МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут економіки та управління
Факультет менеджменту

Кафедра економічної кібернетики і математичного моделювання

МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ МАГІСТЕРСЬКИХ ПРОГРАМ

Опорний конспект лекцій для здобувачів вищої освіти ступеня «магістр»
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»
денної та заочної форм навчання

Миколаїв 2018

УДК 519.85
М 34

Друкується за рішенням науково-методичної комісії факультету менеджменту Миколаївського національного аграрного університету від 26.04.2018 року, протокол № 9.

Укладачі:

О. В. Шебаніна – д-р. екон. наук, професор, професор кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;

М. А. Домаскіна – канд. екон. наук, доцент, доцент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;

І. І. Хилько – старший викладач кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;

І. В. Клочан – д-р. екон. наук, доцент, доцент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;

С. І. Тищенко – канд. пед. наук, доцент, доцент кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет;

Н. С. Ручинська – канд. пед. наук, в.о. доцента кафедри економічної кібернетики і математичного моделювання, Миколаївський національний аграрний університет.

Рецензенти:

Т. І. Олешко – докт. тех. наук, професор, професор кафедри економічної кібернетики Національного авіаційного університету

Л. В. Вахоніна – канд. фіз.-мат. наук, доцент, доцент кафедри електроенергетики, електротехніки та електромеханіки Миколаївського національного аграрного університету

© Миколаївський національний аграрний університет, 2018

ЗМІСТ

Змістовий модуль 1. Математичне формулювання задач оптимального проектування (4 години) 4

Лекція 1. Вступ. Математичне забезпечення та його склад (2 години)..... 4

Лекція 2. Моделювання об’єкту і планування експерименту (2 години)..... 12

Змістовий модуль 2. Оптимізаційні задачі в енергетиці (6 годин).. 27

Лекція 3. Математична модель оптимізації (2 години)..... 27

Лекція 4. Методи розв’язання оптимізаційних задач (2 години)..... 40

Лекція 5. Багатокритеріальні задачі (2 години) 48

Змістовий модуль 3. Математичне програмне забезпечення (6 годин) 53

Лекція 6. Основи роботи в MathCAD (4 години)..... 53

Лекція 7. Основи роботи в MatLAB (2 години) 58

Змістовий модуль 1. Математичне формулювання задач оптимального проектування (4 години)

Лекція 1. Вступ. Математичне забезпечення та його склад (2 години)

1.1. Методи математичного забезпечення

1.2. Вимоги до математичного забезпечення

1.3. Етапи вирішення задач на ЕОМ

1.4. Перспективи розвитку математичного забезпечення

1.1. Методи математичного забезпечення

Математичне забезпечення (*mathematical support*) – сукупність методів, правил, математичних моделей і алгоритмів розв'язання задач;

Розрізняють загальне математичне забезпечення (для організації обчислювального процесу на даній ЕОМ) і спеціальне математичне забезпечення (для вирішення конкретних завдань).

Ступінь розвитку математичного забезпечення визначає ефективність використання певної інформаційної технології. Нині спостерігається тенденція до зростання частки витрат на розроблення математичного апарату у витратах на проект інформаційної системи.

Побудова математичної моделі задач керування покладається на фахівців з організаційно-технологічних рішень – постачальників проблемних задач керування і фахівців з формалізації процесу

прийняття управлінських рішень. Неминучі спрощення процесу, що моделюється, мають бути достатньо обґрунтовані для того, щоб уникнути зайвого спрощення процесу керування. Слід зазначити, що потреби інформатизації виробництва поки випереджають можливості прикладної математики (приміром, найчастіше використовують лінійні моделі, проте майже всі залежності в економіці й управлінні підприємством – нелінійні, тому це призводить до значного спрощення моделі).

Останнє десятиліття характеризується значним розвитком математичних дисциплін, методи яких використовуються для вирішення задач в інформаційних системах.

Мережеві методи найширше застосовуються у проектуванні. Вони дають змогу визначати параметри мережевих моделей та аналізувати хід робіт з реалізації виробничих планів. У рамках мережевого моделювання можлива одно- чи багато-критеріальна оптимізація, у тому числі за часом і ресурсами.

Евристичні методи дають можливість вирішувати слабко структуровані задачі, які неможливо розв’язати повним перебором варіантів, приміром задачі календарного планування. Сутність евристичного методу полягає в тому, щоб запланувати роботи у найкоротші терміни, але так, щоб не перевищити заданий верхній рівень ресурсів. Як правило, використання евристичних методів передбачає наявність діалогу з користувачем, під час якого на комп’ютер покладаються обчислення і видача проміжних результатів, включаючи різні графіки і діаграми. Користувач

залежно від отриманих даних визначає подальший напрямок розрахунків.

Методи комбінаторики, математичної логіки, інформаційної алгебри використовуються для розв'язання інформаційно-логічних задач. Це групування та впорядкування даних, об'єднання масивів даних і коригування інформації, введення, декомпозиція й обмін даними між електронними сховищами у межах однієї або кількох ЕОМ.

Математичне програмування поєднує лінійне, нелінійне, динамічне і стохастичне програмування. Особливо вирізняються транспортні задачі, що розв'язуються із застосуванням методів лінійного програмування. З використанням лінійного програмування вирішуються й аналізуються такі питання, як розроблення та складання прогнозів планів розвитку галузей, оптимального розподілу ресурсів.

Нелінійне математичне програмування застосовується рідше за лінійне, причому найчастіше нелінійні задачі розв'язуються також способами лінійного програмування, для чого криволінійні залежності апроксимуються прямими (лінеаризація).

Типовими задачами *динамічного програмування* є розподіл капітальних вкладень, календарне планування, пошук оптимальної послідовності постачання товарів, управління запасами. Суть динамічного програмування полягає у тому, що з двох шляхів досягнення результату довший шлях відкидається, щоб зменшити обсяг обчислень на ЕОМ.

Стохастичне програмування характеризується введенням у задачі ймовірнісних значень параметрів, що відображають ризик і невизначеність.

Методи теорії ігор дають змогу формалізувати та розв'язувати задачі, що зазвичай розв'язуються емпірично, без використання кількісних вимірників. До таких задач належить, приміром, дослідження конфліктних ситуацій в умовах невизначеності інформації про дії учасників. Методи теорії ігор широко застосовуються при аналізі організаційних, економічних, військових і політичних ситуацій.

