ і $\psi(\mathbf{X})$ — функції-обмеження.

Наступна після синтезу група проектних процедур — процедури *аналізу*. Ціль аналізу — одержання інформації про характер функціонування й значення вихідних параметрів \mathbf{Y} при заданих структурі об'єкта, відомостях про зовнішні параметри \mathbf{Q} й параметри елементів \mathbf{X} . Якщо задані фіксовані значення параметрів \mathbf{X} і \mathbf{Q} , то має місце процедура *одноваріантного аналізу*. Одноваріантний аналіз часто виконується за допомогою моделювання.

Моделювання складається з етапів *формування моделі* (modeling) і *дослідження моделі* (рішення, simulation). У свою чергу, формування моделі включає дві процедури: по-перше, розробку моделей окремих компонентів, по-друге, формування моделі системи з моделей компонентів.

Перша із цих процедур виконується попередньо стосовно типових компонентів поза маршрутом проектування конкретних об'єктів. Як правило, моделі компонентів розробляються фахівцями в прикладних областях, причому знаючих вимоги до моделей і форм їхнього подання в САПР. Звичайно в допомогу розроблювачеві моделей у САПР пропонуються методики й допоміжні засоби, наприклад, у вигляді програм аналізу для експериментального відпрацювання моделей. Створені моделі включаються в бібліотеки моделей прикладних програм аналізу.

На маршруті проектування кожного нового об'єкта виконується друга процедура (Рис. 1) — формування моделі системи з використанням бібліотечних моделей компонентів. Як правило, ця процедура виконується автоматично по алгоритмах, включених у заздалегідь розроблені програми аналізу. Приклади таких програм є в різних додатках і насамперед у галузях загального машинобудування й радіоелектроніки.

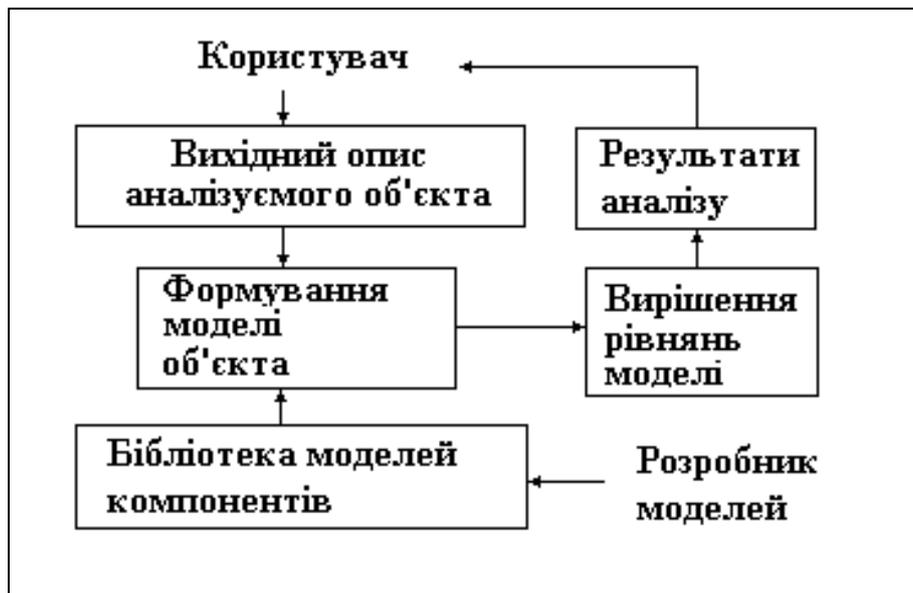


Рис. 1. Формування моделі системи

При застосуванні цих програм користувач описує досліджуваний об'єкт вхідною мовою програми аналізу не у вигляді системи рівнянь, що буде отримана автоматично, а у вигляді списку елементів структури, еквівалентної схеми, ескізу або кресленню конструкції.

Друга процедура моделювання — *simulation* — зводиться до рішення рівнянь математичної моделі, наприклад, системи диференціальних рівнянь, і обчисленню вектора вихідних параметрів \mathbf{Y} .

Якщо задані статистичні відомості про параметри \mathbf{X} й потрібно одержати оцінки числових характеристик розподілів вихідних параметрів (наприклад, оцінки математичних очікувань і дисперсій), то це процедура *статистичного аналізу*. Якщо потрібно розрахувати матриці абсолютної \mathbf{A} й (або) відносної \mathbf{B} чутливості, то має місце задача *аналізу чутливості*.

Елемент A_{ji} матриці \mathbf{A} називають абсолютним коефіцієнтом чутливості, він являє собою частинну похідну J -го вихідного параметра y_j по i -ому параметру x_i . Інакше кажучи, A_{ji} є елементом вектора градієнта J -го вихідного параметра. На практиці зручніше використовувати безрозмірні відносні коефіцієнти чутливості B_{ji} , що характеризують ступінь впливу змін параметрів елементів на зміни вихідних параметрів:

$$B_{ji} = \frac{A_{ji} x_{i \text{ ном}}}{y_{j \text{ ном}}}$$

де $x_{i \text{ ном}}$ й $y_{j \text{ ном}}$ — номінальні значення параметрів x_i і y_j відповідно.

У процедурах *багатоваріантного аналізу* визначається вплив зовнішніх параметрів, розкиду й нестабільності параметрів елементів на вихідні параметри. Процедури статистичного аналізу й аналізу чутливості — характерні приклади процедур багатоваріантного аналізу.

Виконання аналізу й зіставлення отриманих результатів з бажаними значеннями називають процедурою *верифікації*.

3.1. Життєвий цикл виробів

Життєвий цикл промислових виробів (ЖЦВ) включає ряд етапів, починаючи від зародження ідеї нового продукту до його утилізації по закінченні строку використання. До них відносяться етапи проектування, технологічної підготовки виробництва (ТПВ), виробництва, реалізації продукції, експлуатації й, нарешті, утилізації (у число етапів життєвого циклу можуть також входити маркетинг, закупівля матеріалів і комплектуючих, надання послуг, упакування й зберігання, монтаж і введення в експлуатацію).

Розглянемо зміст основних етапів ЖЦВ для виробів машинобудування.

На етапі проектування виконуються проектні процедури — формування принципового рішення, розробка геометричних моделей і креслень, розрахунки, моделювання процесів, оптимізація й т.п.

На етапі підготовки виробництва розробляються маршрутна й операційна технології виготовлення деталей, реалізовані в програмах для верстатів ЧПК; технологія зборки й монтажу виробів; технологія контролю й випробувань.

На етапі виробництва здійснюються: календарне й оперативне планування; придбання матеріалів і комплектуючих з їхнім вхідним контролем; механообробки й інших необхідні види обробки; контроль результатів обробки; зборка; випробування й підсумковий контроль.

На поствиробничих етапах виконуються консервація, упакування, транспортування; монтаж у споживача; експлуатація, обслуговування, ремонт; утилізація.

На всіх етапах життєвого циклу є свої цільові настанови. При цьому учасники життєвого циклу прагнуть досягти поставлених цілей з максимальною ефективністю. На етапах проектування, ТПВ і виробництва потрібно забезпечити виконання вимог, пропорованих до продукту, при заданому ступені надійності виробу й мінімізації матеріальних і тимчасових витрат, що необхідно для досягнення успіху в конкурентній боротьбі в умовах ринкової економіки. Поняття ефективності охоплює не тільки зниження собівартості продукції й скорочення строків проектування й виробництва, але й забезпечення зручності освоєння й зниження витрат на майбутню експлуатацію виробів.

Досягнення поставлених цілей на сучасних підприємствах, що випускають складні технічні вироби, виявляється неможливим без широкого використання *автоматизованих систем* (АС), заснованих на застосуванні комп'ютерів і призначених для створення, переробки й використання всієї необхідної інформації про властивості виробів і супровідних процесів. Специфіка задач, розв'язуваних на різних етапах життєвого циклу виробів, обумовлює різноманітність застосовуваних АС.

3.2. Структура САПР

Як і будь-яка складна система, САПР складається з підсистем. Розрізняють підсистеми що проектують і обслуговують.

Підсистеми, що проектують, безпосередньо виконують проектні процедури. Прикладами підсистем, що проектують, можуть служити підсистеми геометричного тривимірного моделювання механічних об'єктів, виготовлення конструкторської документації, схемотехнічного аналізу, трасування з'єднань у друкованих платах.

Обслуговуючі підсистеми забезпечують функціонування підсистем, що проектують, їхню сукупність часто називають системним середовищем (або оболонкою) САПР. Типовими обслуговуючими підсистемами є підсистеми керування проектними даними, підсистеми розробки й супроводу програмного забезпечення CASE (Computer Aided Software Engineering), підсистеми, що навчають, для освоєння користувачами технологій, реалізованих у САПР.

Структурування САПР по різних аспектах обумовлює поява *видів забезпечення САПР*. Прийнято виділяти сім видів забезпечення:

- *технічне забезпечення* (ТЗ), що включає різні апаратні засоби (ЕОМ, периферійні пристрої, мережне комутаційне устаткування, лінії зв'язку, вимірювальні засоби);
- *математичне забезпечення* (МЗ), що поєднує математичні методи, моделі й алгоритми для виконання проектування;
- *програмне забезпечення* (ПЗ), що представляє комп'ютерними програмами САПР;
- *інформаційне забезпечення* (ІЗ), що складається з баз даних (БД), систем керування базами даних (СКБД), а також включає інші дані, що використовуються при проектуванні;
- *лінгвістичне забезпечення* (ЛЗ), що виражається мовами спілкування між проектувальниками й ЕОМ, мовами програмування й мовами обміну даними між технічними засобами САПР;
- *методичне забезпечення* (МетЗ), що включає різні методики проектування, іноді до МетЗ відносять також математичне забезпечення;
- *організаційне забезпечення* (ОЗ), що представляється штатними розкладами, посадовими інструкціями й іншими документами, що регламентують роботу проектного підприємства.

Відзначимо, що вся сукупність використовуваних при проектуванні даних називається інформаційним фондом САПР. *Базою даних* називають упорядковану сукупність даних, що відображають властивості об'єктів і їхнього взаємозв'язку в деякій предметній області. Доступ до БД для читання, запису й модифікації даних здійснюється за допомогою СКБД, а сукупність БД і СКБД називають *банком даних* (Бнд).

Класифікацію САПР здійснюють по ряду ознак, наприклад, по застосуванню, цільовому призначенню, масштабам (комплексності розв'язуваних задач), характеру базової підсистеми - ядра САПР.

По застосуванню найбільш представницькими й широко використовуваними є наступні групи САПР.

1. САПР для застосування в галузях загального машинобудування. Їх часто називають *машинобудівними САПР* або МСАД (Mechanical CAD) системами.
2. САПР в області радіоелектроніки: системи ЕСАД (Electronic CAD) або ЕДА (Electronic Design Automation).
3. САПР в області архітектури й будівництва.

Крім того, відомо велике число спеціалізованих САПР, або виділюваних у зазначених групах, або що представляють самостійну галузь у класифікації. Прикладами таких систем є САПР великих інтегральних схем (ВІС); САПР літальних апаратів; САПР електричних машин і т.п.

По цільовому призначенню розрізняють САПР або підсистеми САПР, що забезпечують різні аспекти проектування. Так, у складі МСАД з'являються САЕ/CAD/CAM системи.

По масштабах розрізняють окремі *програмно-методичні комплекси* (ПМК) САПР, наприклад, комплекс аналізу міцності механічних виробів відповідно до методу скінченних елементів (МСЕ) або комплекс аналізу електронних схем; системи ПМК; системи з унікальними архітектурами не тільки програмного (software), але й технічного (hardware) забезпечень.

По характеру базової підсистеми розрізняють наступні різновиди САПР.

1. САПР на базі підсистеми машинної графіки й геометричного моделювання. Ці САПР орієнтовані на застосування, де основною процедурою проектування є конструювання, тобто визначення просторових форм і взаємного розташування об'єктів. Тому до цієї групи систем

відноситься більшість САПР в області машинобудування, побудованих на базі графічних ядер. У цей час широко використовуються уніфіковані графічні ядра, застосовувані більш ніж в одній САПР, це ядра Parasolid фірми EDS Unigraphics і ACIS фірми Intergraph.

2. САПР на базі СКБД. Вони орієнтовані на застосування, у яких при порівняно нескладних математичних розрахунках переробляється великий об'єм даних. Такі САПР переважно зустрічаються в техніко-економічних галузях, наприклад, при проектуванні бізнес-планів, але мають місце також при проектуванні об'єктів, подібних до щитів керування в системах автоматики.

3. САПР на базі конкретного прикладного пакета. Фактично це автономно використовувані програмно-методичні комплекси, наприклад, імітаційного моделювання виробничих процесів, розрахунку міцності по методу скінченних елементів, синтезу й аналізу систем автоматичного керування й т.п. Часто такі САПР відносять до систем САЕ. Прикладами можуть служити програми логічного проектування на базі мови VHDL, математичні пакети типу MathCAD.

4. Комплексні (інтегровані) САПР, що складаються із сукупності підсистем попередніх видів. Характерними прикладами комплексних САПР є САЕ/CAD/CAM-системи в машинобудуванні або САПР БІС. Так, САПР БІС містить у собі СКБД і підсистеми проектування компонентів, принципів, логічних і функціональних схем, топології кристалів, тестів для перевірки придатності виробів. Для керування настільки складними системами застосовують спеціалізовані системні середовища.

3.3. Етапи проектування автоматизованих систем

Методики й засоби проектування автоматизованих систем (АС) і, зокрема, САПР підрозділяються на три великих рівні, що відповідають створенню інтегрованої корпоративної системи, окремих автоматизованих систем проектування й керування, програмно-методичних комплексів і компонентів автоматизованих систем.

На ринку програмного забезпечення пропонується ряд систем типу PLM, CPC і PDM. Існує ряд фірм, що спеціалізуються на розробці проектів системної інтеграції (наприклад, у Росії Jet Info, Consistent Software, Interface і ін.)

Для кожного класу АС (САПР, ERP, геоінформаційні системи й т.д.) можна вказати фірми, що спеціалізуються на розробці програмних (а іноді й програмно-апаратних) систем. Багато хто з них на основі однієї з базових технологій реалізують свій підхід до створення АС і дотримуються стратегії або тотального постачальника, або відкритості й розширення системи додатками й доповненнями третіх фірм.

Діє державний стандарт на стадії створення автоматизованих систем ДЕРЖСТАНДАРТ 34.601-90. Існує й міжнародний стандарт на стадії життєвого циклу програмної продукції (ISO 12207:1995). Як АС, так і компоненти АС є складними системами й при їхньому проектуванні потрібно використати один із стилів проектування:

- спадне проектування (Top-of-Design); чітка реалізація спадного проектування приводить до спіральної моделі розробки ПЗ, на кожному витку спирали блоки попереднього рівня деталізуються, використовуються зворотні зв'язки (альтернативою є каскадна модель, що відноситься до почергової реалізації частин системи);

- висхідне проектування (Bottom-of-Design);
- еволюційне проектування (Middle-of-Design).

Найчастіше застосовують спадний стиль блочно-ієрархічного проектування. Розглянемо етапи спадного проектування АС.

Верхній рівень проектування АС часто називають концептуальним проектуванням. Концептуальне проектування виконують у процесі передпроектних досліджень, формулювання ТЗ, розробки ескізного проекту й прототипування (відповідно ДО ДЕРЖСТАНДАРТУ, ці стадії називають формуванням вимог до АС, розробкою концепції АС і ескізний проект).

Передпроектні дослідження проводять шляхом аналізу (обстеження) діяльності підприємства (компанії, установи, офісу), на якому створюється або модернізується АС. При цьому потрібно одержати відповіді на питання: що не влаштовує в існуючій технології? Що можна поліпшити? Кому це потрібно й, отже, який буде ефект? Перед обстеженням формуються та у процесі його проведення уточнюються мети обстеження – визначення можливостей і ресурсів для підвищення ефективності функціонування підприємства на основі автоматизації процесів керування, проектування, документообігу й т.п. Зміст обстеження – виявлення структури підприємства, виконуваних функцій, інформаційних потоків, наявного досвіду й засобів автоматизації. Обстеження проводять системні аналітики (системні інтегратори) разом із представниками організації-замовника.

На основі аналізу результатів обстеження будують модель, що відбиває діяльність підприємства на даний момент (до реорганізації). Її називають *моделлю "As Is"*. Далі розробляють вихідну концепцію АС. Ця концепція містить у собі пропозиції по зміні структури підприємства, взаємодії підрозділів, інформаційним потокам, що виражається в *моделі "To Be"* (як повинне бути).

Результати аналізу конкретизуються в ТЗ на створення АС. У ТЗ вказують потоки вхідної інформації, типи вихідних документів і надаваних послуг, рівень захисту інформації, вимоги до продуктивності (пропускної здатності) і т.п. ТЗ направляють замовникові для обговорення й остаточного узгодження.

Ескізний проект (технічну пропозицію) представляють у вигляді проектної документації, що описує архітектуру системи, структуру її підсистем, склад модулів. Тут же вказуються пропозиції на вибір базових програмно-апаратних засобів, які повинні враховувати прогноз розвитку підприємства.

Відносно апаратних засобів і особливо ПЗ такий вибір найчастіше є вибір фірми-постачальника необхідних засобів (або, принаймні, базового ПЗ), тому що правильна спільна робота програм різних фірм досягається з великими складностями. У проекті може бути запропоновано кілька варіантів вибору. При аналізі з'ясовуються можливості покриття автоматизуємих функцій наявними програмними продуктами й, отже, обсяги робіт по розробці оригінального ПЗ. Подібний аналіз необхідний для попередньої оцінки тимчасових і матеріальних витрат на автоматизацію. Облік ресурсних обмежень дозволяє уточнити досяжні масштаби автоматизації, підрозділити проектування АС на роботи першої, другої і т.д. черги.

Після прийняття ескізного проекту розробляють прототип АС, що представляє собою набір програм, емулюючих роботу готової системи. Завдяки прототипуванню можна не тільки розроблювачам, але й майбутнім користувачам АС побачити контури й особливості системи й, отже, завчасно внести корективи в проект.

Як на етапі обстеження, так і на наступних етапах доцільно дотримуватися певної дисципліни фіксації й подання одержуваних результатів, заснованої на тій або іншій методиці формалізації специфікацій. Формалізація потрібна для однозначного розуміння виконавцями й замовником вимог, обмежень і прийнятих рішень.

При концептуальному проектуванні застосовують ряд специфікацій, серед яких центральне місце займають моделі перетворення, зберігання й передачі інформації. Моделі, отримані в процесі обстеження підприємства, є моделями його функціонування. У процесі розробки АС моделі, як правило, перетерплюють істотні зміни (перехід від "As Is" до "To Be") і в остаточному виді модель "To Be" розглядають як модель проектованої АС.

Розрізняють функціональні, інформаційні, поведінкові й структурні моделі. *Функціональна модель* системи описує сукупність виконуваних системою функцій. *Інформаційна модель* відбиває структури даних — їхній склад і взаємозв'язки. *Поведінкова модель* описує інформаційні процеси (динаміку функціонування), у ній фігурують такі категорії, як стан системи, подія, перехід з одного стану в інше, умови переходу, послідовність подій, здійснюється прив'язка вчасно. *Структурна модель* характеризує морфологію системи (її побудову) — склад підсистем, їхнього взаємозв'язку.

Змістом наступних етапів спадного проектування (відповідно до ДЕРЖСТАНДАРТ, це стадії розробки технічного проекту, робочої документації, запровадження в дію) є уточнення переліків устаткування, що здобуває, і готових програмних продуктів, побудова системного середовища, детальне інфологічне проектування БД і їхнього первісного наповнення, розробка власного оригінального ПЗ, що, у свою чергу, ділиться на ряд етапів спадного проектування. Ці роботи становлять зміст робочого проектування. Після цього виконуються закупівля й інсталяція програмно-апаратних засобів, впровадження й експериментальна експлуатація системи.

Особливе місце в ряді проектних задач займає розробка проекту корпоративної обчислювальної мережі, оскільки технічне забезпечення АС має мережну структуру.

Якщо територіально АС розташовується в одному будинку або в декількох близько розташованих будинках, то корпоративна мережа може бути виконана у вигляді сукупності декількох локальних підмереж зв'язаних опорною локальною мережею. Крім вибору типів підмереж, зв'язних протоколів і комутаційного устаткування доводиться вирішувати задачі розподілу вузлів по підмережах, виділення серверів, вибору мережного ПЗ, визначення способу керування даними в обраній схемі розподілених обчислень і т.п.

Якщо АС розташовується у вилучених друг від друга пунктах, зокрема, розташованих у різних містах, то вирішується питання про оренду каналів зв'язку для корпоративної мережі, оскільки альтернативний варіант використання виділеного каналу в більшості випадків виявляється неприйнятним через високу ціну. Природно, що при цьому насамперед розглядається можливість використання послуг Internet. Виникаючі при цьому проблеми пов'язані із забезпеченням інформаційної безпеки й надійності доставки повідомлень.

Лекція 4. Технічне забезпечення САПР

4.1. Вимоги до технічного забезпечення САПР

Технічне забезпечення САПР містить у собі різні технічні засоби (hardware), використовувані для виконання автоматизованого проектування, а саме обчислювальні системи, ЕОМ (комп'ютери), периферійні пристрої, мережне устаткування, а також устаткування деяких допоміжних систем (наприклад, вимірювальних), що підтримують проектування. Відзначимо, що *обчислювальною системою* (на відміну від ЕОМ і обчислювальної мережі) називають сукупність апаратних і програмних засобів, спільно використовуваних при рішенні задач і розташованих компактно на території, розміри якої порівнянні з розмірами апаратних засобів.

