

Лекція №11

Розділ 2. Етапи побудови математичної моделі.

Реалізація математичної моделі як програми для ЕОМ

В даний час математичні моделі, як правило, застосовують як основу для створення різних програмних комплексів, що використовуються для вирішення різноманітних дослідницьких, проектно-конструкторських та управлінських завдань. Ця обставина призводить до необхідності реалізації моделі як програми для ЕОМ. Процес розробки надійного та ефективного програмного забезпечення не менш складний, ніж розробка попередніх етапів створення математичної моделі. Успішне вирішення цього питання можливе лише при впевненому володінні сучасними алгоритмічними мовами та технологіями програмування, наявного програмного забезпечення, знання можливостей обчислювальної техніки, особливостей реалізації на ЕОМ методів обчислювальної математики, досвіду вирішення подібних завдань.

Процес створення програмного забезпечення можна розбити на ряд етапів:

- розробка технічного завдання створення програмного забезпечення;
- проектування структури програмного комплексу;
- кодування алгоритму;
- тестування та налагодження;
- супровід та експлуатація.

Технічне завдання розробку програмного забезпечення оформляють як специфікації. Орієнтовна форма специфікації включає наступні сім розділів.

1) Назва задачі

Надається коротке визначення розв'язуваної задачі, назва програмного комплексу, вказується система програмування для його реалізації та вимоги до апаратного забезпечення (комп'ютера, зовнішніх пристроїв тощо).

2) Опис

Докладно викладається математична постановка задачі, описується застосовувана математична модель для завдань обчислювального характеру, метод обробки вхідних даних для задач не обчислювального характеру і т.д.

3) Управління режимами роботи програми

Формуються основні вимоги до способу взаємодії користувача з програмою (інтерфейс "користувач-комп'ютер").

4) Вхідні дані

Описуються вхідні дані, вказуються межі, у яких можуть змінюватися, значення, які вони можуть приймати, тощо.

5) Вихідні дані

Описуються вихідні дані, вказується, в якому вигляді вони повинні бути представлені - в числовому, графічному або текстовому, наводяться відомості про точність та обсяг вихідних даних, способи їх збереження і т.д.

6) Помилки

Перераховуються можливі помилки користувача під час роботи з програмою (наприклад, помилки під час введення вхідних даних). Вказуються способи діагностики (у разі під діагностикою розуміється виявлення, виявлення помилок під час роботи програмного комплексу) та захисту від цих помилок на етапі проектування, і навіть можлива реакція користувача під час здійснення ним помилкових дій і реакція програмного комплексу (комп'ютера) ці дії.

7) Тестові задачі Наводиться один або кілька тестових прикладів, на яких у найпростіших випадках проводиться налагодження та тестування програмного комплексу.

Приклад:

Специфікація задачі про баскетболіста.

1) Назва задачі Назва програми Basketball Система програмування Delphi Комп'ютер IBM PC Pentium Операційна система Windows 10

2) Опис Наводиться математична постановка задачі та опис методу її вирішення.

3) Керування режимами роботи програми Для керування режимами роботи програми необхідно використовувати інтерфейс Windows з використанням меню, діалогових вікон, полів введення даних, кнопок.

4) Вхідні дані Вхідними даними є радіус і маса м'яча, його початкові координати та швидкість, кут кидання, координати кошика.

5) Вихідні дані Траєкторія центру м'яча, розрахункова величина дальності та точність кидка.

6) Помилки При введенні вихідних даних передбачити контроль:

- всі значення, що вводяться, повинні бути позитивними;
- кут кидання лежить у межах від 5 до 85 градусів;
- початкова швидкість м'яча лежить у межах від 0 до 30 м/с;

- горизонтальна координата центру кошика більша за початкову горизонтальну координату м'яча.

При діагностуванні перерахованих помилок програма повинна видавати відповідні повідомлення, які можуть супроводжуватися звуковим сигналом, та пропонувати повторити введення.

7) Тестові приклади

При $x_0 = y_0 = u_k = 0$; $x_k = 4,225$; $v_0 = 6,44$; $\alpha = 45^\circ$;

Отримуємо: $L = 4,225$; $\Delta = 0$.