Теорія черг або масового обслуговування вивчає імовірнісні моделі поведінки систем. Базою для вирішення задач масового обслуговування є теорія ймовірностей.

Математична статистика, один з розділів теорії ймовірностей, дозволяє дати оцінку певній сукупності даних.

Метод статистичних іспитів також призначений для вивчення імовірнісних систем і застосовується при моделюванні найрізноманітніших ситуацій. Цим методом вдається, зокрема, одержати характеристики системи без проведення натурних експериментів.

Метод теорії розкладів дає змогу знайти оптимальну послідовність побудови об'єктів за якимось критерієм. Приміром, критерієм може бути «найменший термін будівництва», «мінімум простоїв виконавців на об'єктах», «максимальна щільність робіт на об'єктах» тощо.

Методи теорії множин дають можливість значно компактніше описувати задачі керування, знаходити ефективні шляхи їхнього розв'язання.

1.2. Вимоги до математичного забезпечення

Універсальність МЗ визначає можливість його застосування до широкого класу проектованих об'єктів. Це особливо важливо при створенні комплексних САПР, що включають різні види завдань: від конструювання виробу і проектування технологічних процесів до вибору ріжучого інструменту і проектування конструкцій спеціального оснащення на основі аналізу типових технологічних рішень. Універсальність МЗ спрощує методику автоматизованого проектування. Універсальність не має кількісної оцінки.

Алгоритмічна надійність – властивість компоненту МЗ давати при його застосуванні правильні результати. Кількісною оцінкою алгоритмічної надійності служить ймовірність отримання правильних результатів при дотриманні обумовлених обмежень на застосування методу. Метод алгоритмічно надійний, якщо ця ймовірність рівна одиниці або близька до неї.

З алгоритмічною надійністю тісно пов'язана проблема обумовленості математичних моделей і завдань. При поганій обумовленості малі похибки початкових даних приводять до великих похибок результатів. Через це знижується точність результатів проектування і зростають витрати машинного часу. Для аналізу і оптимізації об'єктів з погано обумовленими моделями

необхідно застосовувати спеціальні методи з підвищеною алгоритмічною надійністю.

Точність є найбільш важливою властивістю всіх компонентів МЗ і визначає ступінь збігу розрахункових і дійсних результатів. Алгоритмічно надійні методи можуть давати різну точність: якщо точність виявляється нижчою гранично допустимих значень, а також якщо рішення неможливе, говорять не про точність, а про алгоритмічну надійність. В більшості випадків рішення проектних задач характеризується сумісним використанням багатьох компонентів МЗ, що утрудняє оцінку впливу похибок окремих компонентів. При необхідності оцінки їх точності проводять обчислювальні експерименти з використанням тестових завдань.

Витрати машинного часу багато в чому визначаються складністю проектованих об’єктів і розмірністю вирішуваних задач. Машинний час обчислювального процесу є головним обмежуючим чинником при спробах підвищити складність проектованих на ЕОМ об’єктів. Одним з шляхів скорочення термінів проектування є застосування в САПР багатопроцесорних обчислювальних систем, що забезпечують паралельне проведення обчислень. У зв’язку з цим найважливішим показником економічності МЗ є його пристосованість до паралельного процесу проектування.

Використана пам’ять є другим після витрат машинного часу показником економічності МЗ. Витрати пам’яті визначаються довжиною програми і об’ємом масивів даних, що використовуються. Не дивлячись на значне збільшення оперативної пам’яті в сучасних

ЕОМ, вимоги до зниження витрат пам'яті залишаються актуальними. Це пов'язано з тим, що в мультипрограмному режимі функціонування ЕОМ завдання із запитом більшого об'єму пам'яті отримує нижчий пріоритет, і в результаті час її перебування в системі збільшується і продуктивність процесу проектування знижується.

1.3. Етапи вирішення задач на ЕОМ

1. Постановка (математичне формулювання) завдання, розробка математичної моделі.
2. Вибір методу чисельного вирішення задачі.
3. Розробка алгоритму вирішення і структури даних.
4. Складання програми – реалізація алгоритму на вхідній мові.
5. Введення програми в пам'ять ЕОМ.
6. Відлагодження програми на контрольному прикладі.
7. Підготовка і запис початкових даних.
8. Вирішення задачі на ЕОМ, аналіз і оформлення результатів.

Перші чотири етапи в основному визначають не тільки ефективність програм, що розробляються, але і трудомісткість процесу їх розробки.

1.4. Перспективи розвитку математичного забезпечення

Підвищення ефективності методів оптимізації систем автоматичного проектування може бути досягнуте за рахунок:

- єдності фізичних і математичних принципів, що використовуються для розробки моделей;

- універсальності моделей – можливості опису різних класів пристроїв;
- блоковості, що забезпечує отримання моделей складних об’єктів компоновкою простих моделей і алгоритмів;
- адаптації моделей до умов проектування, що змінюються;
- можливості повної або часткової формалізації процесу побудови математичних моделей проектованих пристроїв.

Питання для самоперевірки:

1. Дайте визначення поняття «математичне забезпечення».
2. Які існують методи математичного забезпечення?
3. Сформулюйте вимоги до математичного забезпечення.
4. Які є основні етапи розв’язання задач на ЕОМ?

Література:

1. Математичне забезпечення магістерських програм: навчально-методичний посібник / Уклад.: О. М. Нещадим. – К.: НУБіПУ, 2014. – 134 с.

Лекція 2. Моделювання об’єкту і планування експерименту (2 години)

2.1. Планування експерименту для отримання математичної моделі технологічного процесу

2.2. Складання плану факторного експерименту

2.3. Рандомізація дослідів, розрахунок похибок вимірювань

2.4. Загальні схеми планування дослідів для чотирьох або трьох факторів

2.5. Критерії оптимізації

2.1. Планування експерименту для отримання математичної моделі технологічного процесу

З середини XX століття в різних галузях людської діяльності стали широко застосовувати математичні методи і електронно-обчислювальну техніку. Виникли такі нові дисципліни, як «математична економіка», «математична лінгвістика», «математична біологія» і т.п., які вивчають математичні моделі відповідних об’єктів і явищ, а також методи дослідження цих моделей.