Технічні засоби, що використовуються в САПР, повинні забезпечувати:

- виконання всіх необхідних проектних процедур, для яких є відповідне ПЗ;
- взаємодію між проектувальниками й ЕОМ, підтримку інтерактивного режиму роботи;
- взаємодію між членами колективу, що виконують роботу над загальним проектом.

Перша із цих вимог виконується при наявності в САПР обчислювальних машин і систем з достатніми продуктивністю і ємністю пам'яті.

Друга вимога ставиться до користувальницького інтерфейсу й виконується за рахунок включення в САПР зручних засобів вводу-виводу даних і, насамперед, пристроїв обміну графічною інформацією.

Третя вимога обумовлює об'єднання апаратних засобів САПР в обчислювальну мережу.

У результаті загальна структура ТЗ САПР являє собою мережу вузлів, зв'язаних між собою середовищем передачі даних. Вузлами (станціями даних) є робочі місця проектувальників, що часто мають назву *автоматизованих робочих місць* (АРМ) або робочих станцій (WS — Workstation), ними можуть бути також великі ЕОМ (мейнфрейми), окремі периферійні й вимірювальні пристрої. Саме в АРМ повинні бути засоби для інтерфейсу проектувальника з ЕОМ. Що стосується обчислювальної потужності, то вона може бути розподілена між різними вузлами обчислювальної мережі.

Середовища передачі даних представлені каналами передачі даних, що складаються з ліній зв'язку й комутаційного устаткування.

4.2. Процесори ЕОМ

Призначення *процесора* — керування обчислювальним процесом і обробка даних відповідно до заданої програми. Процесори можуть бути універсальними, здатними вирішувати широке коло задач і спеціалізованими, орієнтованими на певне вузьке коло задач. Прикладом універсальних процесорів можуть служити процесори Pentium компанії Intel, а до спеціалізованих процесорів відносяться процесори графічні, сигнальні, вводу/виводу й ін.

Основна характеристика процесора — продуктивність або *швидкодія*. Використають кілька різних показників продуктивності. Найбільш очевидним і історично першим показником продуктивності є швидкодія, вимірювана числом команд комп'ютера, виконуваних в одиницю часу, тобто вимірюване в Mips (1 Mips відповідає 10^6 команд у секунду). При рішенні науково-технічних задач характерне використання чисел із плаваючою крапкою, у цьому випадку більш справедлива оцінка продуктивності в Mflops, тобто числом операцій над числами із плаваючою крапкою в одиницю часу.

Однак оцінки в Mips або Mflops залежать не тільки від властивостей комп'ютера, але й від особливостей розв'язуваної задачі. Тому використовують і інші показники продуктивності. Так, швидкодію ПК звичайно характеризують тактовою частотою комп'ютера. У деяких випадках швидкодію оцінюють часом, витраченим на рішення деяких тестових задач. Такими тестовими задачами є LINPACK, SPECint92, SPECfp92 і деякі інших. Наприклад, LINPACK — це пакет програм для рішення систем лінійних алгебраїчних рівнянь. У набір базових підпрограм в LINPACK входять підпрограми скалярного добутку векторів, множення вектора на скаляр, додавання векторів.

Лекція 5. Пам'ять ЕОМ

Пам'ять комп'ютера призначена для зберігання інформації й характеризується наступними основними параметрами:

- об'ємом (ємністю), тобто максимально можливим числом блоків даних, розташовуваних у пам'яті;
- швидкодією, що характеризується середнім часом звертання до пам'яті (середнім часом пошуку, зчитування й/або запису одного блоку даних);
- ціною, віднесеної до одного блоку збережених даних.

Пам'ять комп'ютера має ієрархічну структуру у зв'язку з тим, що в єдиному запам'ятовувальному пристрої не вдається одночасно належною мірою задовольнити вимоги великого об'єму пам'яті й високої швидкодії. Тому звичайно до складу процесора включають швидкодіючу *кеш-пам'ять* порівняно малого об'єму (часто також поділювану на два або три рівні), далі в міру росту об'єму й часу звертання до пам'яті виділяють *оперативну пам'ять* і *зовнішню пам'ять*.

Фізично кеш і оперативна пам'ять у сучасних комп'ютерах, як правило, є напівпровідниковими. Для реалізації зовнішньої пам'яті використовують магнітні й оптичні принципи запису й читання інформації.

Залежно від складу виконуваних операцій розрізняють кілька типів пам'яті. *Пам'ять із довільним доступом* (операціями як читання, так і запису) звичайно позначають RAM (Random Access Memory), а *постійну пам'ять*, призначену тільки для читання, — ROM (Read Only Memory).

В якості елементів оперативної пам'яті використовують осередки, що представляють собою конденсатори. Заряджений конденсатор зберігає "1", розряджений — "0". Під час зчитування інформації конденсатори розряджаються. Крім того, заряд у конденсаторі через витік зберігається обмежений час (декілька мілісекунд). Тому необхідна підзарядка, що виконується в процесі регенерації інформації. Ця обставина обумовила назву пам'яті подібного типу — *динамічна пам'ять* або DRAM (Dynamic RAM). Малі розміри й простота елементів DRAM дозволяють одержати досить велику ємність пам'яті, але через витрати часу на регенерацію знижується швидкодія.

З метою підвищення швидкодії DRAM розроблено кілька модифікацій цього типу пам'яті. *Синхронна пам'ять* типу SDRAM (Synchronous DRAM) відрізняється від асинхронної пам'яті типу DRAM тим, що такти роботи пам'яті засинхронізовані з тактами роботи процесора. Це дозволяє виключити цикли очікування, що мають місце в DRAM.

У пам'яті типу DDR SDRAM (Double Date Rate SDRAM) швидкодію вдалося збільшити вдвічі в порівнянні з попередніми типами DRAM за рахунок того, що звертання до пам'яті відбуваються двічі за такт - як по передньому, так і по задньому фронту тактових сигналів.

У більш швидкій *статичній пам'яті* SRAM (Static RAM) елементами пам'яті є бістабільні осередки, при зчитуванні інформація не губиться й регенерації не потрібно. Тому SRAM більш швидкодіюча пам'ять і використовується як кеш-пам'ять. Однак SRAM значно дорожче DRAM.

Пам'ять ROM може бути однократно програмувальною, тобто дозволяти лише однократний первісний запис інформації, або бути *перепрограмувальною пам'яттю*. В останньому випадку розрізняють види пам'яті EPROM (Electrically Programmable ROM) і EEPROM (Electrically Erasable Programmable ROM), що називається також флеш-памятью. Властивість репрограмуємості заснована на використанні Моп-транзисторів із двома затворами – основним і схованим у шарі діелектрика. Подача імпульсу підвищеної напруги на Моп-транзистори в осередках, які повинні зберігати нулі, приводить до появи на схованому

затворі електричного заряду, що перешкоджає утворенню провідного каналу, що й відповідає зберіганню нуля. Стан нуля зберігається практично необмежений час, але при перепрограмуванні заряд ліквідується за допомогою зовнішнього ультрафіолетового випромінювання (в EPROM) або електричного імпульсу, що подається на затвори (в EEPROM). Ультрафіолетове випромінювання приводить до поступової зміни властивостей напівпровідника, тому пам'ять EPROM допускає лише обмежене число перепрограмувань, крім того, перезапис відбувається з порівняно більшими витратами часу. Тому в цей час основна увага приділяється розвитку пам'яті типу EEPROM.

Прикладом використання ROM може служити пам'ять ROM BIOS (ROM Basic Input Output System) у ПК, що зберігає дані для керування стандартними зовнішніми пристроями.

Іноді говорять про пам'ять типу ECC. Це тип пам'яті, у якій реалізований режим ECC (Error Control Correction) контролю функціонування пам'яті з відновленням помилок. У цьому режимі виправляються одиночні помилки в групах з 8 байтів.

Зовнішня пам'ять, реалізована на магнітних носіях, представлена *накопичувачами на магнітних дисках* і магнітних стрічках. Розрізняють *накопичувачі на твердих магнітних дисках* (вінчестери) і *накопичувачі на гнучких магнітних дисках*. У них зберігають багаторазово використовувані програми й файли даних. Накопичувачі на магнітних стрічках, що мають низьку вартість зберігання, але порівняно великий час пошуку інформації, в основному використовують для архівів даних.

Для підвищення швидкодії зовнішньої пам'яті й відмовостійкості застосовують підсистеми зовнішньої пам'яті з високою готовністю, що реалізують технологію надлишкових масивів дисків RAID (Redundant Arrays of Inexpensive Disks), об'єднаних в один блок із загальним інтелектуальним контролером. RAID є спосіб організації роботи декількох фізичних дисків аналогічно одному логічному диску, при якому дисковий масив представляється прикладній задачі як один диск. Як правило, масиви RAID використовуються в серверах для забезпечення надійності за рахунок дублювання даних.

5.1. Монітори

У більшості сучасних *моніторів* використовуються електронно-променеві трубки, у яких світіння шару люмінофора, яким покритий екран, відбувається під дією його опромінення потоком електронів, що випускаються нагрітим катодом трубки. У растрових дисплеях електронний промінь під дією відхиляючої системи трубки переміщається по горизонталі (рядкове розгорнення) і вертикалі (кадрове розгорнення), викликаючи засвічення всіх точок (пікселей) екрану. У кольорових моніторах є три промені, а кожний піксель екрана складається із трьох близько розташованих точок, що мають відповідно світіння червоного, зеленого й синього (RGB) кольору. Інтенсивність потоку, а, отже, і яскравість світіння регулюються напругою, що подається на керуючі електроди. Кольори світіння кожного пікселя визначається яскравістю світіння всіх трьох точок пікселя, тому, управляючи інтенсивністю потоків електронів у всіх трьох променях, можна одержати множину різних відтінків світіння.

Пам'ять відеосистеми має стільки осередків, скільки повинне бути пікселей зображення, і в кожному осередку записані три n -розрядних двійкових числа, що задають інтенсивність RGB променів. Ці 2^n значень перетворюються в електричний струм (напругу), що подається на керуючі електроди. Розгорнення засинхронізоване із читанням даних з осередків пам'яті, що й створює необхідне зображення на екрані.

Основними параметрами моніторів є розгортка (число пікселей), смуга пропускання відеопідсилювача, частоти кадрового й рядкового розгорнень. Сучасні монітори можуть підтримувати частоту кадрового розгорнення 100 Гц (чим вище частота, тим менше негативний вплив мерехтіння екрана) і розгортку до 1600x1200 пікселей.

Поряд з дисплеями на електронно-променевих трубках все більше поширення одержують рідкокристалічні дисплеї (LCD — Liquid Cristal Display). Екран LCD-дисплея являє собою простір між двома скляними пластинами, заповненими рідкими кристалами. Рідкі кристали здатні під дією електричного поля змінювати свою орієнтацію й тим самим змінювати свою здатність відбивати світло, створюючи необхідне зображення.

З'явилися перші зразки стереоскопічних моніторів, що дозволяють без додаткових стереоокулярів спостерігати тривимірні зображення.

5.2. Периферійні пристрої

Периферійні пристрої комп'ютерів ділять на пристрої вводу, пристрою виводу й зовнішні запам'ятовувальні пристрої. До *пристроїв вводу* відносяться клавіатура, миша, сканер, дигітайзер. *Пристрої виводу* представлені принтерами, плотерами (графобудівниками), дисплеями.

Звичайно в автоматизованих системах користувачі працюють на *автоматизованих робочих місцях* (АРМ), на яких є комп'ютер з виходом у корпоративну мережу й необхідні периферійні пристрої. Типовий склад пристроїв АРМ: ПК або робоча станція з вінчестером; пристрої вводу-виводу, що включають, як мінімум, клавіатуру, мишу, дисплей; додатково до складу АРМ можуть входити принтер, сканер, плотер і деякі інші периферійні пристрої.

АРМ розрізняються своєю орієнтацією на певні групи проектних або бізнес процедур і відповідно типами й характеристиками обчислювального та периферійного устаткування. Так, в АРМ, призначених для САПР, що вимагають обробки графічної інформації, у дисплеях використовуються растрові монітори з кольоровими трубками. Типові значення характеристик моніторів перебувають у наступних межах: розмір екрана по діагоналі 17...24 дюйма (фактично зображення займає площу на 5...8 % менше, ніж вказується в паспортних даних). Розгортка монітора, тобто число помітних пікселів (окремих точок, з яких складається зображення), визначається кроком між отворами в масці, через які проходить до екрана електронний промінь в електронно-променевій трубці. Цей крок перебуває в межах 0,21...0,28 мм, що відповідає кількості пікселів зображення від 800x600 до 1600x1200 і більше. Чим вище розгортка, тим ширше повинна бути смуга пропущення електронних блоків відеосистеми при однаковій частоті кадрового розгорнення. Смуга пропущення відеопідсилювача перебуває в межах 110...150 МГц і тому частота кадрового розгорнення звичайно знижується з 135 Гц для 640x480 до 60 Гц для 1600x1200. Відзначимо, що чим нижче частота кадрового розгорнення, а це є частота регенерації зображення, тим помітніше мерехтіння екрана. Бажано, щоб ця частота була не нижче 75 Гц.

Для уведення інформації в ЕОМ використовують дигітайзери, інтерактивні дошки, сканери, візуалізатори, цифрові відеокамери (Web-камери).

Дигітайзери призначені для ручного уведення графічної інформації, у цей час їх застосовують досить рідко. Дигітайзер має вигляд кульмана, по його електронній дошці переміщається курсор, на якому розташований візир і кнопкова панель. Курсор має електромагнітний зв'язок із сіткою провідників в електронній дошці. При натисканні кнопки в деякій позиції курсору відбувається занесення інформації про координати цієї позиції. У такий спосіб може здійснюватися ручний ввід креслень.

Інтерактивна дошка (smart board), називана також інтелектуальною дошкою, дозволяє користувачам виводити на поверхню дошки вміст файлів з пам'яті комп'ютера, Internet-джерел, CD-ROM, редагувати виведене зображення, наносити поверх нього свої графічні й текстові фрагменти. Усе, що користувач намалював або написав на дошці, буде збережено у вигляді комп'ютерних файлів, може бути роздруковано, послано по електронній пошті, поміщено в Web-сторінку.

Інтерактивні дошки є зручним засобом подання, редагування й збереження навчальних матеріалів, формованих викладачем безпосередньо під час проведення навчальних занять. Крім того, за допомогою інтерактивної дошки викладач може управляти комп'ютером, вибираючи потрібні програми або файли простою вказівкою на екранну кнопку на поверхню дошки.

Для автоматичного уведення інформації з наявних текстових або графічних документів використовують *сканери* планшетного або протяжливого типу й візуалізатори. Спосіб зчитування – оптичний. У скануючій голівці сканера розміщуються оптоволоконні самофокусуєчі лінзи й фотоелементи. Розгортка у різних моделях становить від 300 до 2400 точок на дюйм (цей параметр часто позначають dpi). Зчитана інформація має растрову форму, програмне забезпечення сканера представляє її в одному зі стандартних форматів, наприклад tiff, gif, psx, jpeg, і для подальшої обробки може бути виконана *векторизація* – переклад графічної інформації у векторну форму, наприклад, у формат dxf.

Візуалізатор служить для зчитування інформації з документа і її передачі на вхід комп'ютера для збереження в пам'яті (режим сканера) або на вхід мультимедійного проектора для збільшеного зображення на екрані під час презентацій або навчальних занять. Як документ можуть використатися не тільки плоскі сторінки з текстом, малюнками, кресленнями, але й інші предмети, наприклад, деталі механічних або електронних пристроїв порівняно невеликих розмірів. Можна накладати зображення з візуалізатора на інші зображення, що надходять із комп'ютера.

Для зйомки відеофрагментів з метою їхнього наступного включення в мультимедійні навчальні матеріали використовують цифрові *відеокамери*.

Для висновку інформації у вигляді твердих копій застосовують *принтери* й *плотери* (графобудівники). Перші з них орієнтовані на одержання документів малого формату (A3, A4), другі – для висновку графічної інформації на широкоформатні носії.

У цих пристроях переважно використовується растровий (тобто порядковий) спосіб виводу із струминною технологією печаті. Друкуюча система в струминних пристроях містить у собі картридж і голівку. Картридж – балон, заповнений чорнилом (у кольорових пристроях є кілька картриджів, кожний із чорнилом своїх кольорів). Голівка – матриця із сопел, з яких дрібні чорнильні краплі надходять на носій. Фізичний принцип дії голівки термічний або п'єзоелектричний. При термопечаті викидання крапель із сопла відбувається під дією його нагрівання, що викликає утворення пари й викидання крапельок під тиском. При п'єзоелектричному способі пропускання струму через п'єзоелемент приводить до зміни розміру сопла й викиданню краплі чорнила. Другий спосіб дорожче, але дозволяє одержати більш високоякісне зображення.

Типова розгортка принтерів і плотерів перебуває в межах від 300 до 2400 dpi. Швидкість печатки в монохромному режимі від 10 до 18 сторінок у хвилину, у кольоровому режимі – трохи менше.

Дигітайзери, сканери, принтери, плотери можуть входити до складу АРМ або розділятися користувачами декількох робочих станцій у складі локальної обчислювальної мережі.

Вивід інформації на екран збільшених розмірів, що потрібно під час презентацій і навчальних занять, виконується за допомогою мультимедійних проекторів. Вони відрізняються від відеопроєкторів тим, що сприймають не відеосигнали, а цифрові сигнали від комп'ютера.

5.3. Шини комп'ютера

Зв'язок різних пристроїв комп'ютера між собою здійснюється за допомогою *шин*. До шин

пред'являються вимоги високої швидкодії (пропускної здатності), високій надійності й малій вартості. Пропускна здатність шини визначається обсягом інформації, переданої по шині в одиницю часу, і дорівнює добутку тактової частоти шини на її розрядність.

У найпростішому випадку комп'ютер може мати єдину шину, що має назву *системної*. Однак застосування єдиної шини не дозволяє запаралелити операції обміну, крім того, для ряду пристроїв виявляються не погодженими параметри швидкодії пристрою й шини. Тому звичайно використовують ієрархію шин. У сучасних ПК є *системна шина*, що зв'язує процесор з оперативною пам'яттю (і можливо з кеш-пам'яттю другого рівня), *шина розширення* для підключення мережного контролера й швидких зовнішніх пристроїв і шина повільних пристроїв, таких як клавіатура, миша, принтер і т.п. Часто окрема шина, більш швидкодіюча чим шина розширення, зв'язує відеоадаптер з оперативною пам'яттю. Наприклад, 64-розрядна системна шина в мікропроцесорі Itanium має тактову частоту 133 МГц, а шина мікропроцесора Itanium II є вже 128-розрядної й працює із частотою 400 МГц. Шина розширення PCI може бути 32- або 64-розрядної. Її тактова частота звичайно дорівнює половині тактової частоти системної шини. Шини зв'язані між собою мікросхемами-мостами. Крім того, усередині кристала мікропроцесора є процесорна швидкодіюча шина, що зв'язує арифметико-логічний пристрій з кеш-пам'яттю першого рівня.

Шини можуть бути декількох профілів залежно від кількості провідних пристроїв, тобто пристроїв, які можуть ініціювати транзакції зчитування або запису. При декількох провідних пристроях надання повноважень на захоплення шини здійснює схема арбітражу.

Розрізняють синхронні й асинхронні шини. У синхронних шинах передбачається, що тактові сигнали зв'язуючих пристроїв засинхронізовані, тому можлива більша швидкодія, але при істотному обмеженні довжини шини. В асинхронних шинах використовується старто-стопний режим передачі з попереднім запитом зв'язку й згодою на нього.

Лекція 6. Типи обчислювальних машин і систем

Комп'ютери (ЕОМ) і обчислювальні системи (ОС) прийнято класифікувати по ряду ознак.

Залежно від продуктивності й вартості обчислювального встаткування виділяють кілька типів ЕОМ і ОС, причому різні покоління обчислювальної техніки мали свою шкалу типів. Так, до середини 80-х років минулого століття ЕОМ ділили на мікро-ЕОМ, міні-ЕОМ, великі ЕОМ (ЕОМ високої продуктивності) і суперЕОМ. У наш час ЕОМ і ОС підрозділяють на персональні комп'ютери, робочі станції, сервери, мейнфрейми, кластери й суперкомп'ютери.

Найбільше поширення в САПР одержали персональні комп'ютери (ПК), робочі станції й сервери.

Персональні комп'ютери з'явилися на початку 80-х років минулого століття в результаті трансформації міні-ЕОМ у порівняно дешеві настільні системи для індивідуального використання, завдяки розвитку елементної бази обчислювальної техніки, що привело до створення великих (ВІС) і надвеликих (НВІС) інтегральних схем.

Робочі станції являють собою обчислювальні системи, орієнтовані на рішення деяких задач, насамперед в автоматизованих системах проектування (САПР). Комп'ютери в САПР орієнтовані на рішення складних проектних задач, що обумовлює підвищені швидкодію й об'єм пам'яті, а також розширені можливості обробки й візуалізації графічної інформації робочих станцій у порівнянні з персональними комп'ютерами. Тому звичайно робочі станції мають більш складну структуру й більш дорогі пристрої, чим ПК.