На етапі проектування формується загальна структура програмного комплексу. Уся програма розбивається на програмні модулі. Для кожного програмного модуля формулюються вимоги щодо функцій, що реалізуються, і розробляється алгоритм, що виконує ці функції. Визначається схема взаємодії програмних модулів, яка називається схемою потоків даних програмного комплексу. Розробляється план та вихідні дані для тестування окремих модулів та програмного комплексу в цілому.

Більшість програм, що реалізують математичні моделі, складаються із трьох основних частин:

- *препроцесор* (Підготовка та перевірка вихідних даних моделі);
- *процесор* (вирішення завдання, реалізація обчислювального експерименту);
- *постпроцесор* (Відображення отриманих результатів).

Лише щодо простих випадків дані три складові можуть бути оформлені як однієї програми. При вирішенні сучасних завдань з моделювання поведінки рідин, газів і твердих тіл кожна з частин може включати цілий комплекс програм. Наприклад, постпроцесор повинен вміти представляти інформацію у табличному, а й графічному вигляді (діаграми, графіки залежно від різних параметрів, відображення скалярних, векторних (тензорних) полів тощо.). Особливо сильний розвиток можливості пре-і постпроцесора отримують у сучасних системах автоматизованого проектування (САПР), де вони значною мірою можуть скоротити час на отримання даних та оцінку результатів моделювання.

Як правило, створення сучасних математичних моделей у будь-якій галузі та доведення їх до програмних комплексів потребує значних часових витрат (мінімум 3-5 років). Потрібен час не тільки на освоєння методик і підходів до моделювання в досліджуваній галузі, але й час на напрацювання бібліотек програм з вирішення математичних завдань, що підходять, з підготовки вихідних даних і відображення одержуваних результатів. Якісні, надійні, що володіють дружнім інтерфейсом, легко модифіковані та добре супроводжувані програмні комплекси можна створювати лише за наявності добре продуманої стратегії розвитку

програмного забезпечення, що забезпечує його модульність та сумісність за вхідними та вихідними параметрами.

Велике значення слід надавати освоєнню сучасних технологій програмування: структурної, абстрактної, об'єктно-орієнтованої та візуальної. Призначення будь-якої технології - це насамперед підвищення надійності програмного забезпечення та збільшення продуктивності праці програміста. Причому, чим серйозніший і об'ємніший програмний проект, тим більшого значення набувають питання використання сучасних технологій програмування. Нехтування цими питаннями може призвести до значних тимчасових витрат та зниження надійності програмного комплексу.

Найважливішим фактором, що визначає надійність та малі терміни створення програмного комплексу для вирішення певного класу завдань, є наявність розвиненої бібліотеки сумісних між собою програмних модулів. Програма виходить більш надійною та створюється за менші терміни при максимальному використанні стандартних програмних елементів. Для ефективної розробки програмного забезпечення в галузі математичного моделювання насамперед слід звернути увагу на створення таких стандартних бібліотек програм:

- наближені та чисельні методи;
- засоби підготовки вихідних даних (препроцесори);
- засоби візуалізації та подання результатів (постпроцесори).

Розробка таких загальних бібліотек програм можлива лише за стандартизації потоків передачі між препроцесором, процесором і постпроцесором. У найпростішому випадку мова може йти про уніфікацію форматів файлів, що передаються.

Перевірка адекватності моделі

Під адекватністю математичної моделі розумітиметься ступінь відповідності результатів, отриманих за розробленою моделлю, даними експерименту або тестової задачі. Перш ніж переходити до перевірки адекватності моделі, необхідно переконатися у правильному комплексному функціонуванні всіх алгоритмів і програм моделі, виконати незалежне тестування та налагодження всіх окремих алгоритмів (наприклад, програмних модулів, що використовуються, що реалізують чисельний метод).

Перевірка адекватності моделі має дві мети:

1) Переконатися у справедливості сукупності гіпотез, сформульованих на етапах концептуальної та математичної постановок. Переходити до перевірки гіпотез слід лише після перевірки використаних методів вирішення, комплексного налагодження та усунення всіх помилок та конфліктів, пов'язаних із програмним забезпеченням.

2) Переконатися, що точність отриманих результатів відповідає точності, обумовленої в технічному завданні.