Математична модель – це приблизний опис будь-якого класу явищ або об’єктів реального світу на мові математики. Основна мета моделювання – досліджувати ці об’єкти і передбачити результати майбутніх спостережень. Проте моделювання це ще і метод пізнання навколишнього світу, що дає можливість управляти ним.

Математичне моделювання і пов'язаний з ним комп'ютерний експеримент незамінні в тих випадках, коли натурний експеримент неможливий або утруднений з тих або інших причин. Наприклад, неможливо перевірити правильність тієї або іншої космологічної теорії. В принципі, можливо, але навряд чи розумно поставити експеримент із розповсюдження якої-небудь хвороби, наприклад чуми, або здійснити ядерний вибух, щоб вивчити його наслідки. Проте все це цілком можна зробити на комп'ютері, побудувавши заздалегідь математичні моделі явищ, що вивчаються.

У моделі відображаються тільки ті фактори і параметри оригінального об'єкту, які мають істотне значення для вирішення досліджуваної проблеми. Крім того, вимірювання істотних факторів і параметрів практично завжди містять помилки, що викликаються неточністю вимірювальних приладів і незнанням деяких факторів. Через це математична модель є тільки наближеним описом властивостей об'єкту, що вивчається. А математичну модель можна визначити ще і як **абстракцію** реальної **суті,** що вивчається.

Єдиним способом перевірки ступеня відповідності моделі і модельованого об'єкта є науковий експеримент. Під **експериментом** розуміють цілеспрямовано організований дослід, що складається з відтворення і спостереження досліджуваного явища з необхідною точністю. Математичне моделювання також є експериментом, мета якого полягає в тому, щоб дослідним шляхом, або шляхом логічного висновку, знайти невідомі параметри θ_1[1], …,

[1] Тета

θ_m, математичної моделі процесу, що вивчається, або об'єкта і побудувати співвідношення $\mathbf{Y^* = F^* (x_1, \ldots, x_n, z_1, \ldots, z_r, \theta_1, \ldots, \theta_m)}$ у явному вигляді так, щоб в області варіювання факторів $\mathbf{x_1, \ldots, x_n, z_1, \ldots, z_r}$, функція $\mathbf{Y^*}$ найкращим чином наближалася б до функції **Y**.

Досягнення цієї мети засноване на використанні **принципу максимальної правдоподібності математичної моделі (функції $\mathbf{F^*}$)** за експериментально отриманим даним про досліджуваний об'єкт.

Дуже часто, приступаючи до розв'язання науково-технічних задач, енергетик не має вичерпних відомостей про механізм досліджуваного процесу, про його властивості. Він може тільки назвати параметри, що визначають умови протікання процесу, і вимоги до його результатів. У цих умовах доцільним є використання так званого кібернетичного підходу до вирішення цієї проблеми, в основі якого лежить запропонована Н. Вінером ідея «чорного ящика». «Чорний ящик» − це об'єкт дослідження (Рис. 1).

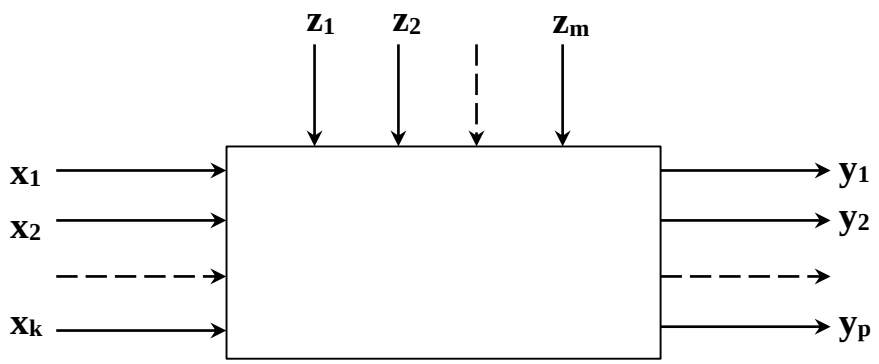


Рис. 1. «Чорний ящик» Віннера

Стрілками, що входять в об'єкт, показані вхідні параметри, які можуть бути керованими (**x**) і некерованими (**z**). Робота об'єкта може

характеризуватися декількома вихідними параметрами, які на схемі позначені стрілками, що виходять з прямокутника.

Необхідно навчитися керувати об’єктом, інформація про елементарні операції усередині якого надзвичайно мала, що по суті аналогічно до розв’язання поставлених задач. Зазвичай вхідні величини називають **факторами**, вихідні – **відгуком**, а залежність, яку намагаються встановити, – **функцією відгуку**.

Об’єкт моделювання (предмет) – це обмежена галузь реальної дійсності або галузь ідеальних уявлень, що підлягає опису (моделюванню) і дослідженню. Предметна галузь складається з об’єктів, що розрізняються за якими-небудь ознаками (властивостям) і знаходяться в певних відносинах між собою, або що взаємодіють яким-небудь чином. Тобто кожну галузь дослідження можна уявити як щось ціле, причому ця досліджувана галузь реально існує, і має свої особливості та властивості. Вивчення цих особливостей та властивостей дозволяє досліднику однозначно ідентифікувати об’єкт дослідження.

Параметр оптимізації. При плануванні експерименту важливо правильно обрати параметр оптимізації. Рух до оптимуму можливий, якщо обраний один параметр оптимізації, а інші виступають як обмеження. Можлива побудова узагальненого параметра як функції від безлічі початкових параметрів. Параметр оптимізації повинен бути кількісним, доступним для вимірювання і повинен виражатися одним числом. Якщо вимірювання параметра неможливе, то користуються ранговою оцінкою.

Ранг – це оцінка параметра оптимізації за наперед обраною шкалою: двобальною, п’ятибальною, десятибальною і т.п. Ранговий параметр має обмежену дискретну область визначення. У простому випадку область містить два значення: так – ні; добре – погано; брак – придатні деталі і т.п. За інших рівних умов перевагу необхідно віддавати кількісному вимірюванню, оскільки рангова оцінка носить суб’єктивний характер.