В обчислювальних мережах виконання функцій, пов'язаних з обслуговуванням всіх вузлів мережі, покладається на *сервери*. По функціональному призначенню розрізняють сервери файлові, баз даних, комутаційні, прикладні, поштові й т.п. Сервери, як правило,

повинні мати більшу швидкість, надійність й у багатьох випадках збільшену ємність пам'яті в порівнянні з комп'ютерами в клієнтських вузлах.

Мейнфреймами називають великі ЕОМ. Висока продуктивність і велика ємність пам'яті забезпечують рішення складних проблем, дозволяють використовувати такі комп'ютери як центральний вузол ОС, що управляє роботою багатьох простих терміналів. Однак по відношенню продуктивність/ціна мейнфрейми звичайно помітно уступають попереднім типам ОС, їхнє використання в цей час досить обмежене.

Кластер — це розподілена система комп'ютерів, що функціонує як єдина система із загальними ресурсами. Основна мета, що обумовила появу кластерів, — збереження працездатності ОС шляхом перерозподілу навантаження при виході з ладу частини ресурсів. Крім того, кластеризація — один із шляхів підвищення продуктивності за рахунок спільного використання багатьох комп'ютерів. Кластери дозволяють нарощувати обчислювальну потужність, оскільки легко масштабуються.

Комп'ютери, що характеризуються найбільшими значеннями продуктивності й ціни серед інших типів ЕОМ і ОС, відносять до категорії *суперкомп'ютерів*. Так, суперкомп'ютер ASCI White SP Power3, створений в ІВМ і встановлений у Ліверморській національній лабораторії США, що був ще в 2001 р. світовим лідером по продуктивності, включає 8192 процесорів. Його продуктивність (по тесту Linpack) оцінюється в 7226 Gflops, пікова продуктивність — в 12288 Gflops. Устаткування цього суперкомп'ютера займає площу, рівну площі двох баскетбольних площадок. Очевидно, що це унікальні обчислювальні системи, що найчастіше використовуються в наукових і освітніх установах для рішення складних задач при наукових дослідженнях.

4.1. Класифікація обчислювальних систем по співвідношенню потоків команд і даних

Популярна класифікація ОС, заснована на архітектурних особливостях систем і запропонована Флином. Відповідно до неї виділяють чотири класи систем: *SISD* (Single Instruction Single Data — один потік команд і один потік даних), *SIMD* (Single Instruction Multi Data — один потік команд і багато потоків даних), *MISD* (Multi Instruction Single Data — багато потоків команд і один потік даних), *MIMD* (Multi Instruction Multi Data — багато потоків команд і багато потоків даних).

Класу *SISD* відповідають однопроцесорні ЕОМ без застосування розпаралелювання обчислювального процесу. Клас *MISD* не знайшов практичного застосування. У сучасних високопродуктивних ОС застосовують класи *SIMD* і *MIMD*.

В *SIMD* системах та сама машинна команда виконується одночасно над різними числами, що характерно, наприклад, для матрично-векторних обчислень. Зокрема, до цього класу належать однопроцесорні векторні й векторно-конвейерні ОС (наприклад, Cray-1, Hitachi S3600), а також ОС на основі матриць процесорів.

Більшість сучасних суперкомп'ютерів має структуру *MIMD*. У цьому класі розрізняють кілька підкласів.

По-перше, це системи *SM* (Shared Memory), що розділяють загальну пам'ять, тобто що мають загальний адресний простір. До цього підкласу відносяться багатопроцесорні векторні ОС і системи *SMP* (Symmetric Multi-Processor). Очевидно, що в таких системах жорсткі вимоги пред'являються до швидкодії каналів зв'язку процесорів з загальною пам'яттю. У векторних ОС використовується високошвидкісна шина між оперативною пам'яттю й векторними регістрами, у системах *SMP* висока продуктивність підтримується за рахунок введення кеш-пам'яті великого об'єму, тобто вимоги до пропускну здатності тракту "оперативна пам'ять - кеш-пам'ять" істотно знижуються.

По-друге, системи з *розподіленою пам'яттю* DM (Distributed Memory) — це системи, у яких кожний процесор має свою оперативну пам'ять. Прикладом систем DM можуть служити кластерні системи. У свою чергу, кожний вузол може бути системою типу SIMD або MIMD. Тоді говорять про системи *DM-SIMD* або *DM-MIMD* відповідно. Якщо операційна система працює з єдиним простором адрес, а кожний процесор виконаний як система MIMD, то маємо систему *SM-MIMD*.

По-третє, системи з *масовим паралелізмом* MPP (Massively Parallel Processing), у яких кілька копій однієї програми паралельно виконуються в різних вузлах з різними даними. Відмітна ознака MPP – помітно більше число процесорів, ніж у системах SMP, що ще більше ускладнює проблему ефективного розпаралелювання. Для користувача системи MPP можуть виглядати як SM-MIMD.

Лекція 7. Математичне забезпечення САПР

7.1. Вимоги до математичних моделей і методів у САПР

Основними вимогами до математичних моделей є вимоги адекватності, точності, економічності.

Модель завжди лише приблизно відбиває деякі властивості об'єкта. Адекватність має місце, якщо модель відбиває задані властивості об'єкта із прийнятною точністю. Під точністю розуміють ступінь відповідності оцінок однойменних властивостей об'єкта й моделі.

Економічність (обчислювальна ефективність) визначається витратами ресурсів, необхідних для реалізації моделі. Оскільки в САПР використовуються математичні моделі, далі мова йтиме про характеристики саме математичних моделей, і економічність буде характеризуватися витратами машинних часу й пам'яті.

Адекватність оцінюється переліком відбиваємих властивостей і областями адекватності. Область адекватності – область у просторі параметрів, у межах якої погрішності моделі залишаються в припустимих межах. Наприклад, область адекватності лінеаризованої моделі поверхні деталі визначається системою нерівностей:

$$\max |\varepsilon_{ij}| \leq \varepsilon_{\text{доп}},$$

де $\varepsilon_{ij} = \frac{x_{ij\text{ист}} - x_{ij\text{мод}}}{x_{ij\text{ист}}}$, $x_{ij\text{ист}}$ і $x_{ij\text{мод}}$ — i -я координата J -й точки поверхні в об'єкті й моделі відповідно; ε_{ij} і $\varepsilon_{\text{доп}}$ — допущена й гранично припустима відносні погрішності моделювання поверхні, максимум береться по всіх координатах і контрольованих точках.

Відзначимо, що в більшості випадків області адекватності будуються в просторі зовнішніх змінних

Аналогічні вимоги по точності й економічності фігурують при виборі чисельних методів рішення рівнянь моделі.

7.2. Вихідні рівняння для формування моделей на макрорівні

Вихідний математичний опис процесів в об'єктах на макрорівні представлено системами звичайних диференціальних і алгебраїчних рівнянь. Аналітичні рішення таких систем при типових значеннях їхніх порядків у практичних задачах одержати не вдається, тому в САПР

переважно використовуються алгоритмічні моделі. У цьому параграфі викладений узагальнений підхід до формування алгоритмічних моделей на макрорівні, справедливий для більшості застосувань.

Вихідними для формування математичних моделей об'єктів на макрорівні є компонентні й топологічні рівняння.

Компонентними рівняннями називають рівняння, що описують властивості елементів (компонентів), інакше кажучи, математична модель елемента (ММЕ) представляється компонентними рівняннями.

Топологічні рівняння описують взаємозв'язки в складі системи, що моделюється.

У сукупності компонентні й топологічні рівняння конкретної фізичної системи являють собою вихідну математичну модель системи (ММС).

Очевидно, що компонентні й топологічні рівняння в системах різної фізичної природи відбивають різні фізичні властивості, але можуть мати однаковий формальний вид. Однакова форма запису математичних співвідношень дозволяє говорити про формальні аналогії компонентних і топологічних рівнянь. Такі аналогії існують для механічних поступальних, механічних обертальних, електричних, гідравлічних (пневматичних), теплових об'єктів. Наявність аналогій приводить до практично важливого висновку: значна частина алгоритмів формування й дослідження моделей у САПР виявляється інваріантною й може бути застосована до аналізу проєктованих об'єктів у різних предметних областях. Єдність математичного апарата формування ММС особливо зручно при аналізі систем, що складаються з фізично різномірних підсистем.

У перерахованих вище застосуваннях компонентні рівняння мають вигляд:

$$\mathbf{F}_k\left(\frac{d\mathbf{V}}{dt}, \mathbf{V}, t\right) = 0, \quad (3.1)$$

топологічні рівняння:

$$\mathbf{F}_T(\mathbf{V}) = 0, \quad (3.2)$$

де $\mathbf{V} = (v_1, v_2, \dots, v_n)$ — вектор фазових змінних, t — час.

Розрізняють фазові змінні двох типів, їхні узагальнені найменування — фазові змінні типу потенціалу (наприклад, електрична напруга) і фазові змінні типу потоку (наприклад, електричний струм). Кожне компонентне рівняння характеризує зв'язки між різнотипними фазовими змінними, стосовними до одного компонента (наприклад, закон Ома описує зв'язок між напругою й струмом у резисторі), а топологічне рівняння — зв'язку між однотипними фазовими змінними в різних компонентах.

Моделі можна представляти у вигляді систем рівнянь або в графічній формі, якщо між цими формами встановлена взаємно однозначна відповідність. Як графічна форма часто використовують еквівалентні схеми.

Нижче розглянемо приклади компонентних і топологічних рівнянь для різних типів систем.

Лекція 8. Механічні системи

Фазовими змінними в механічних поступальних системах є сили й швидкості. Використовують одну із двох можливих електромеханічних аналогій. Надалі будемо використати ту з них, у якій швидкість відносять до фазових змінних типу потенціалу, а силу вважають фазовою змінною типу потоку. З огляду на формальний характер подібних аналогій, рівною мірою можна застосовувати й протилежну термінологію.

Компонентне рівняння, що характеризує інерційні властивості тіл, у силу другого закону Ньютона має вигляд:

$$F = M \frac{du}{dt}, \quad (3.3)$$

де F — сила; M — маса; u — поступальна швидкість.

Пружні властивості тіл описуються компонентним рівнянням, яке можна одержати з рівняння закону Гука. В одновірному випадку (якщо розглядаються поздовжні деформації пружного стрижня):

$$G = E \varepsilon, \quad (3.4)$$

де G — механічна напруга; E — модуль пружності; $\varepsilon = \frac{\Delta l}{l}$ — відносна деформація; Δl — зміна довжини l пружного тіла під впливом G . З огляду на, що $G = \frac{F}{S}$, де F — сила, S — площа поперечного переріза тіла, і диференціюючи (3.4), маємо:

$$\frac{dF}{dt} = \frac{SE}{l} \frac{d(\Delta l)}{dt}$$

або

$$\frac{dF}{dt} = g u, \quad (3.5)$$

де $g = \frac{SE}{l}$ — жорсткість (величину, зворотню жорсткості, іноді називають гнучкістю L_M),
 $u = \frac{d(\Delta l)}{dt}$ — швидкість.

Дисипативні властивості в механічних системах твердих тіл виражаються співвідношеннями, що характеризують зв'язок між силою тертя й швидкістю взаємного переміщення тіл, причому в цих співвідношеннях похідні сил або швидкостей не фігурують.

Топологічні рівняння характеризують, по-перше, закон рівноваги сил: сума сил, прикладених до тіла, включаючи силу інерції, дорівнює нулю (принцип Даламбера), по-друге, закон швидкостей, відповідно до якого сума відносної, переносної й абсолютної швидкостей дорівнює нулю.

У механічних обертальних системах справедливі компонентні й топологічні рівняння поступальних систем із заміною поступальних швидкостей на кутові, сил – на обертальні моменти, мас – на моменти інерції, жорсткостей – на обертальні жорсткості.

Є істотна відмінність у моделюванні електричних і механічних систем: перші з них одновірні, а процеси в других часто доводиться розглядати у двох- (2D) або тривимірному (3D) просторі. Отже, при моделюванні механічних систем у загальному випадку в просторі 3D потрібно використовувати векторне зображення фазових змінних, кожна з яких має шість складових, відповідним шести ступеням свободи.

Однак відзначені вище аналогії залишаються справедливими, якщо їх відносити до проєкцій сил і швидкостей на кожен просторову вісь, а при графічному зображенні моделей використовувати шість еквівалентних схем – три для поступальних складових і три для обертальних.

,

8.1. Гідравлічні системи

Фазовими змінними в гідравлічних системах є витрати й тиски. Як і в попередньому випадку, компонентні рівняння описують властивості рідини розсіювати або накопичувати енергію.

Розглянемо компонентні рівняння для рідини на лінійній ділянці трубопроводу довжиною Δl й скористаємося рівнянням Нав'є-Стоксу в наступній його формі (для ламінарної течії рідини):

$$\rho \frac{\partial U}{\partial t} = - \frac{\partial P}{\partial x} - 2aU,$$

де ρ — щільність рідини; U — швидкість; P — тиск; a — коефіцієнт лінеаризованого в'язкого тертя. Враховуючи, що $U = \frac{Q}{S}$, де Q — об'ємна витрата; S — площа поперечного перерізу трубопроводу, та, замінюючи просторову похідну відношенням кінцевих різностей, маємо:

$$\frac{\partial Q}{\partial t} = \frac{S \Delta P}{\Delta l \rho} - \frac{2aQ}{\rho},$$

або

$$\Delta P = L_r \frac{\partial Q}{\partial t} + R_r Q. \quad (3.6)$$

Тут ΔP — падіння тиску на розглянутій ділянці трубопроводу; $L_r = \frac{\Delta l \rho}{S}$ — гідравлічна індуктивність, що відображає інерційні властивості рідини; $R_r = \frac{2a \Delta l}{S}$ — гідравлічний опір, що відображає в'язке тертя.

Явище стискальності рідини описується компонентним рівнянням, що впливає із закону Гука:

$$\Delta P = \frac{E \Delta l}{l}. \quad (3.7)$$

Диференціюючи (3.7) і з огляду на, що об'ємна витрата Q зв'язана зі швидкістю співвідношенням $Q = US$, одержуємо: $U = \frac{\partial(\Delta l)}{\partial t}$

$$\frac{\partial \Delta P}{\partial t} = C_r Q,$$

де $C_r = \frac{E}{Sl}$ — гідравлічна ємність.

8.2. Вибір методів аналізу в часовій області

Аналіз процесів у проєктованих об'єктах можна робити в часовій і частотній областях. *Аналіз у часовій області* (динамічний аналіз) дозволяє одержати картину перехідних процесів, оцінити динамічні властивості об'єкта, він є важливою процедурою при дослідженні як лінійних, так і нелінійних систем.

Методи аналізу в часовій області, використовувані в універсальних програмах аналізу в САПР, — це чисельні методи інтегрування систем звичайних диференціальних рівнянь (СЗДР):

$$\mathbf{F}\left(\frac{d\mathbf{V}}{dt}, \mathbf{V}, t\right) = 0.$$

Інакше кажучи, це методи алгебраїзації диференціальних рівнянь. Формули інтегрування СЗДР можуть входити в математичну модель (ММ) незалежно від компонентних рівнянь або бути інтегрованими в ММ компонентів, як це виконано у вузловому методі.

Від вибору методу рішення СЗДР істотно залежать такі характеристики аналізу, як точність і обчислювальна ефективність. Ці характеристики визначаються насамперед типом і

порядком обраного методу інтегрування СЗДР.

Застосовують два типи методів інтегрування — *явні методи* (інакше екстраполяційні або методи, засновані на формулах інтегрування вперед), і *неявні методи* (інтерполяційні, засновані на формулах інтегрування назад). Розходження між ними зручно показати на прикладі найпростіших методів першого порядку — *методів Ейлера*.

Формула явного методу Ейлера являє собою наступну формулу заміни похідних у точці

t_n :

$$\left. \frac{d\mathbf{V}}{dt} \right|_n = \frac{\mathbf{V}_{n+1} - \mathbf{V}_n}{h_n}.$$

Тут індекс дорівнює номеру кроку інтегрування; $h_n = t_{n+1} - t_n$ — розмір кроку інтегрування (звичайно h_n називають просто кроком інтегрування). У формулі неявного методу Ейлера використане диференціювання назад:

$$\left. \frac{d\mathbf{V}}{dt} \right|_n = \frac{\mathbf{V}_n - \mathbf{V}_{n-1}}{h_n}.$$

де $h_n = t_n - t_{n-1}$.

Виконаємо порівняльний аналіз явних і неявних методів на прикладі модельної задачі:

$$\frac{d\mathbf{V}}{dt} = \mathbf{A}\mathbf{V} \quad (3.8)$$

при ненульових початкових умовах $\mathbf{V}_0 \neq 0$ і при використанні методів Ейлера з постійним кроком h . Тут \mathbf{A} — постійна матриця; \mathbf{V} — вектор фазових змінних.

При алгебраїзації явним методом маємо

$$\left(\frac{\mathbf{V}_{n+1} - \mathbf{V}_n}{h} \right) = \mathbf{A}\mathbf{V}_n$$

або

$$\mathbf{V}_{n+1} = (\mathbf{E} + h\mathbf{A})\mathbf{V}_n,$$

де \mathbf{E} — одинична матриця. Вектор \mathbf{V}_{n+1} можна виразити через вектор початкових умов \mathbf{V}_0 :

$$\mathbf{V}_{n+1} = (\mathbf{E} + h\mathbf{A})^n \mathbf{V}_0. \quad (3.9)$$

Позначимо

$$\mathbf{B} = \mathbf{E} + h\mathbf{A} \quad (3.10)$$

і застосуємо перетворення подоби для матриці \mathbf{B}

$$\mathbf{B} = \mathbf{T}^{-1} \mathbf{diag} \{ \lambda_{B_j} \} \mathbf{T}.$$

Тут \mathbf{T} — перетворююча матриця; $\mathbf{diag} \{ \lambda_{B_j} \}$ — діагональна матриця із власними значеннями λ_{B_j} матриці \mathbf{B} на діагоналі. Неважко бачити, що

$$\mathbf{B}^n = \mathbf{T}^{-1} \mathbf{diag} \{ \lambda_{B_j}^n \} \mathbf{T}.$$

З лінійної алгебри відомо, що власні значення матриць, зв'язаних арифметичними операціями, виявляються зв'язаними такими ж перетвореннями. Тому з (3.10) треба

$$\lambda_{B_j} = 1 + h\lambda_{A_j}.$$

Точне рішення модельної задачі (3.8) $\mathbf{V}(t) \rightarrow 0$ при $t \rightarrow \infty$, отже, умовою стійкості процесу чисельного рішення можна вважати

$$\mathbf{V}_{n+1} \rightarrow 0 \text{ при } n \rightarrow \infty,$$

звідки послідовно одержуємо

$$(\mathbf{E} + h\mathbf{A})^n \mathbf{V}_0 \rightarrow 0,$$

у зв'язку з тим, що $\mathbf{V}_0 \neq 0$, та $(\mathbf{E} + h\mathbf{A})^n \rightarrow 0$, оскільки $\mathbf{T} \neq 0$, та $\lambda_{\mathbf{B}_j}^n \rightarrow 0$ й умова стійкості

$$|1 + h\lambda_{\mathbf{A}_j}| < 1. \quad (3.11)$$

Відомо, що для фізично стійких систем власні значення матриці коефіцієнтів у ММС виявляються зі знаком мінус. Якщо до того ж всі $\lambda_{\mathbf{A}_j}$ вещественні величини (характер процесів у ММС із моделлю (3.8) аперіодичний), то природно визначити постійні часу фізичної системи як

$$\tau_j = -\frac{1}{\lambda_{\mathbf{A}_j}},$$

і умова (4) конкретизується в такий спосіб

$$\left|1 - \frac{h}{\tau_j}\right| < 1$$

або

$$0 < h < 2\tau_{\min}, \quad (3.12)$$

де τ_{\min} — мінімальна постійна часу. Якщо використати явні методи більш високого порядку, то може збільшитися коефіцієнт перед τ_{\min} в (3.12), але це принципово не міняє оцінки явних методів.

Якщо порушено умову (3.12), то відбувається втрата стійкості обчислень, а це означає, що в рішенні задачі виникають помилкові коливання з амплітудою, що збільшується від кроку до кроку, і швидким аварійним останком ЕОМ внаслідок переповнення розрядної сітки. Звичайно, ні про яку адекватність рішення говорити не доводиться.

Для дотримання (3.12) застосовують ті або інші алгоритми автоматичного вибору кроку. Відзначимо, що в складній моделі розрахунок τ_{\min} для безпосереднього вибору кроку по (3.12) занадто трудомісткий, крім того, однократний розрахунок τ_{\min} мало ефективний, тому що в нелінійних моделях τ_{\min} може змінюватися від кроку до кроку.

Умова (3.12) накладає жорсткі обмеження на крок інтегрування. У результаті обчислювальна ефективність явних методів різко падає з погіршенням *обумовленості* ММС. Справді, тривалість $T_{\text{инт}}$ моделюємого процесу повинна бути порівнянною із часом заспокоєння системи після збудливого впливу, тобто порівнянна з максимальною постійною часу τ_{\max} . Необхідне число кроків інтегрування дорівнює

$$\text{III} = \frac{T_{\text{инт}}}{h} \sim \frac{\tau_{\max}}{\tau_{\min}}.$$

Відношення $\text{Ч} = \frac{\tau_{\max}}{\tau_{\min}}$ називають розкидом постійних часу або *числом обумовленості*.