Перевірка розробленої математичної моделі виконується шляхом порівняння з наявними експериментальними даними про реальний об'єкт або з результатами інших, створених раніше моделей, що добре себе зарекомендували. У першому випадку говорять про перевірку шляхом порівняння з експериментом, у другому – про порівняння з тестовим завданням.

Вирішення питання про точність моделювання залежить від вимог, що пред'являються до моделі, та її призначення. У цьому повинна враховуватися точність отримання експериментальних результатів чи особливості постановок тестових завдань. Для моделей, призначених до виконання оціночних і прикидочних розрахунків, задовільною вважається точність 10-15%. Для моделей, призначених для використання в керуючих та контролюючих системах, необхідна точність може бути 1-2% і навіть вищою.

Як правило, розрізняють якісний та кількісний збіг результатів порівняння. При якісному порівнянні потрібно лише збіг деяких характерних рис у розподілі досліджуваних параметрів (наприклад, наявність екстремальних точок, позитивне чи негативне значення параметра, його зростання чи спадання тощо.). Фактично при якісному порівнянні оцінюється збіг лише виду функції розподілу параметрів (зменшується або зростаюча, з одним екстремумом або з декількома). Питання кількісному порівнянні можна ставити лише після задовільного відповіді питання якісної відповідності результатів. При кількісному порівнянні велике значення слід надавати точності вихідних даних для моделювання та відповідних значень порівнюваних параметрів.

Неадекватність результатів моделювання можлива принаймні з трьох причин:

а) Значення параметрів моделі, що задаються, не відповідають допустимій області цих параметрів, яка визначається прийнятою системою гіпотез. Наприклад, у задачі про баскетболіст гіпотезу про відсутність опору повітря можна використовувати лише за відносно малих (<5 м/с) швидкостях руху тіла. За більших значень початкової швидкості м'яча вплив сили опору буде суттєвим.

б) Прийнята система гіпотез правильна, але константи та параметри у використаних визначальних співвідношеннях встановлені не точно. Наприклад, у разі завдання про баскетболіст значення прискорення вільного падіння g може бути уточнено залежно від широти місцевості, де знаходиться баскетболіст.

в) Неправильна вихідна сукупність гіпотез.

Всі три випадки вимагають додаткового дослідження як об'єкта, що моделюється (з метою накопичення нової додаткової інформації про його поведінку), так і дослідження самої моделі (з метою уточнення меж її застосування).

Зауваження. У разі не аналізується вплив обраного чисельного методу на точність одержуваного рішення, отже, і адекватність моделі. Питання збіжності алгоритму і стійкості одержуваного обраним чисельним методом рішення, і навіть накопичення похибок, що з помилками округлення під час використання ЕОМ, не розглядається.

У разі виникнення проблем, пов'язаних з адекватністю моделі, її коригування слід починати з послідовного аналізу всіх можливих причин, що призвели до розбіжності результатів моделювання з результатами експерименту. Насамперед слід досліджувати модель і оцінити ступінь її адекватності при різних значеннях параметрів, що варіюються (початкових і граничних умовах, параметрів, що характеризують властивості об'єктів моделювання). Якщо модель неадекватна в області областей параметрів, що цікавить дослідника, то слід спробувати уточнити значення констант і вихідних параметрів моделі. Якщо й у разі немає позитивних результатів, то єдиною можливістю поліпшення моделі залишається зміна прийнятої системи гіпотез. Дане рішення фактично означає повернення до другого етапу процесу розробки моделі і може спричинити не тільки серйозну зміну математичної постановки завдання, а й зміну методів її вирішення (наприклад, перехід від аналітичних до чисельних), повну переробку програмного забезпечення та новий цикл перевірки моделі на адекватність. Тому рішення про зміну прийнятої системи гіпотез має бути всебічно зважено, і прийматися лише в тому випадку, коли вичерпано всі інші можливості для покращення адекватності моделі.

Слід застерегти дослідників-початківців від спроб "перестрибнути" розглянутий етап моделювання, від бажання швидше перейти до вирішення "справжнього завдання". Як показує власний досвід авторів, подібний образ дій призводить до великих часових витрат (не кажучи вже про психологічні). Особливо небезпечною є ситуація, коли при вирішенні реального завдання з використанням належним чином не перевіреної моделі виходять правдоподібні результати. Для інших умов модель може дати якісно неправильні результати, але витoki помилок розробники шукатимуть не в моделі.