Параметр оптимізації повинен бути однозначним у статистичному сенсі, тобто заданому поєднанню рівнів факторів повинне відповідати одне (з точністю до помилки експерименту) значення параметра оптимізації; ефективним в статистичному сенсі, тобто визначатися з найбільшою точністю, що дозволяє скоротити до мінімуму число паралельних експериментів; існувати для всіх станів досліджуваного об’єкта; мати фізичний сенс.

Фактори. Фактором називають незалежну змінну величину, що впливає на параметр оптимізації. Кожен фактор має область визначення – сукупність усіх значень, яких може набувати фактор.

При дослідженні процесу необхідно враховувати всі істотні фактори. Якщо з яких-небудь причин вплив деяких факторів неможливо врахувати в експерименті, то ці фактори повинні бути стабілізовані на певних рівнях протягом усього експерименту. *Рівнями* називають значення факторів в експерименті. Якщо число факторів велике, то необхідно відсіяти ті фактори, які несуттєво впливають на параметр оптимізації. Відсівання неістотних факторів

проводять на основі апріорного ранжирування або за допомогою постановки відсіваючих експериментів.

Проте в моделі відображаються тільки ті фактори і параметри оригінального об’єкта, які мають істотне значення для вирішення досліджуваної проблеми. Крім того, вимірювання істотних факторів і параметрів практично завжди містять помилки, що викликаються неточністю вимірювальних приладів і незнанням деяких факторів. Через це математична модель є тільки приблизним описом властивостей об’єкта, що вивчається, і головним питанням математичного моделювання є питання про те, як точно складена математична модель відображає відносини між факторами, що враховуються, параметрами і показником оцінюваної властивості реального об’єкта.

Планування експерименту полягає в процедурі вибору числа і умов проведення дослідів, необхідних і достатніх для дослідження об’єкту із заданою точністю. Планування експерименту забезпечує:

– одночасне варіювання всіх факторів по спеціальних правилах;

– використання математичного апарату, що формалізує багато дій експериментатора;

– вибір чіткої стратегії, що дозволяє ухвалювати обґрунтовані рішення після кожної серії експериментів;

– мінімізацію числа дослідів, ресурсів (фінансових, тимчасових, матеріальних, людських).

Сукупність рівнів вхідних факторів об'єкта (по одному рівню від фактора) визначає один стан об'єкта. Якщо число рівнів всіх вхідних факторів однаково, то число всіх станів об'єкта (N_c) можна визначити за формулою

$$N_c = p^k$$

де k – загальна кількість вхідних факторів; p – число рівнів кожного фактора.

Мета планування експеримента – визначити кількість факторів і їх рівнів для отримання необхідної і достатньої інформації про об'єкт дослідження.

При проведенні дослідів дуже важливо, чи може дослідник під час дослідів встановлювати ті рівні факторів, які представляють для нього інтерес. З цієї точки зору розрізняють наступні фактори:

– контрольовані і керовані – це фактори, для яких можна не тільки зареєструвати їх рівень, а й задати в кожному досліді будь-яке можливе значення;

– контрольовані, але не керовані – це фактори, рівні яких можна тільки реєструвати, але задавати в кожному дослідженні визначене значення практично неможливо;

– неконтрольовані – це фактори, рівні яких не реєструються дослідником, він навіть може не підозрювати про їх існування.

Якщо дослідник має можливість контролювати і управляти рівнями факторів, то такий експеримент можна назвати активним. Якщо дослідник може тільки спостерігати і реєструвати, але не має

можливості управляти рівнями факторів, то це пасивний експеримент.

План експеримента – сукупність даних, що визначають число, умови і порядок реалізації дослідів.

Раніше було сказано, що кожен фактор має свою область визначення. Сукупність областей визначення вхідних факторів назвемо факторним простором. Кожна точка факторного простору являє собою цілком певне поєднання конкретних значень вхідних факторів і відповідає одному стану об’єкта. На основі аналізу завдання дослідження і апріорної інформації про досліджуваний об’єкт для кожного вхідного фактора виділяється область, в межах якої він буде змінюватися під час експерименту. Поєднання таких областей по всім вхідним факторам будемо називати факторним простором експерименту.

Планування експерименту починають з вибору нульового рівня кожного вхідного фактора, в якості якого може бути взята будь-яка точка факторного простору експерименту. Але однієї точки нульового рівня для проведення експерименту і отримання необхідної інформації недостатньо. Потрібні ще точки. Побудова плану експеримента – це вибір точок (рівнів вхідних факторів) відносно нульового. Для визначення інших рівнів вхідних факторів вводиться інтервал варіювання кожного вхідного фактора. Щоб позначити верхній рівень вхідного фактора, слід інтервал варіювання додати до нульового рівня даного фактора, а щоб

визначити нижній рівень – відняти інтервал варіювання від нульового рівня.

На інтервал варіювання накладаються обмеження природного характеру знизу і зверху. До інтервалу варіювання вхідного фактора пред'являються наступні вимоги:

– він не може бути менше помилки, з якої вимірюється даний фактор, інакше рівні фактора будуть невиразні;

– він не може бути занадто великим, тобто нижні і верхні рівні не повинні покидати області визначення фактора та області проведення експерименту.

Зазвичай при первинному плануванні експерименту кількість рівнів по всіх вхідних факторах вибирають однакову. Тоді кількість дослідів в експерименті (N_e) може бути визначена по формулі

$$N_e = p_e^{k_e}$$

де p_e – число рівнів кожного вхідного фактора; k_e – число вхідних факторів, досліджуваних в експерименті.

Якщо з аналізу апріорної інформації відомо, що досліджувана залежність $Y_j = f(X_1, X_2 . X_k)$ є лінійною, то досить реалізувати експеримент, в якому кожен вхідний фактор має в експерименті тільки два рівні, тобто

$$N_e = 2^{k_e}$$

Такий план експерименту називається **планом першого порядку**.