Більше строго число обумовленості матриці \mathbf{A} визначається як $\|\mathbf{A}\| \times \|\mathbf{A}^{-1}\|$, а число обумовленості системи рівнянь $\mathbf{AX}=\mathbf{B}$ як $\|\mathbf{A}^{-1}\| \frac{\|\mathbf{B}\|}{\|\mathbf{X}\|}$. Чим більше це число, тим гірше обумовленість. Спроби застосування явних методів до різних ММС найчастіше приводять до неприпустимо низької обчислювальної ефективності, оскільки в реальних моделях $\text{Ч} > 10^5$ — звичайна ситуація. Тому в цей час в універсальних програмах аналізу явні методи рішення СЗДР не застосовують.

Аналогічний аналіз числової стійкості неявних методів дає наступні результати. Замість (3.9) маємо

$$\mathbf{V}_T = (\mathbf{E} - h\mathbf{A})^{-n} \mathbf{V}_0$$

і умова числової стійкості приймає вид:

$$\left| \frac{1}{1 + \frac{h}{\tau_j}} \right| < 1$$

яке виконується при будь-яких $h > 0$. Отже, неявний метод Ейлера має так названу А-стійкість.

Метод інтегрування СЗДР називають А-стійким, якщо погрішність інтегрування залишається обмеженою при будь-якому кроці $h > 0$.

Застосування *А-стійких методів* дозволяє істотно зменшити необхідні числа кроків **III**. У цих методах крок вибирається автоматично не з умов стійкості, а тільки з міркувань точності рішення.

Вибір порядку методу рішення СЗДР досить простий: по-перше, більше високий порядок забезпечує більше високу точність, по-друге, серед неявних різницевих методів, крім методу Ейлера, А-стійкі також методи другого порядку й серед них – метод трапецій. Тому переважне поширення в програмах аналізу одержали методи другого порядку – модифікації методу трапецій.

Лекція 9. Алгоритми чисельного інтегрування систем диференціальних рівнянь

Одна із вдалив реалізацій неявного методу другого порядку, яку можна вважати модифікацією *методу трапецій*, заснована на комбінованому використанні явної й неявної формул Ейлера. Розглянемо питання, чому таке комбінування знижує погрішність і приводить до підвищення порядку методу.

Попередньо відзначимо, що в методах P -го порядку локальна погрішність, тобто погрішність, допущена на одному n -м кроці інтегрування, оцінюється старшим із членів, що відкидають у розкладанні $\mathbf{V}(t)$ рішення в ряд Тейлора

$$\delta = c \left| \mathbf{V}^{(p+1)}(\tau) \right| h^{p+1},$$

де c – постійний коефіцієнт, що залежить від методу, $\left| \mathbf{V}^{(p+1)}(\tau) \right|$ норма $(p+1)$ -х похідних вектора $\mathbf{V}(t)$, що оцінюється за допомогою кінцево-різничної апроксимації, τ – значення часу t усередині кроку.

Якщо n -й крок інтегрування в *комбінованому методі* був неявним, тобто виконаним по неявній формулі, то наступний крок з тим же значенням h повинен бути явним. Використовуючи розкладання рішення $\mathbf{V}(t)$ в ряд Тейлора рядом з точкою t_{n+1} , одержуємо для $(n+1)$ -го неявного кроку

$$\mathbf{V}(t_n) = \mathbf{V}(t_{n+1}) - \frac{d\mathbf{V}}{dt} h_n + \frac{d^2\mathbf{V}}{dt^2} \frac{h_n^2}{2!} - \frac{d^3\mathbf{V}}{dt^3} \frac{h_n^3}{3!} + \dots, \quad (3.13)$$

і для $(n+2)$ -го явного кроку

$$\mathbf{V}(t_{n+2}) = \mathbf{V}(t_{n+1}) + \frac{d\mathbf{V}}{dt} h_y + \frac{d^2\mathbf{V}}{dt^2} \frac{h_y^2}{2!} + \frac{d^3\mathbf{V}}{dt^3} \frac{h_y^3}{3!} + \dots, \quad (3.14)$$

де h_n й h_y — величини неявного і явного кроків, а значення похідних ставляться до моменту

t_{n+1} . Підставляючи (3.12) в (3.14), при $h = h_n = h_x$ одержуємо:

$$\mathbf{V}(t_{n+2}) = \mathbf{V}(t_n) + 2 \frac{d\mathbf{V}}{dt} h + 2 \frac{d^3 \mathbf{V}}{dt^3} \frac{h^3}{3!} + \dots, \quad (3.15)$$

тобто погрішності, що обумовлюють квадратичними членами в (3.13) і (3.14) взаємно компенсуються, і старшим зі членів, що відкидають, стає член з h^3 . Отже, викладене комбінування неявної і явної формул Ейлера дає метод інтегрування другого порядку.

Неявні методи й, зокрема, розглянутий комбінований метод доцільно використати тільки при змінній величині кроку. Дійсно, при помітних швидкостях зміни фазових змінних погрішність залишається в припустимих межах тільки при малих кроках, у квазистатичних режимах крок може бути в багато разів більше.

Алгоритми *автоматичного вибору кроку* засновані на порівнянні допущених і припустимої локальних погрішностей. Наприклад, уводиться деякий діапазон (коридор) погрішностей δ , у межах якого крок зберігається незмінним. Якщо ж допущена погрішність перевищує верхню границю діапазону, то крок зменшується, якщо ж виходить за нижню границю, то крок збільшується.

9.1. Методи рішення систем лінійних алгебраїчних рівнянь

У програмах аналізу в САПР для рішення СЛАР найчастіше застосовують метод Гауса або його різновиду. *Метод Гауса* — метод послідовного виключення невідомих із системи рівнянь. При виключенні k -ї невідомої x_k із системи рівнянь

$$\mathbf{A}\mathbf{X} = \mathbf{B} \quad (3.16)$$

всі коефіцієнти a_{ij} при $i > k$ й $j > k$ перераховують по формулі

$$a_{ij} := a_{ij} - \frac{a_{ik} \cdot a_{kj}}{a_{kk}}. \quad (3.17)$$

Виключення $n-1$ невідомих, де n — порядок системи (3.16), називають прямим ходом, у процесі якого матриця коефіцієнтів здобуває трикутний вид. При зворотному ході послідовно обчислюють невідомі, починаючи з x_n .

У загальному випадку число арифметичних операцій для рішення (3.16) по Гаусу пропорційно n^3 . Це приводить до значних витрат машинного часу, оскільки СЛАР вирішується багаторазово в процесі одноваріантного аналізу, і істотно обмежує складність аналізованих об'єктів.

Можна помітно підвищити обчислювальну ефективність аналізу, якщо використати характерну практично для всіх застосувань властивість високої *розрідженості матриці* \mathbf{A} в моделі (3.16).

Матрицю називають розрідженою, якщо більшість її елементів дорівнює нулю. Ефективність обробки розріджених матриць велика тому, що, по-перше, перерахування по формулі (3.17) не потрібно, якщо хоча б один з елементів a_{ik} або a_{kj} виявляються нульовим, по-друге, не потрібні витрати пам'яті для зберігання нульових елементів. Хоча алгоритми обробки розріджених матриць більш складні, але в результаті вдається одержати витрати машинного часу, близькі до лінійних, наприклад, витрати виявляються пропорційними $n^{1,2}$.

При використанні *методів розріджених матриць* потрібно враховувати залежність

обчислювальної ефективності від того, як представлена матриця коефіцієнтів A , точніше від того, у якому порядку записані її рядки й стовпці.

Для пояснення цієї залежності розглянемо два варіанти подання однієї й тієї ж СЛАР. У першому випадку система рівнянь має вигляд

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 = b_1;$$

$$a_{21}x_1 + a_{22}x_2 = b_2;$$

$$a_{31}x_1 + a_{33}x_3 = b_3;$$

$$a_{41}x_1 + a_{44}x_4 = b_4;$$

$$a_{51}x_1 + a_{55}x_5 = b_5.$$

При прямому ході відповідно до формули (3.17) всі елементи матриці, які спочатку були нульовими, стають ненульовими, а матриця виявляється повністю насиченою. Елементи, що стають ненульовими в процесі гаусових виключень, називають вторинними ненулями. Вторинні нулі в Табл. 1 відзначені знаком «*».

У другому випадку міняються місцями перше й п'яте рівняння. Матриці коефіцієнтів мають вигляд Табл. 1 і Табл. 2, де ненульові елементи представлені знаком «+». Тепер вторинні ненульові елементи не з'являються, матриця залишається розрідженою, висока обчислювальна ефективність зберігається.

Таким чином, методи розріджених матриць повинні містити в собі способи оптимального впорядкування рядків і стовпців матриць. Використовують кілька критеріїв оптимальності впорядкування. Найпростішим з них є критерій розташування рядків у порядку збільшення числа первинних нулів, більш складні критерії враховують не тільки первинні нулі, але й вторинні нулі, що з'являються.

Таблиця 1

+	+	+	+	+
+	+	*	*	*
+	*	+	*	*
+	*	*	+	*
+	*	*	*	+

Таблиця 2

+	.	.	.	+
.	+	.	.	+

.	.	+	.	+
.	.	.	+	+
+	+	+	+	+

Методом розріджених матриць називають метод рішення СЛАР на основі методу Гауса з урахуванням розрідженості (первинної й вторинної) матриці коефіцієнтів.

Метод розріджених матриць можна реалізувати шляхом інтерпретації й компіляції. В обох випадках створюються масиви ненульових коефіцієнтів матриці (з урахуванням вторинних нулів) і масиви координат цих ненульових елементів.

При цьому виграш у витратах пам'яті досить значний. Так, при матриці помірною розміру 200x200 без урахування розрідженості буде потрібно 320 кбайт. Якщо ж взяти характерне значення 9 для середнього числа нулів в одному рядку, то для коефіцієнтів і покажчиків координат буде потрібно не більше 28 кбайт.

У випадку інтерпретації моделююча програма для кожної операції по (3.17) при $a_{ik} \neq 0$ й $a_{kj} \neq 0$ знаходить, використовуючи покажчики, потрібні коефіцієнти й виконує арифметичні операції по (3.17). Оскільки СЛАР в процесі аналізу вирішується багаторазово, те й операції пошуку потрібних коефіцієнтів також повторюються багаторазово, на що природно витрачається машинний час.

Спосіб компіляції більше економічний по витратах часу, але уступає способу інтерпретації по витратах пам'яті. При компіляції пошук потрібних для (3.17) коефіцієнтів виконується однократно перед чисельним рішенням задачі. Замість безпосереднього виконання арифметичних операцій для кожної з них компілюється команда зі знайденими адресами ненульових коефіцієнтів. Такі команди утворюють робочу програму рішення СЛАР, що і буде вирішуватися багаторазово. Очевидно, що тепер у робочій програмі буде виконуватися мінімально необхідне число арифметичних операцій.

До числа різновидів методу Гауса відноситься метод *LU-розкладання*, що заснований на розкладанні матриці коефіцієнтів \mathbf{A} системи (3.16) на верхню \mathbf{U} й нижню \mathbf{L} трикутні матриці, що еквівалентно прямому ходу в методі Гауса. Елементи матриць \mathbf{U} і \mathbf{L} обчислюються по рекурентним формулам

$$l_{ij} = a_{ij} - \sum_{q=1}^{j-1} l_{iq} u_{qj} \text{ при } i \geq j,$$

$$u_{ij} = \frac{a_{ij} - \sum_{q=1}^{j-1} l_{iq} u_{qj}}{l_{ii}} \text{ при } j > i,$$

причому $u_{ii} = 1$, $l_{i1} = a_{i1}$, $u_{1i} = a_{1i}$. Стовпці матриці \mathbf{L} обчислюються в порядку збільшення j , починаючи с $j=2$. Після обчислення k -го стовпця \mathbf{L} безпосередньо виконується розрахунок k -го рядка \mathbf{U} . Зворотний хід полягає в розрахунку вектора \mathbf{Z} з матричного рівняння $\mathbf{LZ} = \mathbf{B}$ й, нарешті, вектора \mathbf{X} з рівняння $\mathbf{UX} = \mathbf{Z}$.

Крім методу Гауса, для рішення систем лінійних алгебраїчних рівнянь (3.17) застосовують ітераційні методи: метод простої ітерації, метод Зейделя, метод Якобі, методи послідовних верхньої й нижньої релаксації. Наприклад, у *методі простої ітерації* обчислення виконують по формулі

$$\mathbf{X} := \mathbf{X} - h(\mathbf{A}\mathbf{X} - \mathbf{B}),$$

причому для забезпечення збіжності параметр h потрібно вибирати з умови $|1+h\lambda_j| < 1$ для будь-якого j , де λ_j — j -е власне значення матриці \mathbf{A} .

9.2. Методи рішення систем нелінійних алгебраїчних рівнянь

Обчислення при рішенні СНАР складаються з декількох вкладених один в іншій циклічних процесів. Зовнішній цикл – цикл покорокового чисельного інтегрування, параметром циклу є номер кроку. Якщо модель аналізованого об'єкта нелінійна, то на кожному кроці виконується проміжний цикл – ітераційний цикл рішення системи нелінійних алгебраїчних рівнянь (СНАР). Параметр циклу – номер ітерації. У внутрішньому циклі вирішується система лінійних алгебраїчних рівнянь (СЛАР), наприклад, при застосуванні вузлового методу формування ММС такою системою є

$$\mathbf{Y}_n \boldsymbol{\varphi}_n = \mathbf{V}_n, \quad (3.18)$$

де \mathbf{Y}_n — матриця Якобі, \mathbf{V}_n — вектор правих частин. Тому в математичне забезпечення аналізу на макрорівні входять методи рішення СНАР й СЛАР.

Для рішення систем алгебраїчних рівнянь можна застосовувати прямі ітераційні методи. До них відносяться методи простої ітерації, Зейделя, Якобі, релаксації. Для них необхідне виконання жорстких умов збіжності, характерна порівняно повільна збіжність.

Тому в сучасних програмах аналізу найбільше поширення одержав *метод Ньютона*, заснований на лінеаризації СНАР. Сама модель (3.18) отримана саме відповідно до методу Ньютона. Основна перевага методу Ньютона – висока швидкість збіжності.

Представимо СНАР у вигляді

$$\mathbf{F}(\mathbf{X}) = 0. \quad (3.19)$$

Розкладаючи $\mathbf{F}(\mathbf{X})$ в ряд Тейлора біля деякої точки \mathbf{X}_k , одержуємо

$$\mathbf{F}(\mathbf{X}) = \mathbf{F}(\mathbf{X}_k) + \frac{\partial \mathbf{F}}{\partial \mathbf{X}} (\mathbf{X} - \mathbf{X}_k) + (\mathbf{X} - \mathbf{X}_k)^T \frac{\partial^2 \mathbf{F}}{\partial \mathbf{X}^2} \frac{\mathbf{X} - \mathbf{X}_k}{2} + \dots = 0.$$

Зберігаючи тільки лінійні члени, одержуємо СЛАР з невідомим вектором \mathbf{X} :

$$\mathbf{Y}_k (\mathbf{X} - \mathbf{X}_k) = -\mathbf{F}(\mathbf{X}_k), \quad (3.20)$$

де $\mathbf{Y}_k = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{X}} \right|_k$ — матриця Якобі. Рішення системи (3.20) дає чергове наближення до кореня системи (3.19), що зручно позначити \mathbf{X}_{k+1} .

Обчислювальний процес стартує з початкового наближення \mathbf{X}_0 у випадку збіжності ітерацій закінчується, коли погрішність, оцінювана як

$$|\Delta \mathbf{X}_k| = |\mathbf{X}_k - \mathbf{X}_{k-1}|,$$

стане менше припустимої погрішності ϵ .

Однак метод Ньютона не завжди приводить до збіжних ітерацій. Умови збіжності методу Ньютона виражаються досить складно, але існує легко використовуваний підхід до поліпшення збіжності. Це близькість початкового наближення до шуканого кореня СНАР. Використання цього фактору привело до появи методу рішення СНАР, названого продовженням рішення по параметрі.

У *методі продовження рішення по параметрі* в ММС виділяється деякий параметр α , такий, що при $\alpha = 0$ корінь $\mathbf{X}_{\alpha=0}$ системи (3.19) відомий, а при збільшенні α від 0 до його дійсного значення складові вектора \mathbf{X} плавно змінюються від $\mathbf{X}_{\alpha=0}$ до дійсного значення кореня. Тоді задача розбивається на ряд подзадач, послідовно розв'язуваних при послідовних

значеннях α , і при досить малому кроці $\Delta\alpha$ зміни α умови збіжності виконуються.

Як параметр α можна вибрати деякий зовнішній параметр, наприклад, при аналізі електронних схем їм може бути напруга джерела харчування. Але на практиці при інтегруванні СНАР в якості α вибирають крок інтегрування h . Очевидно, що при $h=0$ корінь СНАР дорівнює значенню вектора невідомих на попередньому кроці. Регулювання значень h покладають на алгоритм автоматичного вибору кроку.

В цих умовах очевидна доцільність подання математичних моделей для аналізу статичних станів у вигляді СНАР, як і для динамічного аналізу.

До інших методів рішення систем алгебраїчних рівнянь, використовуваним у математичному забезпеченні САПР, відносяться методи простої ітерації, Зейделя, Якобі, релаксації.

Відповідно до *методу простої ітерації* обчислення виконують по формулі

$$\mathbf{X} := \mathbf{X} - h\mathbf{F}(\mathbf{X}), \quad (3.21)$$

причому для забезпечення збіжності параметр h потрібно вибрати з умови $|1 + h\lambda_j| < 1$ для будь-якого j , де λ_j — j -е власне значення матриці Якобі.

Метод Зейделя відрізняється від методу простої ітерації тим, що права частина ітераційної формули (3.24) обновляється відразу ж після обчислення чергового елемента вектора \mathbf{X} .

Відповідно до *методу Якобі* обчислення виконують по формулі

$$\mathbf{X} := \mathbf{X} - h\mathbf{D}^{-1}\mathbf{F}(\mathbf{X}), \quad (3.22)$$

де $\mathbf{D} = \text{diag}\{\mathbf{Y}\}$ — діагональ матриці Якобі системи рівнянь (3.18).

9.3. Аналіз у частотній області

Аналіз у частотній області більш специфічний у порівнянні з аналізом у часовій області. Його застосовують, як правило, до об'єктів з лінеаризуємими ММ при дослідженні коливальних стаціонарних процесів, аналізі стійкості, розрахунку перекручувань інформації, що представляється спектральними складових сигналів, і т.п.

Аналіз у частотній області виконується відповідно лінеаризованим моделям об'єктів. Для лінійних СЛАР справедливе застосування для алгебраїзації диференціальних рівнянь перетворення Фур'є, у якому оператор d/dt замінюється на оператор $j\omega$.

Характерною особливістю є комплексний характер матриці коефіцієнтів, що до деякої міри ускладнює процедуру рішення, але не створює принципових труднощів. При рішенні задають ряд частот ω_k . Для кожної частоти вирішують СЛАР й визначають дійсні й мнимі частини шуканих фазових змінних. По них визначають амплітуду й фазовий кут кожної спектральної складової, що й дозволяє побудувати амплітудно-частотні характеристики, знайти власні частоти коливальної системи й т.п.

9.4. Багатоваріантний аналіз

Одноваріантний аналіз дозволяє одержати інформацію про стан і поведінку проектного об'єкта в одній точці простору внутрішніх \mathbf{X} і зовнішніх \mathbf{Q} параметрів. Очевидно, що для оцінки властивостей проектного об'єкта цього недостатньо. Потрібно виконувати багатоваріантний аналіз, тобто досліджувати поведінку об'єкта у ряді точок згаданого простору, що для стислості будемо далі називати простором аргументів.

Найчастіше *багатоваріантний аналіз* у САПР виконується в інтерактивному режимі,

коли розроблювач неодноразово міняє в математичній моделі ті або інші параметри із множин \mathbf{X} і \mathbf{Q} , виконує одноваріантний аналіз і фіксує отримані значення вихідних параметрів. Подібний багатоваріантний аналіз дозволяє оцінити *області працездатності*, ступінь виконання умов працездатності, а отже, ступінь виконання ТЗ на проектування, розумність прийнятих проміжних рішень по зміні проекту й т.п.

Областю працездатності називають область у просторі аргументів, у межах якої виконуються всі задані умови працездатності, тобто значення всіх вихідних параметрів перебувають у припустимі по ТЗ межах.

Серед процедур багатоваріантного аналізу можна виділити типові, виконувані по заздалегідь складених програмах. До таких процедур ставляться аналіз чутливості й статистичний аналіз.

Найбільш просто *аналіз чутливості* реалізується шляхом чисельного диференціювання. Нехай аналіз проводиться в деякій точці $\mathbf{X}_{\text{ном}}$ простору аргументів, у якій попередньо проведений одноваріантний аналіз і знайдені значення вихідних параметрів $y_{j\text{ном}}$. Виділяється N параметрів-аргументів x_i (із числа елементів векторів \mathbf{X} і \mathbf{Q}), вплив яких на вихідні параметри підлягає оцінити, по черзі кожний з них одержує приріст Δx_i , виконується одноваріантний аналіз, фіксуються значення вихідних параметрів y_j і підраховуються значення абсолютних

$$A_{j\bar{i}} = \frac{y_j - y_{j\text{ном}}}{\Delta x_i}$$

і відносних

$$B_{j\bar{i}} = A_{j\bar{i}} \frac{x_{j\text{ном}}}{y_{j\text{ном}}}$$

коефіцієнтів чутливості.

Такий метод чисельного диференціювання називають *методом приростів*. Для аналізу чутливості, відповідно до методу приростів, потрібно виконати $N+1$ раз одноваріантний аналіз. Результат його застосування — матриці абсолютної й відносної чутливості, елементами яких є коефіцієнти $A_{j\bar{i}}$ й $B_{j\bar{i}}$.

Аналіз чутливості – це розрахунок векторів градієнтів вихідних параметрів, що входить складовою частиною в програми параметричної оптимізації, що використовують градієнтні методи.

Ціль *статистичного аналізу* — оцінка законів розподілу вихідних параметрів і (або) числових характеристик цих розподілів. Випадковий характер величин y_j обумовлений випадковим характером параметрів елементів x_i , тому вихідними даними для статистичного аналізу є відомості про закони розподілу x_i . Відповідно до результатів статистичного аналізу прогнозують такий важливий виробничий показник, як відсоток бракованих виробів у готовій продукції (Рис. 2). На малюнку представлена розрахована щільність P розподілу вихідного параметра y_j , що має умову працездатності $y_j < T_j$, затемнена ділянка характеризує частку виробів, що не задовольняють умові працездатності параметра y_j .

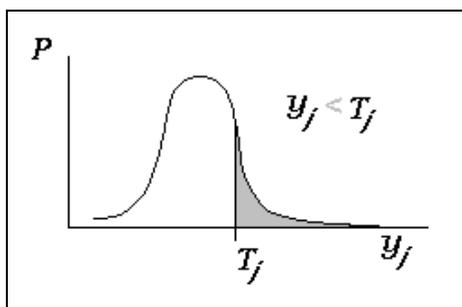


Рис. 2. Результат статистичного аналізу.

У САПР статистичний аналіз здійснюється чисельним методом — *методом Монте-Карло* (статистичних випробувань). Відповідно до цього методу виконуються N статистичних випробувань, кожне статистичне випробування являє собою одноваріантний аналіз, виконуваний при випадкових значеннях параметрів-аргументів. Ці випадкові значення вибирають відповідно до заданих законів розподілу аргументів x_i . Отримані в кожному випробуванні значення вихідних параметрів накопичують, після N випробувань обробляють, що дає наступні результати:

- гістограми вихідних параметрів;
- оцінки математичних очікувань і дисперсій вихідних параметрів;
- оцінки коефіцієнтів кореляції й регресії між обраними вихідними й внутрішніми параметрами, які, зокрема, можна використати для оцінки коефіцієнтів чутливості.

Статистичний аналіз, виконуваний відповідно до методу Монте-Карло, — трудомістка процедура, оскільки число випробувань N доводиться вибирати досить великим, щоб досягти прийнятної точності аналізу. Інша причина, що ускладнює застосування методу Монте-Карло, — труднощі в одержанні достовірної вихідної інформації про закони розподілу аргументів-параметрів x_i .

Більш типова ситуація, коли закони розподіли x_i невідомі, але з великою долею впевненості можна вказати гранично припустимі відхилення Δx_i параметрів x_i від номінальних значення $x_{i\text{ном}}$ (такі відхилення часто вказуються в паспортних даних на комплектуючі деталі). У таких випадках більш реалістично застосовувати метод *аналізу на найгірший випадок*. Відповідно до цього методу, спочатку виконують аналіз чутливості з метою визначення знаків коефіцієнтів чутливості. Далі здійснюють m раз одноваріантний аналіз, де m — число вихідних параметрів. У кожному варіанті задають значення аргументів, найбільш несприятливі для виконання умов працездатності чергового вихідного параметра $y_j, j \in [1:m]$. Так, якщо $y_j < T_j$ й коефіцієнт чутливості позитивний (тобто $\text{sign}(A_{ji}) = 0$) або $y_j > T_j$ й $\text{sign}(A_{ji}) = 1$, то

$$x_i = x_{i\text{ном}} + \Delta x_i,$$

інакше

$$x_i = x_{i\text{ном}} - \Delta x_i,$$

Однак варто помітити, що, проводячи аналіз на найгірший випадок, можна одержати завищені значення розкиду вихідних параметрів, і якщо домагатися виконання умов працездатності в найгірших випадках, те це часто веде до невиправданого збільшення

вартості, габаритних розмірів, маси й інших показників проєктованих конструкцій, хоча й гарантує із запасом виконання умов працездатності.

9.5. Математичні моделі для аналізу на мікрорівні

Математичними моделями на мікрорівні є диференціальні рівняння в частинних похідних або інтегральні рівняння, що описують поля фізичних величин. Інакше кажучи, на мікрорівні використовуються *моделі з розподіленими параметрами*. У якості незалежних змінних у моделях можуть фігурувати просторові змінні x_1, x_2, x_3 й час t .

Характерними прикладами моделей можуть служити рівняння математичної фізики разом із заданими крайовими умовами.

Приклад 1

Рівняння теплопровідності:

$$C\rho \frac{\partial T}{\partial t} = \operatorname{div} (\lambda \mathbf{grad} T) + g,$$

де C — питома теплоємність, ρ — густина, T — температура, t — час, λ — коефіцієнт теплопровідності, g — кількість теплоти, виділюваної в одиницю часу в одиниці об'єму.

Приклад 2

Рівняння дифузії:

$$\frac{\partial N}{\partial t} = \operatorname{div} (D \mathbf{grad} N),$$

де N — концентрація часток, D — коефіцієнт дифузії.

Крайові умови включають початкові умови, що характеризують просторовий розподіл залежних змінних у початковий момент часу, і граничні, що задають значення цих змінних на границях розглянутої області у функції часу.

Лекція 10. Методи аналізу на мікрорівні

Аналіз сіткових методів

У САПР рішення диференціальних або інтегро-диференціальних рівнянь із частковими похідними виконується чисельними методами. Ці методи засновані на дискретизації незалежних змінних — їхньому представленні кінцевою множиною значень в обраних вузлових крапках досліджуваного простору. Ці точки розглядаються як вузли деякої сітки, тому використовувані в САПР методи — це сіткові методи.

Серед сіткових методів найбільше поширення одержали два методи: метод кінцевих різностей (МКР) і метод скінченних елементів (МСЕ). Звичайно виконують дискретизацію просторових незалежних змінних, тобто використовують просторову сітку. У цьому випадку результатом дискретизації є система звичайних диференціальних рівнянь для нестационарної задачі або система алгебраїчних рівнянь для стаціонарної.

Нехай необхідно вирішити рівняння

$$LV(\mathbf{z}) = f(\mathbf{z})$$

із заданими крайовими умовами

$$MV(\mathbf{z}) = \psi(\mathbf{z}),$$

де L й M — диференціальні оператори, $V(\mathbf{z})$ — фазова змінна, $\mathbf{z} = (x_1, x_2, x_3, t)$ — вектор

незалежних змінних, $f(\mathbf{z})$ і $\Psi(\mathbf{z})$ — задані функції незалежних змінних.

У методі кінцевих різностей алгебраїзація похідних по просторових координатах базується на апроксимації похідних кінцево-різницевиими виразами. При використанні методу потрібно вибрати кроки сітки по кожній координаті й вид шаблону. Під шаблоном розуміють множину вузлових точок, значення змінних у яких використовуються для апроксимації похідної в одній конкретній точці.

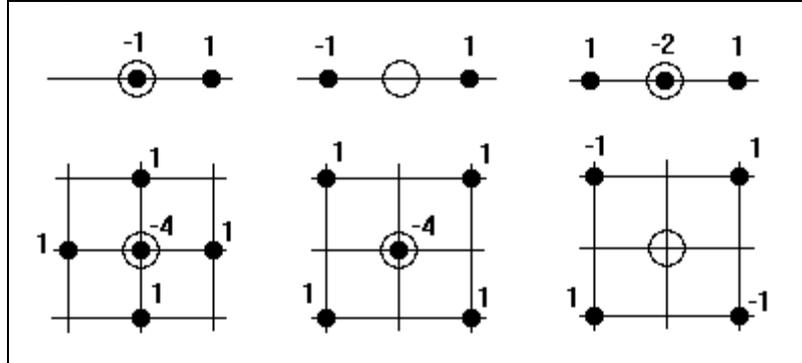


Рис. 3. Приклади шаблонів для одновимірних і двовимірних задач

Приклади шаблонів для одновимірних і двовимірних задач наведені на Рис. 3. На цьому рисунку кружком більшого діаметра позначені вузли, у яких апроксимується похідна. Чорними точками позначені вузли, значення фазової змінної в яких входять в апроксимуючий вираз. Число, записане біля вузла, дорівнює коефіцієнту, з яким значення фазової змінної входить в апроксимуюче вираження. Так, для одновимірних шаблонів у верхній частині малюнка

показана апроксимація похідної $\frac{\partial V}{\partial x}$ в точці k , і зазначеним шаблонам при їхньому перегляді ліворуч-праворуч відповідають апроксимації

$$h \frac{\partial V}{\partial x} = V_{k+1} - V_k;$$

$$2h \frac{\partial V}{\partial x} = V_{k+1} - V_{k-1};$$

$$h^2 \frac{\partial^2 V}{\partial x^2} = V_{k+1} - 2V_k + V_{k-1},$$

де h — крок дискретизації по осі x .

Шаблони для двовимірних задач у нижній частині Рис.3 відповідають наступним кінцево-різничним операторам:

- лівий рисунок:

$$h^2 \nabla^2 V = C^2 \left(\frac{\partial^2 V}{\partial x_1^2} + \frac{\partial^2 V}{\partial x_2^2} \right) = V_{k+1j} + V_{k-1j} + V_{kj+1} + V_{kj-1} - 4V_{kj};$$

- середній рисунок:

$$2h^2 \nabla^2 V = V_{k+1j+1} + V_{k-1j+1} + V_{k+1j-1} + V_{k-1j-1} - 4V_{kj};$$

- правий рисунок:

$$4h^2 \frac{\partial^2 V}{\partial x_1 \partial x_2} = V_{k+1j+1} - V_{k-1j+1} - V_{k+1j-1} + V_{k-1j-1};$$

Тут V_{kj} — значення V в точці (x_{1k}, x_{2j}) ; прийняті однакові значення кроків h по обох координатах.

Метод скінченних елементів заснований на апроксимації не похідних, а самого рішення $V(\mathbf{z})$. Але оскільки воно невідомо, то апроксимація виконується вираженнями з невизначеними коефіцієнтами q_i

$$U(\mathbf{z}) = \mathbf{Q}^T \boldsymbol{\Phi}(\mathbf{z}), \quad (3.23)$$

де $\mathbf{Q}^T = (q_1, q_2, \dots, q_n)^T$ — вектор-рядок невизначених коефіцієнтів, $\boldsymbol{\Phi}(\mathbf{z})$ — вектор-стовпець координатних функцій (опорних функцій), заданих так, що задовольняються граничні умови.

При цьому мова йде про апроксимації рішення в межах скінченних елементів, а з урахуванням їхніх малих розмірів можна говорити про використання порівняно простих апроксимуючих виразів $U(\mathbf{z})$ (наприклад, $\boldsymbol{\Phi}(\mathbf{z})$ — поліноми низьких ступенів). У результаті підстановки $U(\mathbf{z})$ у вихідне диференціальне рівняння й виконання операцій диференціювання одержуємо систему нев'язок

$$\Delta(\mathbf{z}, \mathbf{Q}) = LU(\mathbf{z}) - f(\mathbf{z}) = L(\mathbf{Q}^T \boldsymbol{\Phi}(\mathbf{z})) - f(\mathbf{z}), \quad (3.24)$$

з якої потрібно знайти вектор \mathbf{Q} .

Цю задачу (визначення \mathbf{Q}) вирішують одним з наступних методів:

- метод коллокацій, у якому, використовуючи (3.24), формують N рівнянь із невідомим вектором \mathbf{Q} :

$$L(\mathbf{Q}^T \boldsymbol{\Phi}(\mathbf{z}_i)) - f(\mathbf{z}_i) = 0, \quad i = 1, 2, \dots, N,$$

де N — число невизначених коефіцієнтів;

- метод найменших квадратів, заснований на мінімізації квадратів нев'язок у N точках або в середньому по розглянутій області;

- метод Гальоркіна, за допомогою якого мінімізуються в середньому по області нев'язка зі спеціальними ваговими коефіцієнтами, що задаються.

Найбільше поширення МСЕ одержав у САПР машинобудування для аналізу міцності об'єктів. Для цієї задачі можна використувати розглянутий підхід, тобто виконати алгебраїзацію вихідного рівняння пружності (рівняння Ламе). Однак більш зручним у реалізації МСЕ виявився підхід, заснований на варіаційних принципах механіки.

10.1. Метод скінченних елементів для аналізу механічної міцності

Як вихідне положення в методі скінченних елементів приймають варіаційний принцип Лагранжа (принцип потенційної енергії), відповідно до якого рівноважний стан, у який може прийти система, характеризується мінімумом потенційної енергії.

Потенційна енергія Π визначається як різниця енергії \mathcal{E} деформації тіла й роботи A масових і прикладених поверхневих сил.

У свою чергу, у сталому стані

$$\Theta = 0,5 \int_{\mathbf{R}} \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} d\mathbf{X}, \quad (3.25)$$

де $\boldsymbol{\varepsilon}^T = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23})^T$ — вектор-рядок відносних деформацій, $\boldsymbol{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})$ — вектор-стовпець напруг, $\mathbf{X} = (x_1, x_2, x_3)$, \mathbf{R} — розглянута область у просторі \mathbf{X} . Кожний елемент σ_{ij} вектора $\boldsymbol{\sigma}$ характеризує напругу, спрямовану уздовж осі x_j й прикладену до площадці, перпендикулярній x_i осі. Аналогічний зміст мають індекси усіх елементів вектора $\boldsymbol{\varepsilon}$.

Деформації ε_{ij} можна виразити через переміщення за допомогою рівнянь Коші

$$\varepsilon_{ij} = 0,5 \left(\frac{\partial W_i}{\partial x_j} + \frac{\partial W_j}{\partial x_i} \right), \quad (3.26)$$

де W_i — переміщення уздовж осі x_i , або в матричній формі

$$\boldsymbol{\varepsilon} = 0,5 \mathbf{S} \mathbf{W}, \quad (3)$$

де \mathbf{S} — очевидний з (3.26) оператор диференціювання.

Деформації й напруги зв'язані між собою за допомогою матриці \mathbf{D} , що характеризує пружні властивості середовища, і представлена в Табл. 3:

$$\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon}. \quad (4)$$

Фігуруючі у Табл. 3 коефіцієнт λ і модуль зсуву G називають постійними Ламе. Ці коефіцієнти пов'язані з модулем пружності E й коефіцієнтом Пуассона μ співвідношеннями

$$\lambda(1 + \mu)(1 - 2\mu) = \mu E \quad \text{й} \quad G = \frac{0,5 E}{1 + \mu};$$

$$k = \frac{1 - \mu}{1 - 2\mu^2 - \mu}.$$

Підставляючи (3.28) і (3.27) в (3.25), одержуємо

$$\Theta = 0,5 \int_{\mathbf{R}} (\mathbf{S} \mathbf{W})^T \mathbf{D} \mathbf{S} \mathbf{W} d\mathbf{R},$$

Таблиця 3

kE	λ	λ	0	0	0
λ	kE	λ	0	0	0
λ	λ	kE	0	0	0
0	0	0	G	0	0
0	0	0	0	G	0
0	0	0	0	0	G

Рішенням задачі повинне бути поле переміщень $\mathbf{W}(\mathbf{X})$. Відповідно до МСЕ це рішення апроксимується за допомогою координатних функцій і невизначених коефіцієнтів, які стосовно до сукупності скінченних елементів представимо в матричній формі:

$$\mathbf{U}(\mathbf{X}) = \mathbf{N} \mathbf{Q},$$

де \mathbf{N} — матриця координатних функцій, \mathbf{Q} — вектор невизначених коефіцієнтів.

Замінюючи $\mathbf{W}(\mathbf{X})$ на $\mathbf{U}(\mathbf{X})$, одержуємо

$$\mathfrak{E} = 0,5 \int_{\mathbf{R}} \mathbf{Q}^T (\mathbf{SN})^T \mathbf{D} \mathbf{S} \mathbf{N} \mathbf{Q} d\mathbf{X} = 0,5 \mathbf{Q}^T \left(\int_{\mathbf{R}} (\mathbf{SN})^T \mathbf{D} \mathbf{S} \mathbf{N} d\mathbf{X} \right) \mathbf{Q} = 0,5 \mathbf{Q}^T \mathbf{K} \mathbf{Q}, \quad (3.29)$$

$$\mathbf{K} = \int_{\mathbf{R}} (\mathbf{SN})^T \mathbf{D} \mathbf{S} \mathbf{N} d\mathbf{X},$$

де \mathbf{K} — матриця жорсткості.

Відповідно до принципу потенційної енергії в стані рівноваги маємо

$$\frac{\partial \Pi}{\partial \mathbf{Q}} = \frac{\partial \mathfrak{E}}{\partial \mathbf{Q}} - \frac{\partial \mathcal{A}}{\partial \mathbf{Q}} = 0$$

або, диференціюючи (3.29), знаходимо

$$\mathbf{K} \mathbf{Q} = \mathbf{B}, \quad (3.30)$$

де $\mathbf{B} = \frac{\partial \mathcal{A}}{\partial \mathbf{Q}}$ — вектор навантажень. Таким чином, задача аналізу міцності, згідно МСЕ, зведена до рішення системи лінійних алгебраїчних рівнянь (3.30).

Матриця жорсткості виявляється сильно розрідженою, тому для рішення (3.30) застосовують методи розріджених матриць.

Одним із широко відомих методів розріджених матриць є *метод прогону*, застосовуваний у випадку тридіагональних матриць коефіцієнтів у системі алгебраїчних рівнянь.

Для рішення розподілених задач часто застосовують метод Гальоркіна в рамках МСЕ. У методі Гальоркіна у випадку аналізу механічної міцності виробів замість вираження потенційної енергії використовують нев'язки, що виходять після заміни функції переміщення

$\mathbf{W}(\mathbf{X})$ апроксимуючим вираженням $\mathbf{N}(\mathbf{X})\mathbf{Q}$ у вихідному диференціальному рівнянні Ламе (для статичного режиму)

$$(\lambda + \mathcal{G}) \mathbf{grad} \operatorname{div} \mathbf{W}(\mathbf{X}) + \mathcal{G} \Delta \mathbf{W}(\mathbf{X}) + \mathbf{g}(\mathbf{X}) = 0,$$

$$\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2} \right)$$

Тут Δ — диференціальний оператор, що застосовується до всіх

елементів вектора $\mathbf{W}(\mathbf{X})$, $\mathbf{g}(\mathbf{X})$ — вектор масових і прикладених сил, віднесених до скінченного елемента. Нев'язка по області \mathbf{R} скінченного елемента при заміні $\mathbf{W}(\mathbf{X})$ на $\mathbf{N}(\mathbf{X})\mathbf{Q}$

$$\delta(\mathbf{Q}) = \int_{\mathbf{R}} ((\lambda + \mathcal{G}) \mathbf{grad} \operatorname{div} \mathbf{N}(\mathbf{X})\mathbf{Q} + \mathcal{G} \Delta \mathbf{N}(\mathbf{X})\mathbf{Q} + \mathbf{g}(\mathbf{X})) d\mathbf{X}$$

повинна бути мінімізована. Використовуючи вагові функції, що дорівнюють апроксимуючим вираженням $\mathbf{N}(\mathbf{X})\mathbf{Q}$, і необхідну умову екстремуму

$$\frac{\partial \delta}{\partial \mathbf{Q}} = 0$$

одержуємо систему алгебраїчних рівнянь (3.30).

Для практичного застосування МСЕ необхідно попередньо розробити математичні моделі скінчених елементів (СЕ) і реалізувати їх у бібліотеці СЕ програми аналізу механічної міцності.

Основою математичної моделі k -го СЕ є квадратна матриця жорсткості \mathbf{K}_k скінченного

елемента, розмір якої дорівнює числу Q_k невизначених коефіцієнтів у частині вектора Q , що ставиться до цього СЕ. У свою чергу, розмір цієї частини вектора Q є добуток розмірності d простору й числа b_k вузлів, виділених у моделі k -го СЕ. Це треба для того, щоб кожен невизначений коефіцієнт в Q у задачі аналізу механічної міцності був значенням переміщення W_{ij} в напрямку осі x_i в J -м вузлі. При одержанні моделі СЕ використовується матриця координатних функцій N_k скінченного елемента, у якій число рядків дорівнює d , а число стовпців $d \times b_k$. Наприклад, розмір K_k для СЕ у формі паралелепіпеда з вісьма вузлами дорівнює 24×24 , тому що розміри матриць S , N_k і D є 6×3 , 3×24 і 6×6 . При розрахунку K_k спочатку вибирають вид залежності переміщень від просторових координат. Потім перетворюють ці залежності у вираження $U_k(X) = N_k(X)Q_k$ й за допомогою інтегрування по просторових координатах визначають матрицю жорсткості СЕ.