Якщо з аналізу апріорної інформації відомо, що досліджувана залежність $Y_j = f(X_1, X_2 . X_k)$ є нелінійною, то досить реалізувати

експеримент, в якому кожен вхідний фактор має три рівні. Такий план називається **планом другого порядку**, а

$$N_e = 3^{k_e}.$$

2.2. Складання плану факторного експерименту

Повний факторний експеримент (ПФЕ) – це експеримент, в якому реалізуються всі можливі поєднання всіх рівнів всіх вхідних факторів (наприклад, $N_e = 2^{ke}$, $N_e = 3^{ke}$).

Умови повного факторного експерименту записують у вигляді таблиці – *матриці планування* експерименту. Для експерименту, що досліджує об'єкт з двома вхідними факторами, кожен з яких змінюється по двох рівнях, матриця планування має наступний вигляд (***Таблиця 1***).

Таблиця 1

Матриця планування[2] *повного факторного експерименту* $N_e = 2^2$

№ п/п	X_1	X_2	$Y_{експ}$
1	–	–	$Y_{1експ}$
2	–	+	$Y_{2експ}$
3	+	–	$Y_{3експ}$
4	+	+	$Y_{4експ}$

Так виглядає кодована форма запису.

Якщо для дослідження вхідного фактора було вибрано три рівні, включаючи нульовий, то в матриці планування вони позначаються знаками « – » (нижний), «0» (нульовий), «+» (верхній).

[2] Знаком «+» позначені верхні рівні факторів , знаком «– » – нижні.

На Рис. 2 показана геометрична інтерпретація повного факторного експерименту $N = 2^2$. Якщо вибрано два нижні і два верхні рівні, то вони позначаються як «–2» (другий нижній), «–1» (перший нижній), «+1» (перший верхній) і «+2» (другий верхній).

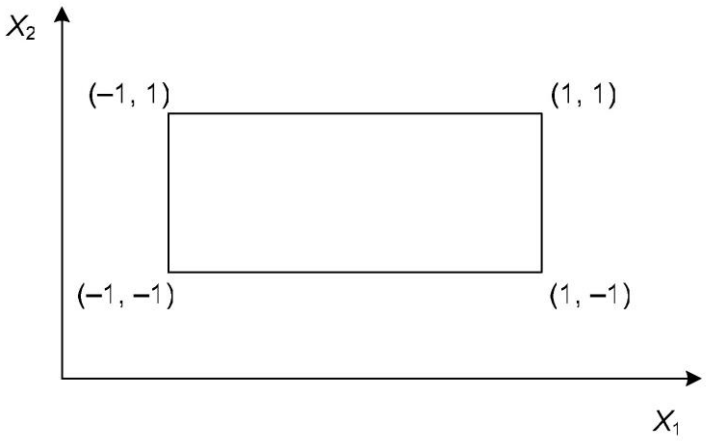


Рис. 2. Геометрична інтерпретація повного факторного експеримента $N = 2^2$

Незалежно від числа факторів матриці ПФЕ володіють наступними загальними властивостями:

– *симетричність щодо центру експерименту* – алгебраїчна сума елементів вектора-стовпця кожного фактора рівна нулю;

– *умова нормування* – сума квадратів елементів кожного стовпця рівна числу дослідів;

– *ортогональність* – сума почленних виразів будь-яких двох вектор-стовпців матриці рівна нулю;

– *рототабельність* – точки в матриці планування підбираються так, що точність прогнозів значень параметра оптимізації однакова на рівних відстанях від центру експерименту і не залежить від напряму.

Повний факторний експеримент страждає надмірністю дослідів. Якщо аналіз апріорної інформації дає підстави вважати, що у вибраній області експерименту об'єкт описується лінійною моделлю, то кількість дослідів можна мінімізувати, скоротивши матрицю планування експерименту. Такий експеримент називається *дробовим факторним експериментом* (ДФЕ), а таблиця його плану – *дробовою реплікою*. Зменшення числа дослідів дозволяє понизити витрати часу, засобів, матеріалів на проведення і обробку експерименту.

2.3. Рандомізація дослідів, розрахунок похибок вимірювань

Щоб врахувати в експерименті систематичну дію змінних, які не піддаються, або піддаються з труднощами, обліку і контролю, матрицю плану піддають рандомізації.

Випадковий порядок реалізації дослідів, тобто випадковий порядок реалізації рядків матриці плану, представляє принцип рандомізації, як один з варіантів рандомізації можна використовувати таблицю випадкових чисел.

Мета рандомізації – виключення появи і впливу систематичних помилок на результати експерименту.

Для генерування випадкової послідовності дослідів можна використовувати програми-генератори випадкових чисел. Вибрану випадковим чином послідовність дослідів порушувати не рекомендується.

2.4. Загальні схеми планування дослідів для чотирьох або трьох факторів

При **k = 2** побудова матриць повного експерименту фактора не викликає утруднень, оскільки всі можливі поєднання рівнів факторів легко знайти простим перебором значень. При збільшенні числа факторів кількість можливих поєднань рівнів швидко зростає, тому виникає необхідність у деяких прийомах побудови матриць. Розглянемо два найбільш простих прийоми.

Перший прийом побудови матриць заснований на правилі чергування знаків. У першому стовпці ($\mathbf{x_1}$) знаки чергуються по черзі, у другому вони чергуються через 2, у третьому – через 4, у четвертому – через 8, у п'ятому – через 16 і т. п. по ступенях двійки.

Другий прийом заснований на послідовному добудовуванні матриці. Для цього при додаванні нового фактора необхідно повторити комбінації рівнів початкового плану спочатку при значенні нового чинника на верхньому рівні, а потім на нижньому. Побудову матриць планування експериментів можна здійснити з допомогою комп'ютерних програм.

2.5. Критерії оптимізації

Сучасне виробництво вимагає широкого використання методів оптимізації (optimization method). Розмірність виробничих оптимізаційних завдань, як правило, велика, тому вирішувати їх доцільно на ЕОМ. Одна і та ж задача може бути вирішена різними методами одночасно, в той час, коли різні задачі можуть

вирішуватися одним методом. У будь-якому випадку повинні послужити оптимальні (як вигідно) показники виробничого процесу.

Всі методи оптимізації умовно можна розділити на класичні та сучасні. До класичних методів відносять: методи диференційного числення; чисельні методи; методи умовної і безумовної оптимізації; методи перебору варіантів. До сучасних методів можна віднести: лінійне програмування; нелінійне програмування; динамічне програмування; стохастичне програмування; теорія масового обслуговування; мережеве планування; теорія ігор, теорія планування експерименту.