При наявності бібліотеки СЕ застосування МСЕ зводиться до наступних операцій:

1. Створення геометричної моделі досліджуваного середовища (наприклад, деталі) за допомогою програми геометричного моделювання або шляхом зображення вручну на екрані дисплея ескізу (креслення) виробу.
2. Вибір бібліотечної моделі СЕ, завдання зовнішніх навантажень і значень геометричних і фізичних параметрів, формулювання граничних умов. Наступні операції виконуються програмою моделювання.
3. Реалізація в моделі сітки скінченних елементів. Тим самим стають відомими координати вузлових точок у моделі.
4. Приведення наявних об'ємних сил і поверхневих навантажень до вузлових точок моделі.
5. Об'єднання моделей СЕ в загальну скінченноелементну модель деталі (3.30), у якій матриця твердості K має порядок, рівний $d \times b$, де b — загальне число вузлів. При об'єднанні елементи матриці K утворюються підсумовуванням тих елементів матриць жорсткості K_k окремих СЕ, які відносяться до того самого вузла й напрямку переміщення. Якщо деякий вузол закріплений (його переміщення дорівнює нулю), то відповідному цьому вузлу рядки в K та Q і стовпці у K викреслюються.
6. Зміст інших операцій відповідає блокам програми, розглянутої стосовно до аналізу на макрорівні. Оскільки аналіз механічної міцності найчастіше проводиться в стаціонарних режимах у межах пружних деформацій, то наступною операцією є рішення системи (3.30).
7. Подання результатів рішення в зручній для користувача формі. Поряд із числовим висновком результатів звичайно використовується графічне зображення деформованої деталі, можливе подання розподілів напруг, деформацій, температур і т.п. усередині деталі із вказівкою їхньої інтенсивності за допомогою колірного розфарбування. До числа відомих програм аналізу по МСЕ відносяться ANSYS, NASTRAN, PATRAN і ін.

Лекція 11. Геометричне моделювання й машинна графіка

11.1. Типи геометричних моделей

Підсистеми машинної графіки й геометричного моделювання займають центральне місце в машинобудівних САПР. Конструювання виробів у них, як правило, проводиться в

інтерактивному режимі при оперуванні геометричними моделями, тобто математичними об'єктами, що відображають форму деталей, склад їх вузлів і, можливо, деякі додаткові параметри (маса, момент інерції, кольори поверхні й т.п.).

У таких підсистемах типовий маршрут обробки даних містить у собі одержання проектного рішення в прикладній програмі, його подання у вигляді геометричної моделі (*геометричне моделювання*), підготовку проектного рішення до візуалізації, візуалізацію в робочій станції й при необхідності коректування рішення в інтерактивному режимі. Дві останні операції реалізуються на базі апаратних засобів *машинної графіки*. Коли говорять про математичне забезпечення МГіГМ, мають на увазі насамперед моделі, методи й алгоритми для геометричного моделювання й підготовки до візуалізації. При цьому часто саме математичне забезпечення підготовки до візуалізації називають математичним забезпеченням машинної графіки.

Розрізняють математичне забезпечення двовимірного (2D) і тривимірного (3D) моделювання. Основні застосування 2D-графіки – підготовка креслярської документації в машинобудівних САПР, топологічне проектування друкованих плат і кристалів ВІС у САПР електронної промисловості. У розвинених машинобудівних САПР використовують як 2D, так і 3D моделювання для синтезу конструкцій, подання траєкторій робочих органів верстатів при обробці заготівель, генерації сітки скінченних елементів при аналізі міцності й т.п.

У процесі 3D моделювання створюються *геометричні моделі*, тобто моделі, що відбивають геометричні властивості виробів. Розрізняють геометричні моделі каркасні (дротові), поверхневі, об'ємні (твердотільні).

Каркасна модель представляє форму деталі у вигляді кінцевої множини ліній, що лежать на поверхнях деталі. Для кожної лінії відомі координати кінцевих точок і зазначена їх інцидентність ребрам або поверхням. Оперувати каркасною моделлю на подальших операціях маршрутів проектування незручно, і тому каркасні моделі в цей час використовують рідко.

Поверхнева модель відображає форму деталі за допомогою завдання обмежуючих її поверхонь, наприклад, у вигляді сукупності даних про грані, ребра й вершини.

Особливе місце займають моделі деталей з поверхнями складної форми, що мають назву *скульптурних поверхонь*. До таких деталей відносяться корпуси багатьох транспортних засобів (наприклад, судів, автомобілів), деталі, що обтікаються потоками рідин і газів (лопатки турбін, крила літаків), і ін.

Об'ємні моделі відрізняються тим, що в них у явній формі втримуються відомості про приналежність елементів внутрішньому або зовнішньому стосовно деталі простору.

Вказані моделі відображають тіла із замкнутими об'ємами, мають назву багатобразій (manifold). Деякі системи геометричного моделювання допускають оперування небагатобразними моделями (*nonmanifold*), прикладами яких можуть бути моделі тіл, що стикаються один з одним в одній точці або уздовж прямої. Небагатобразні моделі зручні в процесі конструювання, коли на проміжних етапах корисно працювати одночасно із тривимірними й двовимірними моделями, не задаючи товщини стінок конструкції, і т.п.

11.2. Методи й алгоритми комп'ютерної графіки

До методів комп'ютерної графіки відносять методи перетворення графічних об'єктів,

виведення (розгорнення) ліній у растровій формі, виділення вікна, видалення схованих ліній, проектування, зафарбування зображень.

Перетворення графічних об'єктів виконується за допомогою операцій переносу, масштабування, повороту.

Перенос точки з положення P в нове положення C можна виконувати по формулах типу

$$C_{xi} = P_{xi} + \Delta x_i,$$

де Δx_i — приріст по координаті x_i . Однак зручніше операції перетворення представляти в єдиній матричній формі

$$C = PT, \tag{4.1}$$

де T — перетворююча матриця. При цьому крапки C й P у двовимірному випадку зображують векторами-рядками 1×3 , у яких крім значень двох координат, що мають назву при такому поданні однорідними, додатково вказують масштабний множник W . Тоді перенос для випадку 2D можна виразити у вигляді (4.1), де T є Табл. 4, а $W = 1$.

Таблиця 4

1	0	0
0	1	0
Δx_1	Δx_2	1

Для операцій масштабування й повороту матриці T представлені в табл. 5 і табл. 6 відповідно, де m_x , m_y — масштабні множники, φ — кут повороту.

Зручність (4.1) пояснюється тим, що будь-яку комбінацію елементарних перетворень можна описати формулою (4.1). Наприклад, вираження для переміщення з одночасним поворотом має вигляд

$$C = PT_{\text{сд}}T_{\text{пов}} = PT,$$

де $T = T_{\text{сд}}T_{\text{пов}}$, $T_{\text{сд}}$ — матриця переміщення, $T_{\text{пов}}$ — матриця повороту.

Таблиця 5

m_x	0	0
0	m_y	0
0	0	1

Таблиця 6

$\cos \varphi$	$\sin \varphi$	0
$-\sin \varphi$	$\cos \varphi$	0
0	0	1

Вивід графічних елементів у растровій формі потрібно для відображення цих елементів на бітову карту растрової відеосистеми. Нехай потрібно розгорнути відрізок AB прямої $y = ax + b$, причому $1 \geq a \geq 0$ (при інших значеннях a розглянутий нижче алгоритм

залишається справедливим після певних модифікацій). Уведемо позначення: $\mathbf{A} = (x_a, y_a)$, $\mathbf{B} = (x_b, y_b)$; за величину дискрета (пікселя) приймемо одиницю. В алгоритмі розгорнення номера рядків і стовпців карти, на перетинанні яких повинні перебувати точки відрізка, визначаються в такий спосіб:

1. $\Delta x := x_b - x_a; \Delta y := y_b - y_a; x := x_a; y := y_a;$
2. $d := 2\Delta y - \Delta x;$
3. якщо $d \geq 0$, то $\{y := y + 1; d := d + 2(\Delta y - \Delta x)\}$; інакше $d := d + 2\Delta y;$
4. $x := x + 1;$
5. перехід до пункту 3, поки не досягнута точка \mathbf{B} .

Економічність цього алгоритму обумовлюється відсутністю довгих арифметичних операцій типу множення.

Виділення вікна потрібно при визначенні тієї частини сцени, що повинна бути виведена на екран дисплея.

Нехай вікно обмежене лініями $x = x_1, x = x_2, y = y_1, y = y_2$ (Рис. 4). По черзі для кожного багатокутника перевіряється розташування його вершин і ребер щодо границь вікна. Так, для багатокутника \mathbf{ABCD} (див. Рис. 4) при відсіканні по границі $x = x_2$ проглядається множина вершин у порядку обходу по годинній стрілці. Можливі чотири ситуації для двох послідовних вершин \mathbf{P} і \mathbf{R} :

1. якщо $x_P > x_2$ й $x_R > x_2$, то обидві вершини й інцидентне їм ребро перебувають поза вікном і виключаються з подальшого аналізу;
2. якщо $x_P \leq x_2$ й $x_R \leq x_2$, то обидві вершини й інцидентне їм ребро залишаються для подальшого аналізу;
3. якщо $x_P \leq x_2$ й $x_R > x_2$, то вершина \mathbf{P} залишається в списку вершин, а вершина \mathbf{R}_3 аміняється новою вершиною з координатами $x = x_2, y = y_P + (y_R - y_P)(x_2 - x_P)/(x_R - x_P)$; в нашому прикладі такою новою вершиною буде \mathbf{E} ;
4. якщо $x_P > x_2$ й $x_R \leq x_2$, то вершина \mathbf{P} заміняється новою вершиною з координатами $x = x_2, y = y_R + (y_P - y_R)(x_2 - x_R)/(x_P - x_R)$, а вершина \mathbf{R} залишається в списку вершин; у нашому прикладі новою вершиною буде \mathbf{F} .

Після закінчення перегляду стосовно до всіх границь у вікні виявляються вершини, що залишилися в списку.

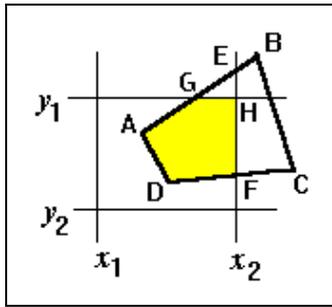


Рис. 4. Виділення вікна

Застосовуючи ці правила в нашому прикладі, одержуємо спочатку багатокутник **AEFD**, а після відсікання по верхній границі $Y = Y_2$ — багатокутник **AGFD** (див. рис. 4). Однак правильний результат трохи інший, а саме багатокутник **AGHFD**. Цей правильний результат виходить при подвійному обході вершин спочатку по годинній стрілці, потім проти із включенням у список нових вершин, що з'являються при кожному обході.

Застосовують ряд алгоритмів *видалення схованих ліній*. Один з найбільше просто реалізованих алгоритмів — алгоритм *z-буфера*, де *z-буфер* — область пам'яті, число осередків у якій дорівнює числу пікселів у вікні виводу. Передбачається, що вісь Z спрямована по нормалі до видової поверхні й спостерігач розташований у крапці $Z = 0$.

На початку виконання алгоритму всі пікселі відповідають максимальному значенню Z , тобто максимальному видаленню від спостерігача, що приводить до приміщення в усі осередки *z-буфера* значень пікселів тіла креслення. Далі по черзі для всіх точок граней розраховуються значення координати Z . Серед точок, що відносяться до того самого пікселя (одному й тому ж осередку *z-буфера* S), вибирається точка з найменшим значенням Z і її код (тобто кольори і яскравість) міститься в S . У підсумку *z-буфер* буде містити пікселі найбільш близьких до спостерігача граней.

Алгоритми побудови проєкцій перетворюють тривимірні зображення у двовимірні. У випадку побудови центральної проєкції кожна точка тривимірного зображення відображається на картинну поверхню шляхом перерахування координат x і y (Рис. 5). Так, координату x'_a точки A' обчислюють по очевидній формулі

$$x'_a = \frac{x_a d}{z},$$

аналогічно розраховується координата y'_a точки A' .

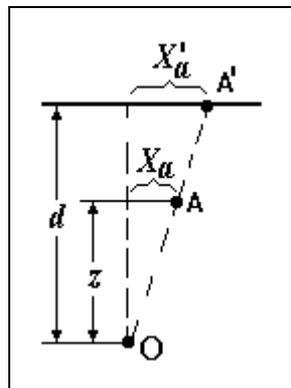


Рис. 5. Побудова центральної проєкції точки A

У паралельних проєкціях $d \rightarrow \infty$ і координати x й y точок A' і A збігаються. Тому побудова паралельних проєкцій зводиться до виділення вікна, при необхідності до повороту зображення й можливо до видалення схованих ліній.

Моделювання ефектів відбиття світла від поверхні об'єкта в геометричних моделях називають *рендерінгом*. Зафарбування матових поверхонь засновані на законі Ламберта, відповідно до якого яскравість відбитого від поверхні світла пропорційні $\cos \alpha$, де α — кут між нормаллю до поверхні й напрямком луча падаючого світла. В *алгоритмі Гуро* яскравість внутрішніх точок розглянутої поверхні визначається лінійною інтерполяцією яскравості у вершинах багатокутника. При цьому спочатку проводиться інтерполяція в крапках ребер, а потім по рядках горизонтального розгорнення. Більш реалістичними виходять зображення в *алгоритмі Фонга*, заснованому на лінійній інтерполяції векторів нормалей до поверхні. Один з алгоритмів рендерінгу полягає в *трасуванні променів* — моделюванні проходження променів світла між джерелами, поверхнями й спостерігачем.

11.3. Програми комп'ютерної графіки

Графічні редактори призначені для створення, збереження й відтворення графічних зображень. Розрізняють редактори векторні й растрові.

Векторні редактори дозволяють створювати зображення типу схем, креслень, діаграм, простих малюнків. Одним з найбільш відомих векторних редакторів є CorelDraw. Характерні можливості цього редактора – прив'язка ліній до певної позиції на зображенні, створення винесень і розмірних ліній, робота із багат шаровими зображеннями, стандартний інтерфейс Windows, багаторівневий відкрит, кросплатформеність (є версії для Windows, OS/2, MAC, різних варіантів UNIX). Приклади інших векторних редакторів – Microsoft Paint, iGrafx Designer, Visio 2000, Adobe Illustrator.

Растрові графічні редактори виконують такі функції, як відтворення складних зображень типу фотографій, їхнє редагування за допомогою операцій зафарбовування, обрізання, маскування, зміни параметрів усього зображення або його ділянок. У цьому класі графічних редакторів лідируюче положення займає Adobe Photoshop. При його використанні все зображення будується з набору окремих картинок, що мають прозорі й зафарбовані ділянки. Можна створювати, видаляти, копіювати, комбінувати ділянки, регулювати прозорість і порядок розташування шарів. У пакеті передбачені можливості побудови й редагування також векторних зображень. Завдяки підключенню додаткових модулів можна розширювати функціональність пакета Adobe Photoshop. Зокрема, тому за допомогою Adobe Photoshop вдається виконувати повний цикл робіт зі створення Web-сторінок. Один з модулів, що підключають, PhotoGraphics – додатковий засіб для малювання векторних об'єктів безпосередньо в Photoshop. У модулі користувачеві надаються всі основні інструменти малювання таких об'єктів, як лінії, прямокутники, овали, текст і т.п.

Інший растровий редактор Painter 6 дозволяє створювати зображення, що імітують картини художників. Поряд із цим у ньому реалізовані звичайні функції растрових редакторів, є можливість створювати анімаційні зображення.

Для відтворення складних зображень, наприклад фотографій, використовують переглядачі популярних 2D і 3D форматів; графічні й відео переглядачі й конвертори відповідних форматів.

Для синтезу тривимірних зображень у САПР переважно використовують засоби таких відомих систем, як CATIA, Unigraphics, Inventor, Solidworks і ін. Однак крім них, існують і застосовуються *графічні бібліотеки* й пакети комп'ютерної графіки й геометричного моделювання, безпосередньо не пов'язані із проектуванням у техніці. До числа графічних бібліотек, що одержали найбільше поширення, ставиться бібліотека графічних процедур

OpenGL.

У бібліотеці OpenGL є процедури побудови *графічних примітивів* (до числа яких відносять точки, лінії, полігони), видалення невидимих ліній, Z-буферизації, реалістичної засвітки, завдання положення тіл і камери спостереження й т.п. Примітиви можуть бути як векторними, так і растровими. Важливою особливістю технології OpenGL є інваріантність до апаратно-програмних платформ. Засоби OpenGL реалізовані в ряді програм, наприклад у програмі побудови анімаційних кадрів GL-animator.

Серед інших систем 3D-графіки виділяють редактор Studio Max 4. Це високорозвинений, але дорогий й досить складний в освоєнні комплекс моделювання, візуалізації й анімації об'єктів, розроблений фірмою Autodesk для операційного середовища Windows. Для генерації ландшафтів типу гірських пейзажів, водних поверхонь, фантастичних картин підходить 3D-редактор Bryce. До числа 3D-редакторів можна віднести відповідні компоненти із програмного забезпечення САПР, а також програми Adobe Dimension, CorelDream 3D, CosmoWorlds і ряд інших.

Лекція 12. Автоматизовані системи в промисловості

12.1. Системи ERP

Функції автоматизованих систем керування підприємством (АСКП) у різних сполученнях поєднуються в кілька груп, відповідно з'являються різновиди АСКП з назвами *ERP*, *MRP*, *MES*, *SCM* і ін. Звичайно функції керування поставками й відносинами із замовниками відносять до функцій *ERP*, але іноді ці функції реалізуються в самостійних системах *SCM* і *CRM* відповідно. Найбільше повно функції керування підприємствами представлені в системах *ERP* (Enterprise Resource Planning), тобто АСКП звичайно ототожнюють із системами *ERP*.

Характерні риси сучасних систем *ERP*:

1. Відкритість стосовно провідних платформ (UNIX, Windows, OS/2) і різним СКБД і насамперед потужним СКБД типу Oracle, Ingres, Informix, Sybase; підтримка технологій типу ODBC (Open Data Base Connection), OLE (Object Linking and Embedding), DDE (Dynamic Data Exchange); підтримка архітектур клієнт/сервер. Важлива характеристика – можливість роботи в середовищі розподілених обчислень.

2. Можливість наскрізного виконання всіх припустимих бізнес-функцій або їхньої частини, що забезпечується модульною побудовою (процедури, що виконують ці функції, часто називають бізнес-функціями, а маршрути рішення задач керування, що складаються з бізнес-функцій, називають бізнес-процесами).

3. Адаптуємість до конкретних замовників і умов ринку.

4. Наявність інструментальних засобів, у тому числі мови розширення або 4GL (мови четвертого покоління). Так, в R/3 використовується мова ABAP, в Elite Series - мова Informix-4GL.

5. Технічне забезпечення АСКП – комп'ютерна мережа, вузли якої розташовані як в адміністративних відділах підприємства, так і в цехах.

У сучасних системах *ERP* виділяють ряд підсистем. Нижче наведений список основних підсистем, що зустрічаються в багатьох системах *ERP*, разом із властивими ним функціями:

- календарне планування виробництва. Основні функції: сіткове планування виробництва, розрахунок потреб у потужностях і матеріалах, міжцехові специфікації й облік руху виробів, контроль виконання планів.

- оперативне керування виробництвом. Функції: супровід даних про виробу, контроль

браку й відходів, розрахунок норм витрати ресурсів, керування обслуговуючими підрозділами.

- керування проектами. Функції: сіткове планування проектних робіт і контроль їхнього виконання, розрахунок потреби у виробничих ресурсах.

- фінансово-економічне керування, бухгалтерський облік. Функції: облік коштів і виробничих витрат, маркетингові дослідження, ціноутворення, складання кошторисів витрат, ведення договорів і взаєморозрахунків, фінансові звіти, звітність по податках, аналіз платоспроможності підприємства.

- логістика. Функції: збут і торгівля, статистика й аналіз реалізації, складське обслуговування, керування постачанням, запасами й закупівлями, керування транспортуванням, оптимізація маршрутів транспортних засобів.

- керування персоналом. Функції: кадровий облік, ведення штатного розкладу, розрахунок зарплати.

- керування сервісним обслуговуванням устаткування й зробленої продукції (післяпродажним технічним обслуговуванням);

- керування ланцюжками поставок (цю функцію часто відносять до спеціальної системи SCM);

- керування інформаційними ресурсами. Функції: керування документами й документообігом, інсталяція й супровід ПЗ, генерація моделей і інтерфейсів систем, імітаційне моделювання виробничих процесів.

Найбільш популярними у світі й розвинених системах ERP є SAP R3, Vaan IV, Oracle Applications, J.D.Edwards, також виділяються системи Галактика, 1С:Підприємство, Флагман.

У системах ERP важлива роль приділяється системам керування даними EDM, аналогічним системам PDM у САПР.

12.2. Стандарт MRP II

Під *MRP* розуміють методологію ефективного планування ресурсів виробничого підприємства, включаючи бізнес-планування, планування продажів і операцій, планування виробництва, формування головного календарного плану виробництва, планування потреби в матеріалах, у потужностях, забезпечення підтримки виконання планів для виробничих потужностей і матеріалів.

Назва MRP відносять також до систем, що реалізують методологію MRP. Системи, сконцентровані на керуванні виробництвом, що оперують інформацією про матеріальні потоки й супровідних їх у процесі виробництва інформаційних потоках, називають системами планування ресурсів виробництва *MRP-2* (або MRP II — Manufacturing Resource Planning). Попередником системи MRP-2 була система планування потреб у матеріалах *MRP-1* (або MRP I — Material Requirement Planning).

Стандарти MRP/ERP підтримуються Американським товариством по контролю за виробничими запасами *APICS* (American Production and Inventory Control Society). MRP/ERP – це набір перевірених на практиці розумних принципів, моделей і процедур керування й контролю, призначених для підвищення показників економічної діяльності підприємства.

12.3. Підсистеми ERP

Типовий склад підсистем ERP:

- керування запасами;
- *керування постачанням*; реалізує наступні функції: формування замовлень на закупівлю; графіка поставок; планування потреби в матеріалах, що розуміється як керування заявками на закупівлю;
- *керування збутом*; базовими функціями цієї підсистеми є: квотування продажів; замовлення на продаж; графік продажів споживачам; конфігурування продуктів; аналіз продажів; керування ресурсами розподілу;
- транспортна логістика: керування транспортуванням вантажів, оптимізація маршрутів транспортних засобів;
- керування виробництвом;
- планування;
- *керування персоналом* – це система організаційних, соціально-економічних, психологічних, моральних і правових відносин, що забезпечують ефективну реалізацію можливостей людини в інтересах як самої людини, так і організації. Основні функції підсистеми: кадровий облік, ведення штатного розкладу, розрахунок зарплати.
- *керування сервісним обслуговуванням*, включаючи післяпродажне технічне обслуговування й технічну підтримку виробленої продукції. Підсистема забезпечує повний спектр необхідних функцій: від створення графіка технічного обслуговування, замовлення комплектуючих, обліку контрактів на обслуговування й формування рахунків до обліку прибутку, одержуваної від післяпродажного обслуговування.
- фінансово-економічне керування,
- *керування інформаційними ресурсами*. Функції: керування документами й документообігом, інсталяція й супровід програмного забезпечення, генерація моделей і інтерфейсів, імітаційне моделювання виробничих процесів.

12.4. Логістичні системи

Складність задач керування, які доводиться вирішувати в сучасних виробничих системах, обумовлює інтерес до логістики й розвитку логістичних систем. Традиційно *логістику* зв'язували з керуванням процедурами руху сировини від джерел постачання до місця виробництва продукції, її складування й руху від виробничої лінії до місця споживання. При цьому логістику визначають як науку про організації менеджменту для ефективного просування продукції по ланцюзі "закупівля сировини - виробництво продукції - збут - розподіл" з метою мінімізації загальних витрат ресурсів і задоволення вимог кінцевих споживачів до якості продукції й послуг. Функції й задачі, виконувані в підсистемах "Логістика" більшості систем ERP, відповідають цьому визначенню.

Однак часто використовується й більш широке трактування логістики, як науки про керування й оптимізацію матеріальних потоків, потоків послуг і пов'язаних з ними фінансових і інформаційних потоків при функціонуванні різних систем. Системи, що реалізують керування названими потоками, називають *логістичними системами*. Основними логістичними функціями є підтримка післяпродажного обслуговування споживачів продукції, керування закупівлями й замовленнями, транспортування, керування запасами, керування виробничими процедурами, ціноутворення, розподіл готової продукції в товаропровідних структурах підприємства й посередників. При цьому під управлінням виробничими процедурами мається на увазі рішення задач об'ємно-календарного планування, мінімізації рівня запасів матеріальних ресурсів і незавершеного виробництва, розрахунку потреб у

матеріалах і т.п.

12.5. Системи SCM

Системи *SCM* призначені для забезпечення ефективного керування матеріальними й відповідними їм інформаційними потоками: від постачальників сировини й комплектуючих через виробництво до продажу готових виробів кінцевому споживачеві. Реалізована в підсистемі ідеологія "керування глобальними ланцюжками поставок" дає промисловим підприємствам можливість представляти свою діяльність у вигляді так званих ефективних ланцюжків логістики: від постачальників сировини й комплектуючих до продажу готових виробів кінцевому споживачеві. При цьому забезпечуються широкі можливості керування транснаціональними компаніями, координації розподіленого між багатьма дочірніми компаніями виробництва. Приклад реалізації функцій *SCM* – логістична підсистема середовища R3. Серед цих функцій: облік і планування виробництва, попиту на вироблену продукцію, постачальників, маршрутів перевезень і розміщення продукції.

Системи *SCM* можуть створюватися в складі систем *ERP* або як самостійні.

12.6. CRM - системи взаємин із замовниками

Система *CRM* — це система, на вхід якої надходять дані, пов'язані із клієнтами компанії, а на виході з'являється інформація, що впливає на поведінку компанії в цілому або на поведінку її окремих елементів (аж до конкретного працівника компанії). Інакше кажучи, *CRM*-система – це насамперед база даних з інформацією про клієнтах. і набір додатків, які дозволяють, по-перше, збирати інформацію про клієнта, по-друге, її обробляти, у третіх, робити певні висновки на базі цієї інформації, експортувати її в інші додатки або просто при необхідності надавати цю інформацію в зручному виді. Властиво, ці моменти і є ключовими функціями *CRM*-систем. Результатами роботи *CRM*-системи можуть користуватися не тільки співробітники компанії, але й безпосередньо сам клієнт. Причому сучасні технології дозволяють це зробити без якої-небудь допомоги співробітників організації.

Існують *ERP*-системи в які *CRM* виконують роль підсистем (наприклад, R3, Oracle E-Business Suite, Ахарта) однак у цей час частіше *CRM* є самостійними системами (наприклад, Siebel, WinPeak CRM, Sales Expert).

Лекція 13. Програмування для верстатів з ЧПК

Вихідними даними для складання програм для верстатів із ЧПК є результати конструкторського проектування, що надходять із *CAD*. Але можливе програмування й при наявності в якості вихідних даних лише креслення деталі й параметрів технологічного процесу.

При програмуванні визначають і кодують геометрію заготовівлі, траєкторії руху рухливих органів верстата й параметри обробки. Для цих цілей використовують спеціалізовані мови, прикладом яких може служити мова *APT* (Automatically Programmed Tools), що відноситься до мов високого рівня. Отриманий вихідний код мовою *APT* перетворюється в програму переміщень інструмента, керування подачею й т.п., що представляє у вигляді апаратно незалежного файлу *CLData* (Cutter Location Data). Файл *CLData* надходить у постпроцесор, що переводить програму у вид, необхідний для конкретного типу верстата.

У мові *APT* є наступні групи команд:

- ідентифікуючі – для вказівки назви оброблюваної деталі й типу використовуваного постпроцесору;
- геометричні – для вказівки геометричних особливостей деталі;
- керуючі переміщеннями різального інструменту;
- керуючими режимами обробки (визначальні швидкість подачі, швидкість обертання шпинделя, включення охолодження й т.п.);
- додаткові (наприклад, вибір інструмента).

Приклади команд АРТ:

- **P5 = POINT/0.0. 2.5. 0.4** — завдання точки P5 з координатами X=0, Y=2,5, Z=0,4.
- **GOTO/P7** — переміщення в точку P7.
- **FEDRAT/6.0** — завдання швидкості подачі 6 дюйм/хв.

12.1. Типи САПР в області машинобудування

Серед САД-систем розрізняють системи нижнього, середнього й верхнього рівнів. Цей поділ виник на рубежі 80-90-х років минулого століття. Системами нижнього рівня (або легкими системами) стали називати порівняно дешеві САПР, орієнтовані на 2D-графіку, тобто на автоматизацію переважно креслярських робіт. Технічним забезпеченням легких САПР були персональні ЕОМ, що у той час значно уступали по своїх можливостях робочим станціям.

Системи верхнього рівня, називані також "*важкими*" САПР (або hi-end), розроблялися для реалізації на робочих станціях або мейнфреймах. Ці системи були більш універсальними, але й дорогими, орієнтованими на геометричне твердотільне й поверхневе моделювання. Оформлення креслярської документації в них звичайно здійснюється за допомогою попередньої розробки тривимірних геометричних моделей. Надалі системи, у яких 3D-моделювання обмежувалося лише твердотільними моделями, тобто займаючи проміжне положення між "легкими" і "важкими" САПР, стали називати системами середнього рівня.

У цей час розвиток САПР привело до того, що в багатьох системах середнього рівня з'явилися засоби поверхневого моделювання, а можливості персональних ЕОМ стали прийнятними для систем верхнього рівня. У результаті змінилися принципи, по яких розрізняють важкі й середні системи. Важкими тепер називають системи CAE/CAD/CAM/PDM, тобто системи з можливостями конструкторського й технологічного проектування, інженерного аналізу, керування проектними даними й з розширеним складом спеціалізованих програмних модулів у підсистемах САД і САМ.

Системи одного рівня по своїх функціональних можливостях приблизно рівноцінні, нові досягнення, що з'явилися в одному із програмно-методичних комплексів САПР, незабаром реалізуються в нових версіях інших комплексів.

У САПР великих підприємств звичайно використовують програми різних рівнів. Зв'язано це з тим, що більше 80% всіх процедур конструювання можна виконати на САД-системах нижнього й середнього рівнів, крім того, "важкі" системи дорогі. Тому підприємство здобуває лише обмежене число екземплярів (ліцензій) програми верхнього рівня, а більшість клієнтських робочих місць забезпечується екземплярами програм нижнього або середнього рівнів. При цьому виникає проблема обміну інформацією між різнотипними САД-системами. Вона вирішується шляхом використання мов і форматів, прийнятих в CALS-технологіях, хоча для неспотвореної передачі геометричних даних за допомогою проміжних уніфікованих мов доводиться переборювати певні труднощі.

13.2. Основні функції САD-систем

Функції САD-систем у машинобудуванні підрозділяють на функції двомірного (2D) і тривимірного (3D) проектування. До функцій 2D відносять креслення, оформлення конструкторської документації; до функцій 3D — одержання тривимірних геометричних моделей, метричні розрахунки, реалістичну візуалізацію, взаємне перетворення 2D і 3D моделей. Тривимірні моделі представляють у вигляді опису поверхонь, що обмежують деталь, або вказівкою елементів простору, займаних тілом деталі. Моделі поверхонь складної форми одержують за допомогою різновидів *кінематичного методу*, до яких відносять витягування заданого плоского контуру по нормалі до його площини, протягання контуру уздовж довільній просторовій кривій, обертання контуру навколо заданої осі, натягування поверхні між декількома заданими перетинами. У випадку побудови скульптурних поверхонь, що проходять через задані крапки простору, застосовують моделі у формі Без'є, а при вимогах високої гладкості поверхні – моделі у формі В-сплайнов. Синтез моделей зборок виконують з застосуванням операцій позиціонування й теоретико-множинних операцій перетинання, об'єднання, вирахування до бібліотечних елементів і знову створеним моделям комплектуючих деталей. У ряді систем передбачене також виконання операцій компонування й розміщення устаткування і т.п.

До важливих характеристик САD-систем відносяться *параметризація* й *асоціативність*. Параметризація має на увазі використання геометричних моделей у параметричній формі, тобто при поданні частини або всіх параметрів об'єкта не константами, а змінними. *Параметрична модель*, що перебуває в базі даних, легко адаптується до різних конкретних реалізацій і тому може використатися в багатьох конкретних проектах. При цьому з'являється можливість включення параметричної моделі деталі в модель складаемого вузла з автоматичним визначенням розмірів деталі, диктуємих просторовими обмеженнями. Ці обмеження у вигляді математичних залежностей між частиною параметрів зборки відбивають асоціативність моделей.

Параметризація й асоціативність відіграють важливу роль при проектуванні конструкцій вузлів і блоків, що складаються з великої кількості деталей. Дійсно, зміна розмірів одних деталей впливає на розміри й розташування інших. Завдяки параметризації й асоціативності зміни, зроблені конструктором в одній частині зборки, автоматично переносяться в інші частини, викликаючи зміни відповідних геометричних параметрів у цих частинах.

Лекція 14. Основні функції САЕ-систем

Функції *САЕ-систем* досить різноманітні, тому що пов'язані із проектними процедурами аналізу, моделювання, оптимізації проектних рішень. До складу машинобудівних САЕ-систем насамперед включають програми для виконання наступних процедур:

- моделювання полів фізичних величин, у тому числі аналіз міцності, що найчастіше виконується відповідно до МСЕ;
- розрахунок станів моделюємих об'єктів і перехідних процесів у них засобами макрорівня;
- імітаційне моделювання складних виробничих систем на основі моделей масового обслуговування й мереж Петрі.

Основними частинами програм аналізу за допомогою МСЕ є бібліотеки скінченних елементів, препроцесор, решатель і постпроцесор.

Бібліотеки скінченних елементів (СЕ) містять моделі СЕ та їхньої матриці жорсткості.

Очевидно, що моделі СЕ будуть різними для різних задач (аналіз пружних або пластичних деформацій, моделювання полів температур, електричних потенціалів і т.п.), різних форм СЕ (наприклад, у двовимірному випадку – трикутні або чотирикутні елементи), різних наборів координатних функцій.

Вихідні дані для препроцесору – геометрична модель об'єкта, найчастіше одержувана з підсистеми конструювання. Основна функція препроцесору – вивід досліджуваного середовища (деталі) у сітковому виді, тобто у вигляді множини скінченних елементів.

Рішатель – програма, що асемблює (збирає) моделі окремих СЕ в загальну систему алгебраїчних рівнянь і вирішує цю систему одним з методів розріджених матриць.

Постпроцесор служить для візуалізації результатів рішення в зручній для користувача формі. У машинобудівних САПР це графічна форма. Користувач може бачити вихідну (до навантаження) і деформовану форми деталі, поля напруг, температур, потенціалів і т.п. у вигляді кольорових зображень, у яких палітра кольорів або інтенсивність світіння характеризують значення фазової змінної.

14.1. Основні функції САМ-систем

Основні функції *САМ-систем*: розробка технологічних процесів, синтез керуючих програм для технологічного встаткування із ЧПК, моделювання процесів обробки, у тому числі побудова траєкторій відносного руху інструмента й заготівлі в процесі обробки, генерація постпроцесорів для конкретних типів устаткування із ЧПК, розрахунок норм часу обробки.

Вихідними даними для складання програм для верстатів із ЧПК є результати конструкторського проектування, що надходять із САД. Але можливе програмування й при наявності в якості вихідних даних лише креслення деталі й параметрів технологічного процесу.

При програмуванні визначають і кодують геометрію заготівлі, траєкторії руху рухливих органів верстата й параметри обробки. Для цих цілей використовують спеціалізовані мови, прикладом яких може служити мова АРТ (Automatically Programmed Tools), що відноситься до мов високого рівня.

14.1. Прототипування

Прототипування — виготовлення прототипів деталей або шаблонів, по яких деталі будуть виготовлятися. Прототипування в САД/САМ – безпосередня реалізація розробленої геометричної моделі.

Для прототипування широко використовується *стереолітографія*, заснована на побудові тривимірного об'єкта з ряду шарів фотополімеру, що вибірково отвержується при опроміненні.

Процес стереолітографії реалізується за допомогою установки, у якій є ванна з рідким полімером і вертикально переміщувана платформа. Платформа при формуванні чергового шару прототипу розташовується нижче поверхні рідкого полімеру на товщину одного шару. Промінь лазера переміщується по ділянці поверхні, що повторює форму перетину прототипу. Ця ділянка затвердіває. Послідовно шар за шаром, починаючи з нижнього шару, формується твердий прототип.

Процес стереолітографії може бути використаний для остаточного виготовлення деталі, якщо для неї полімер є підходящим матеріалом.

Поряд зі стереолітографією використовуються й інші способи прототипування, наприклад, *ламінівання* (LOM — Laminated Object Manufacturing), засноване на послідовному склеюванні шарів робочого матеріалу, що надходить у формі рулону. В установці ламінівання

лазер вирізує шар за формою необхідного перерізу.

Лекція 15. Огляд машинобудівних САПР

Як правило, машинобудівні САПР мають багатомодульну структуру. Типовий поділ модулів на групи програм конструкторського проектування механічних об'єктів, промислового дизайну, інженерного аналізу (функціонального моделювання), технологічного проектування, обміну даними, візуалізації.

У цей час у світі залишилися три реально розвиваються машинобудівні САПР верхнього рівня. Це Unigraphics (UGS - Unigraphics Solution), CATIA (Dessault Systemes), Pro/Engineer (PTC — Parametric Technology Corporation). Донедавна з ними конкурували системи I-DEAS (SDRC), CADD5 (Computervision) і EUCLID (Matra Datavision).

Dessault Systemes, що працює у зв'язку з IBM, поширює систему CATIAv5, у якій використані результати розробок як CATIAv4, так і Euclid Quantum, отримані від Matra Datavision.

Після придбання компанії SDRC корпорацією EDS остання утворила у своєму складі підрозділ EDS PLM Solution, що потім став самостійною компанією Unigraphics Solution. Нова версія Unigraphics, що одержала назву Unigraphics NX, втілила в собі кращі властивості Unigraphics і I-DEAS. Наприклад, в I-DEAS були розвинені засоби інженерного аналізу, але на відміну від Unigraphics слабкі засоби САМ.

Компанія Computervision увійшла до складу PTC ще в 90-роки минулого століття, після чого розвиток CADD5 було припинено. У теперішній час PTC здійснює впровадження нової версії Pro/Engineer за назвою Wildfire.

До числа тенденцій у розвитку CAD/CAM/CAE-систем варто віднести:

- розвиток САПР, як складової частини системи PLM, тобто інтеграція САПР із іншими системами, що підтримують життєвий цикл виробів;
- реалізація можливостей спільного проектування на базі використання Internet.

Система *Unigraphics NX* — універсальна система геометричного моделювання й конструкторсько-технологічного проектування, у тому числі розробки більших зборок, міцностних розрахунків і підготовки конструкторської документації. У ній використовується концепція майстр-процесів – засобів інтерактивного проектування, що враховують особливості конкретних застосувань. У конструкторській частині (підсистема САД) є засоби для твердотільного конструювання, геометричного моделювання на основі сплайнових моделей поверхонь, створення креслень по 3D моделі, проектування зборок (у тому числі із сотнями й тисячами компонентів) з обліком асоціативності, аналіз допусків і ін. Як графічне ядро використовується Parasolid. У технологічній частині (підсистема САМ) передбачені розробка керуючих програм для токарської й електроерозійної обробки, синтез і аналіз траєкторій інструмента при фрезерній трьох- і п'ятикоординатної обробці, при проектуванні прес-форм і штампів і ін. Для інженерного аналізу (підсистема САЕ) у систему включені модулі прочностного аналізу з використанням МСЕ з відповідними пре- і постпроцесорами, кінематичного й динамічного аналізу механізмів з визначенням сил, швидкостей і прискорень, аналізу литєвих процесів пластичних мас.

Система *CATIA V5* використовується на етапах від створення концепції виробу до технологічної підтримки виробництва й планування виробничих ресурсів. У системі реалізоване поверхневе й твердотільне 3D моделювання й оптимізація характеристик виробів. Використовується оригінальне графічне ядро CNEXT. Можливі фотореалістична візуалізація,

відновлення математичної моделі з матеріального макета. Система масштабована. Пропонуються типові конфігурації, у тому числі варіанти для повнофункціонального наскрізного проектування складних виробів і проектування комплектуючих на невеликих і середніх підприємствах. До особливостей системи можна віднести наявність модуля формалізації знань, спрямованих на оптимізацію конструкцій і процесів проектування, і антропологічного модуля для моделювання деяких параметрів людини в людино-машинних системах.

У системі *Pro/Engineer*, що веде свою історію з 1986 р., базові модулі конструкторського проектування призначені для твердотілого й поверхневого моделювання, синтезу конструкцій з базових елементів форми, підтримки параметризації й асоціативності, проекційного креслення й розробки креслень із проставлянням розмірів і допусків. Користувач може поповнювати бібліотеку елементів форми (БЕФ) оригінальними моделями. Для САПР компанії РТС розроблене графічне ядро *Granite One*. Додаткові модулі конструкторського проектування мають більш конкретну, але вузьку спеціалізацію. Прикладами таких модулів можуть служити модулі конструювання панелей з композиційних матеріалів, розробки штампів і ливарних прес-форм, трубопровідних систем, зварених конструкцій, розведення електричних кабелів і джгутів. Модулі функціонального моделювання використовуються як препроцесори й постпроцесори в програмах скінченноелементного аналізу (нанесення сітки скінченних елементів, візуалізація результатів аналізу), для аналізу теплового стану конструкцій, оцінки вібростійкості й ін. Основні модулі технологічного проектування служать для моделювання технологічних процесів фрезерної, токарської, електроерозійної обробки й для розробки постпроцесорів для систем керування встаткуванням із ЧПК, хоча САМ-засоби в *Pro/Engineer* не є сильною стороною системи. Модулі обміну даними (конвертори форматів даних) повинні забезпечувати можливості імпорту/експорту даних в інші САЕ/CAD/CAM-системи.

До числа особливостей нової версії *Wildfire* відноситься використання Internet для організації спільного проектування, у тому числі для організації відеоконференцій розроблювачів. Помітно спрощений інтерфейс користувача, завдяки введенню ієрархії піктограм.

Лідуюче положення в класі САПР середнього рівня займають системи *Solid Works* (*Solid Works Corporation*), *Solid Edge* (*EDS PLM Solution*), *Inventor* (*Autodesk*). Компанія РТС також має САПР середнього рівня за назвою *Pro/Desktop*. У нас нарівні з ними знайшли поширення російські системи *Компас* (*Аскон*) і *T-Flex CAD* (*Топ Системи*), а також деякі інші системи, у числі яких САПР компаній *Autodesk*, *Beantly*, *Інтермех*, *Bee-Pitron*. Всі ці системи, як правило, мають підсистеми конструкторсько-креслярського 2D, твердотілого 3D геометричного моделювання, технологічного проектування, керування проектними даними, ряд підсистем інженерного аналізу й розрахунку окремих видів машинобудівних виробів, а також бібліотеки типових конструктивних рішень.