При вирішенні оптимізаційних завдань спочатку треба встановити математичну модель (математичну залежність) виду досліджуваного процесу. При цьому вона повинна адекватно відображати його властивості. Надалі математичну залежність, яку необхідно дослідити на оптимальність, будемо називати цільовою функцією. Якщо в показник цільової функції вкладений фізичний зміст, то його разом з цільовою функцією називають критерієм оптимізації (критерій ефективності). Одним з найважливіших питань оптимізаційного моделювання виробничих процесів є вибір критерію оптимізації та опис його цільової функції.

У завданнях оптимізації вихідний фактор часто називають критерієм оптимізації.

Розрізняють часткові і загальні критерії оптимізації. Загальні критерії, як правило, характеризують народногосподарський ефект підприємства (прибуток, собівартість, рентабельність і т.д.).

Часткові критерії – окремі сторони виробничого процесу (витрати праці, зарплата і т.д.).

Виділяють наступні види критеріїв оптимізації:

– економічні критерії оптимізації: прибуток, собівартість, витрати;

– технічний (або технологічний) – продуктивність;

– техніко-економічні: безвідмовність, відновлюваність і т.д.;

– інші критерії оптимізації: екологічні, ергономічні, естетичні і т.д..

Питання для самоперевірки:

1. Що таке математична модель?
2. Що таке повний факторній експеримент?
3. Навіщо застосовують рандомізацію?
4. Які є види критеріїв оптимальності?

Література:

1. Гавриш П.А., Васильєва Л.В. Математичне моделювання систем і процесів : навч. посіб.. Краматорськ : ДДМА, 2007. 100 с.

2. Зедгинидзе И.Г. Планирование эксперимента для исследования многокомпонентных систем. Москва : Наука, 1976. 296 с.

3. Красовский Г.И., Филаретов Г.Ф. Планирование эксперимента. Минск: Изд-во БГУ, 1982. 302 с.

Змістовий модуль 2. Оптимізаційні задачі в енергетиці (6 годин)

Лекція 3. Математична модель оптимізації (2 години)

3.1. Постановка задачі оптимізації

3.2. Форми запису задач лінійного програмування

3.3. Геометрична інтерпретація задачі лінійного програмування

3.1. Постановка задачі оптимізації

Однією з найважливіших задач оптимізації є *задача математичного програмування*, що полягає в пошуку екстремуму (мінімуму або максимуму) функції $f(\boldsymbol{x})$ при умовах $g_i(\boldsymbol{x})\leq 0$, $i=1,...,m$, $\boldsymbol{x}\in X\subset E^n$.

Якщо функції $f(\boldsymbol{x})$, $g_i(\boldsymbol{x})$, $i=1,...,m$, – лінійні, а область X задається обмеженнями вигляду $l_j \leq x_j \leq r_j$, $j=1,...,n$, то вказана задача називається *задачею лінійного програмування* (*ЗЛП*). Уперше постановка *ЗЛП* та один із методів її розв'язання були запропоновані **Л. В. Канторовичем** у роботі «Математические методы организации и планирования производства» у 1939 році. У 1947 році **Дж. Данціг** розробив симплексний метод (симплекс-метод) – один із основних методів розв'язування ЗЛП. З тих пір теорія лінійного програмування бурхливо розвивалася і нині носить цілісний, в основному, закінчений характер.

Зауважимо, що на розвиток теорії лінійного програмування суттєво впливало її застосування до розв'язування (з широким використанням ЕОМ) прикладних задач, пов'язаних з оптимальним

плануванням, організацією та управлінням у різноманітних сферах людської діяльності.

Вивчення задач оптимізації розпочнемо з розгляду типових, що стали класичними, прикладів ЗЛП.

Задача про перевезення (транспортна задача)

У пунктах P_i ($i=1,...,m$) виробляється деякий однорідний продукт, причому в пункті P_i виробляється a_i одиниць цього продукту. У пункті Q_j ($j=1,...,n$) споживається b_j одиниць цього ж продукту. Припускаємо, що

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

Нехай c_{ij} – собівартість перевезення одиниці продукту з пункту виробництва P_i у пункт споживання Q_j (транспортні витрати). Необхідно знайти план перевезень продукту таким чином, щоб задовольнити потреби всіх споживачів, мінімізуючи при цьому загальні транспортні витрати.

Нехай x_{ij} – невідома кількість продукту, що планується для перевезення з P_i в Q_j. Тоді транспортна задача, яку прийнято називати *транспортною задачею лінійного програмування* (*ТЗЛП*), полягає у знаходженні матриці перевезень $\boldsymbol{X}=\|x_{ij}\|$, $i=1,...,m$, $j=1,...,n$, що мінімізує загальні транспортні витрати

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

при обмеженнях

$$\sum_{j=1}^{n} x_{ij} = a_i, i = 1, \dots, m$$

$$\sum_{i=1}^{m} x_{ij} = b_j, j = 1, \dots, n$$

$$x_{ij} \geq 0,\ i=1,\dots,m,\ j=1,\dots,n$$

Задача про харчовий раціон (задача про дієту)

Харчовий раціон може складатися з продуктів $P_1,\dots,P_n$ (хліб, масло, молоко і т. ін.). Відомо, що в одиниці продукту P_j міститься a_{ij}, $i=1,\dots m$, одиниць поживної речовини V_i (білки, жири, вуглеводи і т. ін.) Нехай c_j — вартість одиниці продукту P_j. Припускається, що харчовий раціон повинен містити не менше b_i одиниць речовини V_i. При вказаних обмеженнях потрібно знайти раціон найменшої вартості.

Формалізуємо цю задачу. Нехай x_j — кількість продукту P_j у шуканому раціоні. Тоді вектор $\boldsymbol{x}=(x_1,\dots,x_n)$ визначає деякий раціон вартістю

$$\sum_{j=1}^{n} c_j x_j$$

Кількість поживної речовини V_i у раціоні $\boldsymbol{x}$ дорівнює

$$\sum_{j=1}^{n} a_{ij} x_j$$

Таким чином, задача про харчовий раціон зводиться до мінімізації функції

$$\sum_{j=1}^{n} c_j x_j$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \, , i = 1, \dots , m$$

$$x_j \geq 0, j = 1,\dots,n.$$

Задача розподілу ресурсів

Деяке підприємство може реалізувати n виробничо-технологічних процесів $P_1,\dots,P_n$, використовуючи для цього ресурси $R_1,\dots,R_m$.