Система твердотілого параметричного моделювання механічних конструкцій *Solid Works* побудована на графічному ядрі *Parasolid*. Синтез конструкції починається з побудови опорного тіла за допомогою операцій типу видавлювання, протягання або обертання контуру з наступним додаванням та/або вирахуванням тих або інших тіл. Використовується технологія граничного моделювання (*B-representation*) з аналітичним або сплайновим описом поверхонь. При проектуванні зборок на основі БЕФ можна задавати різні умови взаємного розташування деталей, автоматично контролювати зазори й відсутність взаємного пересікання деталей. Передбачені IGES, DXF, DWG-інтерфейси з іншими системами. У систему входить модуль міцностних розрахунків *Cosmos/Works*.

Істотною перевагою системи *Solid Edge* є високий ступінь інтегруємості із системою високого рівня *Unigraphics*, тому що системи підтримує одна компанія, у них

використовується одне графічне ядро. Технологія конструювання в Solid Edge аналогічна до інших розвинених систем: конструктивні елементи задаються у вигляді перетинів 3D форми, ці перетини шляхом витягування або обертання перетворюють у модель деталі.

Система *Inventor* призначена для твердотілого параметричного проектування, орієнтована на розробку великих зборок із сотнями й тисячами деталей, має розвинену бібліотеку стандартних елементів. Система побудована на графічному ядрі ACIS (на базі цього ядра компанія Autodesk розробила власне ядро Shape Manager). Синтез 3D-моделей можливий видавлюванням, обертанням, по перетинах, по траєкторіях. З 3D-моделі можна одержати 2D креслення й специфікації матеріалів. Підтримується колективна робота над проектом, у тому числі в межах однієї й тієї ж зборки. Передбачено автоматичну перевірку кінематики, розмірів деталі з урахуванням положення сусідніх деталей у збірці. Значні зручності роботи конструкторів обумовлені тим, що асоціативні зв'язки задаються не шляхом опису операцій з параметрами й рівняннями, а безпосередньо визначенням форми й положення компонентів.

Autodesk поставляє також системи середнього рівня MDT (Mechanical Desktop) і нижнього рівня AutoCAD LT.

У системі *Компас* для тривимірного твердотілого моделювання використовується оригінальне графічне ядро. Синтез конструкцій виконується за допомогою булевських операцій над об'ємними примітивами, моделі деталей формуються шляхом видавлювання або обертання контурів, побудовою по заданих перетинах. Можливе завдання залежностей між параметрами конструкції, розрахунок масінерційних характеристик. Розробка проектно-конструкторської документації, у тому числі різних специфікацій, виконується підсистемою Компас-Графік. Є бібліотеки з даними про типові деталі й графічними зображеннями, а також програми спеціального призначення (проектування тіл обертання, пружин, металоконструкцій, трубопровідних арматур, штампового оснащення, вибору підшипників кочення, розкрою листового матеріалу й ін.). Проектування технологічних процесів виконується за допомогою підсистеми Компас Автопроект, програмування об'ємної обробки на верстатах із ЧПК – за допомогою підсистеми ГЕМА-3D. Ряд необхідних функцій керування проектними даними покладено на підсистему Компас-Менеджер.

Підсистема тривимірного твердотілого моделювання T-Flex CAD 3D у САПР *T-Flex CAD* побудована на базі ядра Parasolid. Реалізована двонаправлена асоціативність, тобто зміна параметрів креслення автоматично викликає зміну параметрів моделі й навпаки. При проектуванні зборок зміна розмірів або положення однієї деталі веде до коректування положення інших. Модель 3D може бути отримана безпосередньо по наявному кресленню, або за допомогою булевських операцій, або шляхом виштовхування, протягання, обертання профілю й т.п. Передбачений розрахунок масінерційних параметрів. З іншого боку, можна по видах і розрізам тривимірної моделі одержати креслення, для чого використовується підсистема T-Flex CAD 3D SE. Для параметричного проектування й оформлення конструкторсько-технологічної документації служить підсистема T-Flex CAD 2D, для керування проектами й документообігом - підсистема T-Flex DOCs. У підсистемі технологічного проектування T-Flex/Технопро виконуються синтез технологічних процесів, розрахунок технологічних розмірів, вибір ріжучого й допоміжного інструмента, формування технологічної документації, у тому числі операційних і маршрутних технологічних карт, відомостей оснащення й матеріалів, карт контролю. Підготовка програм для верстатів із ЧПК здійснюється в підсистемі T-Flex ЧПК. Крім названих основних підсистем до складу T-Flex CAD включений ряд програм для інженерних розрахунків деталей, проектування штампів і прес-форм.

Розробкою продуктів для САПР ливарного виробництва займається компанія Moldflow, її програми Part Adviser і Mold Adviser призначені для моделювання процесів лиття пластмас.

Важливе місце в конструкторсько-технологічних САПР займають програми

технологічної підготовки виробництва.

Одним з лідерів у цій області є система Mastercam. У системі є модулі розрахунку керуючих програм різних видів обробки заготовель, програми моделювання й доробки моделей, конвертори ряду популярних графічних форматів даних, наприклад, форматів STEP, IGES, Parasolid (xmt, txt), SAT ядра ACIS (sat), систем AutoCAD (dxf, dwg), Inventor (ipt, iam), SolidWorks (sldprt) і ін.

Компанія Consistent Software пропонує систему TechnologiCS для технологічної підготовки дискретного виробництва. Ця система виконує функції складання специфікацій, ведення дерева проекту й бібліотеки креслень, синтезу технологічних процесів, вибору інструмента, розрахунку режимів різання, нормування витрати матеріалів, ведення технологічної документації. Система SolidCAM (CADTech), побудована як і Mechanical Desktop на ядрі ACIS, служить для одержання керуючих програм для токарської, 2,5- і 3-осьової фрезерної обробки на верстатах із ЧПК. Система ТЕХТРАН (НИП "Інформатика") включає модулі токарської, фрезерної, електроерозійної обробки.

Світові лідери серед програм скінченноелементного аналізу є програмно-методичні комплекси Nastran і Patran (компанія MSC Software Corporation) і Ansys (компанія Ansys Inc.).

Як правило, ці комплекси містять у собі ряд програм, близьких по математичному забезпеченню, інтерфейсам, спільності деяких використовуваних модулів. Ці програми розрізняє орієнтація на різні додатки, ступень спеціалізації, ціна або виконувана обслуговуюча функція. Наприклад, у комплексі Ansys основні модулі дозволяють виконувати аналіз механічної міцності, теплопровідності, динаміки рідин і газів, акустичних і електромагнітних полів. В усі варіанти програм входять пре- і постпроцесори, а також інтерфейс із БД. Передбачено експорт (імпорт) даних між Ansys і провідними комплексами геометричного моделювання й машинної графіки.

Світовий лідер серед засобів моделювання механічних процесів на макрорівні шляхом рішення СОДР — програма *Adams*.

Лекція 16. Методичне й програмне забезпечення автоматизованих систем

16.1. Типи CASE-систем

У сучасних інформаційних технологіях важливе місце приділяється інструментальним засобам і середовищам розробки АС, зокрема, системам розробки й супроводу їх ПЗ. Ці технології й середовища утворюють системи, що мають назву CASE-систем.

Використовується двоєке тлумачення аббревіатури CASE, що відповідає двом напрямкам використання CASE-систем. Перше з них — *Computer Aided System Engineering* — підкреслює спрямованість на підтримку концептуального проектування складних систем, переважно слабкоструктурованих. Далі CASE-системи цього напрямку будемо називати системами CASE для концептуального проектування. Друга назва — *Computer Aided Software Engineering* переводиться, як автоматизоване проектування програмного забезпечення.

Серед систем CASE для концептуального проектування розрізняють системи функціонального, інформаційного або поведінкового проектування. Найбільш відомою методикою функціонального проектування складних систем є *методика SADT* (Structured Analysis and Design Technique), запропонована в 1973 р. Р. Росом і згодом стала основою міжнародного стандарту IEEE 1320.1-1998 IDEF0 (Integrated DEFinition 0).

Системи інформаційного проектування реалізують методики інфологічного проектування БД. Широко використаються мова й методика створення інформаційних моделей програм, закріплені в методиці IDEF1X (IEEE 1320.2-1998). Крім того, розвинені комерційні СКБД, як правило, мають у своєму складі сукупність CASE-засобів проектування

програм.

Основні положення стандартів IDEF0 і IDEF1X використані також при створенні комплексу стандартів ISO 10303, що лежать в основі технології STEP для виводу в комп'ютерних середовищах інформації, що відноситься до проектування й виробництва в промисловості.

Поведінкове моделювання складних систем використовують для визначення динаміки функціонування складних систем. У його основі лежать моделі й методи імітаційного моделювання систем масового обслуговування, мережі Петрі, можливе застосування кінцево-автоматних моделей, що описують поведінку системи, як послідовності зміни станів.

Застосування інструментальних CASE-систем веде до скорочення витрат на розробку ПЗ за рахунок зменшення числа ітерацій і числа помилок, до поліпшення якості продукту за рахунок кращого взаєморозуміння розроблювача й замовника, до полегшення супроводу готового ПЗ,

Серед інструментальних CASE-систем розрізняють інтегровані комплекси інструментальних засобів для автоматизації всіх етапів життєвого циклу ПЗ (такі системи називають Workbench) і спеціалізовані інструментальні засоби для виконання окремих функцій (Tools). Засоби CASE-систем по своєму функціональному призначенню належать до однієї з наступних груп

1. засоби програмування;
2. засоби керування програмним проектом;
3. засоби верифікації (аналізу) програм;
4. засоби документування.

До засобів програмування відносяться компілятори для алгоритмічних мов; будівник діаграм потоків даних; планувальники для побудови високорівневих специфікацій і планів ПЗ (можливо на основі баз знань, реалізованих в експертних системах); інтерпретатори *мов специфікацій* і *мов четвертого покоління*; прототайпер для розробки зовнішніх інтерфейсів — екранів, форм вихідних документів, сценаріїв діалогу; генератори програм певних класів (наприклад, конвертори заданих мов, драйвери пристроїв програмного керування, постпроцесори); крос-засоби; відладчики програм. При цьому під мовами специфікацій розуміють засоби укрупненого опису розроблювальних алгоритмів і програм, до мов 4GL відносять мови для компіляції програм з набору готових модулів, що реалізують типові функції досить загальних додатків (найчастіше це функції техніко-економічних систем).

Керування програмним проектом називають також керуванням конфігураціями ПЗ. Цьому поняттю відповідають коректне внесення змін в програмну систему при її проектуванні й супроводі, контроль цілісності проектних даних, керування версіями проекту, організація паралельної роботи членів колективу розроблювачів. Використання засобів керування конфігураціями дозволяє створювати програмні системи із сотень і тисяч модулів, значно скорочувати строки розробки, успішно модернізувати вже поставлені замовникам системи.

Основою засобів керування програмним проектом є репозиторій – БД проекту. Саме в репозиторії відбита історія розвитку програмного проекту, утримуються всі створені версії (вихідний програмний код, що виконують програми, бібліотеки, супровідна документація й т.п.) за допомогою репозиторію здійснюється контроль і відстеження внесених змін.

Засоби верифікації служать для оцінки ефективності виконання розроблювальних програм і визначення наявності в них помилок і протиріч. Розрізняють статичні й динамічні аналізатори. У статичних аналізаторах ПЗ досліджується на наявність невизначених даних, нескінченних циклів, неприпустимих передач керування й т.п. Динамічний аналізатор функціонує в процесі виконання програми, яку перевіряє; при цьому досліджуються траси,

вимірюються частоти звертань до модулів і т.п. Використовуваний математичний апарат – мережі Петрі, теорія масового обслуговування.

В останню з перерахованих груп входять документатори для оформлення програмної документації, наприклад, звітів по даним репозиторію; різні редактори для об'єднання, поділу, заміни, пошуку фрагментів програм і інших операцій редагування.

Проектування ПЗ за допомогою CASE-систем містить у собі кілька етапів. Початковий етап – попереднє вивчення проблеми. Результат представляють у вигляді вихідної діаграми потоків даних і погодять із замовником. На наступному етапі виконують деталізацію обмежень і функцій програмної системи, і отриману логічну модель знову погодять із замовником. Далі розробляють фізичну модель, тобто визначають модульну структуру програми, виконують інфологічне проектування БД, деталізують граф-схеми програмної системи і її модулів.

Підсистема CASE у складі системного середовища САПР призначена для адаптації САПР до потреб конкретних користувачів, розробки й супроводу прикладного ПЗ. Її можна розглядати як спеціалізовану САПР, у якій об'єктом проектування є нові версії підсистем САПР, зокрема, версії, адаптовані до вимог конкретного замовника. Інакше кажучи, такі CASE-підсистеми дозволяють користувачам формувати порівняно з малими витратами зусиль варіанти прикладних програмно-методичних комплексів (ПМК) з наявного базового набору модулів під заданий вузький діапазон конкретних умов проектування. У таких випадках CASE-підсистеми називають інструментальними середовищами.

CASE-система, як система проектування ПЗ, містить компоненти для розробки структурних схем алгоритмів і "екранів" для взаємодії з користувачем в інтерактивних процедурах, засоби для інфологічного проектування БД, налагодження програм, документування, збереження "історії" проектування й т.п. Поряд із цим, в CASE-підсистемі САПР входять і компоненти зі специфічними для САПР функціями.

Так, до складу САПР Microstation (фірма Bentley Systems) включене інструментальне середовище Microstation Basic і мова MDL (Microstation Development Language) з відповідною програмною підтримкою. Мова MDL – C-подібна, з її допомогою можна лаконічно виразити звертання до проектних операцій і процедур. У цілому середовище Microstation Basic близьке за своїми функціями до середовища MS Visual Basic, у ній є генератор форм, редактор, конструктор діалогу, відладчик.

Інший приклад CASE-системи в складі САПР – система CAS.CADE фірми MatraDatavision, з її допомогою фірма у свій час розробила чергову версію Euclid Quantum своєї САПР Euclid.

Лекція 17. Специфікації проектів програмних систем

Важливе значення в процесі розробки ПЗ мають засоби специфікації проектів ПЗ. Засоби специфікації значною мірою визначають суть методів CASE.

Способи й засоби специфікації класифікують по базовій методології, що використовується для декомпозиції ПЗ, як складної системи, і по аспектах моделювання ПЗ.

Розрізняють два підходи до декомпозиції ПЗ. Перший спосіб називають функціональним або структурним. Він заснований на виділенні функцій і потоків даних. Другий спосіб – об'єктний, виражає ідеї об'єктно-орієнтовного проектування й програмування.

Аспектами моделювання додатків є функціональний, поведінковий і інформаційний описи.

Практично всі способи функціональних специфікацій мають наступні загальні риси:

- модель має ієрархічну структуру, що представляється у вигляді діаграм декількох рівнів;
- елементарною частиною діаграми кожного рівня є конструкція вхід-функція-вихід;
- необхідна додаткова інформація утримується у файлах тексту, що пояснює.

У більшості випадків функціональні діаграми є *діаграмами потоків даних* (DFD — Data Flow Diagram). В DFD блоки (прямокутники) відповідають функціям, дуги – вхідним і вихідним потокам даних. Текст, що пояснює, представлений у вигляді "словників даних", у яких зазначені компонентний склад потоків даних, число повторень циклів і т.п.

Лекція 18. Програмне забезпечення CASE-систем

На ринку програмних продуктів є багато CASE-систем для концептуального проектування АС.

Найчастіше в них підтримується методологія IDEF. Широко відомі програми BPwin, ERwin, OOWin фірми Platinum Technology, Design/IDEF фірми Meta Software і ін.

BPwin (Business Processing) призначена для розробки функціональних моделей за методикою IDEF0.

ERwin призначена для розробки інформаційних моделей за методикою IDEF1X. Є засоби, що забезпечують інтерфейс із серверами БД (від користувача сховане спілкування на SQL-мові), переклад графічних зображень ER-діаграм в SQL-форми або у формати інших популярних СКБД. Передбачено інтерактивні процедури для зв'язування дуг IDEF0 із атрибутами IDEF1X, тобто для встановлення зв'язків між BPwin і ERwin. У систему включені також типові для CASE засоби розробки екранних форм.

OOWin служить для підтримки об'єктно-орієнтовних технологій проектування інформаційних систем. Один зі способів використання OOWin – деталізація об'єктно-орієнтовної моделі на базі створеної ER-моделі. При перетворенні ER в OO-подання сутності й атрибути стають класами (множинами подібних об'єктів). Класи можуть бути доповнені описом послуг класу, тобто виконуваних операцій, переданих і параметрів, що повертають, подій. Інший спосіб використання OOWin - реінжиніринг, тому що модернізація проводиться на рівні існуючої моделі.

Система Design/IDEF (фірма Meta Software) призначена для концептуального проектування складних систем. З її допомогою розробляються специфікації, IDEF0- і IDEF1X-діаграми, словники даних, проводиться документування й перевіряється несуперечність проектів. Є додаткова система Design/CPN, що дозволяє проводити імітаційне моделювання на основі моделей, перетворених у кольорові мережі Петрі.

Методологія об'єктно-орієнтованого аналізу й проектування ПЗ за методикою Г.Буча з використанням мови UML реалізована в системах Rational Rose (фірма Rational Software Corporation) і Platinum Paradigm Plus (фірма Platinum Technology). В Rational Rose підтримується генерація коду по побудованих діаграмах класів, зворотне моделювання (тобто побудова UML-моделі по програмному коді на таких мовах, як C++, Java, Visual Basic, IDL CORBA), візуальне програмування. Мова UML застосовують і в ряді інших систем, наприклад, в інструментальному середовищі об'єктно-орієнтованого проектування ПЗ objecti (фірма micro TOOL), у якій автоматично генерується програмний код по графічному UML-описі.

Ряд програмних продуктів, що реалізують IDEF-моделі, розроблені фірмою KBSI, зокрема, ProSim реалізує IDEF3, SmartER - IDEF1 і IDEF1X, SmartClass - IDEF4.

Поведінкове моделювання підприємств передбачене також у деяких системах реінжинірингу, наприклад, у системі BAAN IV.

Для перетворення функціональних або поведінкових моделей в імітаційні застосовують спеціальні програми. Так, разом із програмою BPWin для одержання імітаційних моделей використовують програму BPSimulator.

Список літератури для конспекту лекцій

Основна література

1. Lee Kunwoo. Principles of CAD/CAM/CAE Systems. Lee K. *Principles of CAD/CAM/CAE Systems*. – Pearson Education, 2015. – 560 p. (Базові поняття САПР, інтеграція CAD–CAE–CAM)
2. Zeid I., Sivasubramanian R. CAD/CAM Theory and Practice. Zeid I., Sivasubramanian R. *CAD/CAM Theory and Practice*. – McGraw-Hill, 2018. – 688 p. (Методологія проектування, моделювання, автоматизація)
3. Groover M.P. Automation, Production Systems, and Computer-Integrated Manufacturing. Groover M.P. *Automation, Production Systems, and Computer-Integrated Manufacturing*. – Pearson, 2020. – 816 p. (СІМ, автоматизація, цифрове виробництво)
4. Moaveni S. Finite Element Analysis: Theory and Application with ANSYS. Moaveni S. *Finite Element Analysis: Theory and Application with ANSYS*. – Pearson, 2020. – 816 p. (CAE, метод кінцевих елементів у прикладах)

Додаткова література

5. Zienkiewicz O.C., Taylor R.L., Zhu J.Z. The Finite Element Method. Zienkiewicz O.C., Taylor R.L., Zhu J.Z. *The Finite Element Method*. – Butterworth-Heinemann, 2013–2014. (Теоретичні основи МКЕ)
6. Bathe K.J. Finite Element Procedures. Bathe K.J. *Finite Element Procedures*. – Cambridge, 2014. – 1037 p. (Поглиблене чисельне моделювання)
7. Tickoo S. AutoCAD and AutoCAD LT Essentials. Tickoo S. *AutoCAD and AutoCAD LT Essentials*. – SDC Publications, 2022. – 520 p. (Практичні навички CAD)
8. Sadovoy Oleksii, Havrysh V., Kalinichenko A., Hruban V. *A Hybrid Power Supply with Variable Speed Drive for Automatically Moving Irrigation Equipment*. IEEE MEES, 2021. (Приклад інтеграції САПР, електроприводу та енергосистем)
9. Havrysh V., Kalinichenko A., Sadovoy O. *Decentralized Power Supply Systems for Irrigation Machines with Variable Speed Drives*. Energies, 2022. (Децентралізовані системи живлення)
10. Sadovoy O., Kalinichenko A., Havrysh V. *Energy Efficiency of Electric Drives in Agricultural Water Supply Systems*. Renewable Energy, 2021. (Енергоефективність електромеханічних систем)

Навчальне видання

СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЄКТУВАННЯ
Конспект лекцій

Укладачі: **Садовий** Олексій Степанович

Формат 60×84 1/16. Ум. друк. арк. 10,1,
Тираж 20 прим. Зам. № __.

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54020, м. Миколаїв, вул. Георгія Гонгадзе, 9
Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02.2013 р.