Відомо, що

а) підприємство має b i одиниць ресурсу R_i , $i=1,\dots,m$;

б) витрати ресурсу R_i на одиницю продукції, що виготовляється за технологією P_j , дорівнюють a_{ij} ;

в) реалізація одиниці продукції, виготовленої за технологією P_j, приносить підприємству прибуток c_j.

Задача полягає у визначенні об’єму виробництва x_j продукції за технологією P_j , $j=1,\dots,n$, що максимізує за цих умов прибуток підприємства.

Легко бачити, що сформульована задача формально зводиться до знаходження вектора $\boldsymbol{x}=(x_1,\dots,x_n)$, що максимізує функцію

$$\sum_{j=1}^{n} c_j x_j$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i\,, i = 1, \dots, m$$

$$x_j \geq 0, j = 1,\dots,n.$$

Незважаючи на змістовні відмінності, усі приведені приклади мають багато спільного. Це спільне полягає в лінійності функцій, що підлягають мінімізації (максимізації) та функцій, що входять в обмеження (рівності або нерівності).

Загальна задача лінійного програмування (*ЗЗЛП*) формулюється таким чином.

Знайти вектор $\mathbf{x}=(x_1,\dots,x_n)\in E^n$, *що мінімізує (максимізує) лінійну функцію*

$$L(x) = \sum_{j=1}^{n} c_j x_j \tag{3.1}$$

і задовольняє систему лінійних обмежень

$$\sum_{j=1}^{n} a_{ij}x_j\ R_i\ b_i\,, i = 1, \dots, m \tag{3.2}$$

$$x_j \geq 0, j = 1,\dots,k,\ k \leq n. \tag{3.3}$$

де символ R_i *замінює один із знаків* $\leq, =, \geq$.

Обмеження (3.3) називаються *прямими* і, як правило, не включаються до *непрямих* обмежень (3.2).

Вектор $\boldsymbol{x}$, що задовольняє обмеження (3.2), (3.3), називається *допустимим розв'язком* (*вектором, точкою, планом*) *ЗЛП*.

Множина допустимих розв'язків *ЗЛП* називається *допустимою областю* (*множиною*) *ЗЛП*, позначається буквою D і записується також у вигляді

$$D = \left\{ \boldsymbol{x} = (x_1, \ldots, x_n) \in E^n : \sum_{j=1}^{n} a_{ij} x_j \; R_i \; b_i \, , i = 1, \ldots, m, x_j \geq 0, j = 1, \ldots, k, k \leq n \right\}$$

Функція $L(\boldsymbol{x})$ називається *цільовою*, $\boldsymbol{x}^* \in D$, що доставляє цільовій функції мінімум (максимум), називається *оптимальним розв'язком ЗЛП* і часто записується у вигляді

$\boldsymbol{x}^* = \arg\min\,(\max) L(\boldsymbol{x})$.

$\boldsymbol{x} \in D$

Величина $L(\boldsymbol{x}^*)$ називається *оптимальним значенням цільової функції.*

Зауважимо також, що той факт, що цільова функція $L(\boldsymbol{x})$ має бути мінімізована (максимізована), часто записується у вигляді

$$L(x) = \sum_{j=1}^{n} c_j x_j \to \min(\max)$$

3.2. Форми запису задач лінійного програмування

Задача лінійного програмування в стандартній формі має вигляд:

$$f(x) = (c, x) \to \max(min)$$
$$\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

При чому, якщо задача лінійного програмування на максимум, то це перша стандартна форма:

$$f(x) = (c, x) \to max \qquad – \qquad \textit{I-а стандартна форма}$$
$$\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

Якщо задача лінійного програмування на мінімум, то це друга стандартна форма:

$$f(x) = (c, x) \to min - \quad \textit{II-а стандартна форма}$$
$$\begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

Розглянемо канонічну форму ЗЛП. Цільова функція задається скалярним добутком $f(x) = (c, x) \to max$, тобто розглядається задача максимум і обмеження задані рівняннями:

$$f(x) = (c, x) \to max \quad - \quad \textit{канонічна форма ЗЛП}$$
$$\begin{cases} Ax = b \\ x \geq 0 \end{cases}$$

Якщо задано розв'язати ЗЛП стандартної форми симплекс-методом, то потрібно спочатку привести її у канонічну форму. Симплекс-метод також вимагає додатності вектора b: $b \geq 0$.

3.3. Геометрична інтерпретація задачі лінійного програмування

Алгоритм рішення ЗЛП графічним методом:

1. Побудувати допустиму область D. Якщо нерівності з системи обмежень подати як рівняння, то їх графіками будуть прямі на координатній площині. Побудувати ці прямі можна, знайшовши хоча б по дві точки, які належать відповідно цим прямим. Якщо обмеження несуперечливі, якщо область вдалось D побудувати, то можна розв'язувати задачу.

2. Знайти і побудувати вектор нормалі n. Для лінійної функції вектор нормалі являє собою вектор коефіцієнтів цільової функції.

3. Побудувати опорну пряму. Опорна пряма це лінія рівня. Зазвичай її будують таким чином, щоб вона була перпендикулярна вектору нормалі і проходила через середину допустимої області.

4. Якщо задача на *max*, то переміщуємо опорну пряму у напрямку вектора нормалі (якщо задача на *min*, то переміщуємо опорну пряму у напрямку, протилежному вектору нормалі) доти, доки допустима область не залишиться по один бік від опорної прямої. Отримана точка (або множина точок) дотику опорної прямої до допустимої області і буде оптимальним розв'язком x^*.

5. Підставити координати x^* у цільову функцію і обчислити максимальне (мінімальне) значення.

Приклад 1. Розглянемо задачу лінійного програмування. Задано цільову функцію $L(x) = x_1 + x_2$. Потрібно максимізувати цільову функцію $L(x) = x_1 + x_2 \to max$, враховуючи задані обмеження:

$$-x_1 + 2x_2 \leq 6 \quad (1)$$

$$9x_1 + 4x_2 \leq 56 \quad (2)$$

$$3x_1 + 5x_2 \geq 4 \quad (3)$$

$$x_1 \geq 0, x_2 \geq 0$$

Розв'язання:

Необхідно визначити допустиму область D цієї ЗЛП. Для цього у кожній нерівності виразимо x_2 через x_1:

$$x_2 \leq \frac{6 + x_1}{2}$$

$$x_2 \leq \frac{56 - 9x_1}{4}$$

$$x_2 \geq \frac{4 + 3x_1}{5}$$

Якщо нерівності (1), (2), (3) подати як рівняння, то їх графіками будуть прямі на координатній площині. Побудувати ці прямі можна, знайшовши хоча б по дві точки, які належать відповідно цим прямим. Знайдемо координати x_1 і x_2, побудувавши наступні таблички:

x_1	0	-2
x_2	3	2

x_1	4	56/9
x_2	5	0

x_1	0	4/3
x_2	4/5	0

Знаючи координати потрібних точок і враховуючи умови $x_1 \geq 0, x_2 \geq 0$, побудуємо відповідні прямі (Рис. 3).

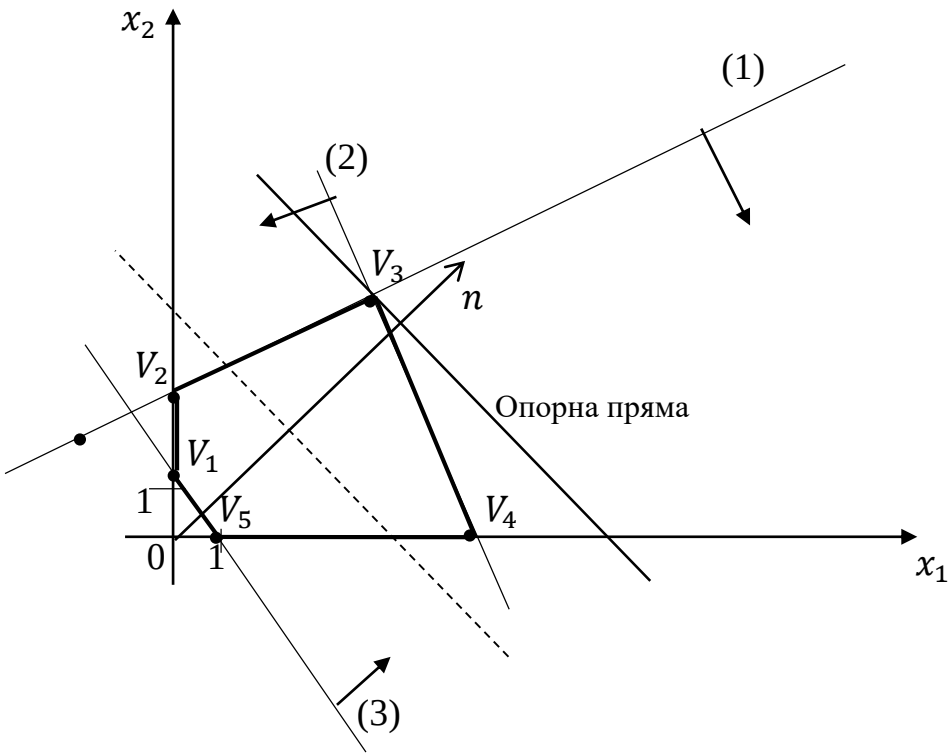


Рис. 3

Оскільки з нерівності (1) ми виразили $x_2 \leq \cdots$, то допустима область знаходиться нижче прямої (1). З нерівності (2) ми виразили $x_2 \leq \cdots$, тому допустима область знаходиться нижче прямої (2). І з нерівності (3) ми виразили $x_2 \geq \cdots$, отже допустима область

знаходиться вище прямої (3). І згідно умов $x_1 \geq 0, x_2 \geq 0$, допустима область знаходиться вище вісі x_1 і справа від вісі x_2.

Отримали, що допустима область D цієї ЗЛП являє собою п'ятикутник з вершинами $V_1(0,4/5)$, $V_2(0,3)$, $V_3(4,5)$, $V_4(56/9,0)$, $V_5(4/3,0)$.

Для того, щоб знайти максимум цільової функції потрібно побудувати вектор нормалі. Для лінійної функції вектор нормалі являє собою вектор коефіцієнтів цільової функції $n(1,1)$. Побудуємо цей вектор із точки (0,0) в точку (1,1).

Далі побудуємо опорну пряму. Опорна пряма це лінія рівня (на Рис. 3 пунктиром). Зазвичай її будують таким чином, щоб вона була перпендикулярна вектору нормалі і проходила через середину допустимої області. Лінія рівня у цьому прикладі має вигляд $x_1 + x_2 = const$.

Відомо, що значення const зростає, якщо лінію рівня переміщувати у напрямку нормалі n. При задачі на максимум переміщуємо опорну пряму у напрямку нормалі n до тих пір, поки вона не буде дотикатись до допустимої області (до границі допустимої області), тобто допустима область залишиться зліва-нижче від опорної прямої. Якби задача була на мінімум, то опорну пряму переміщували б у протилежному напрямі до вектора нормалі.

У цьому прикладі опорна пряма буде дотикатись до допустимої області у точці V_3. Координати точки V_3 знайдемо, розв'язавши систему рівнянь прямих (1) і (2), оскільки вони перетинаються в точці V_3:

$$\begin{cases} -x_1 + 2x_2 = 6 \\ 9x_1 + 4x_2 = 56 \end{cases}$$

Для розв'язання даної системи рівнянь домножимо перше рівняння на 9 і додамо друге рівняння. Отримаємо $22x_2 = 110 \Rightarrow x_2 = 5$. Тоді $x_1 = 4$. Отже, $V_3(4,5)$.

Точка $V_3 \longrightarrow x^*, x^* = (4,5)$ і буде оптимальним розв'язком ЗЛП.

Для того, щоб обчислити максимальне значення цільової функції, підставимо значення x^* у $L(x)$.

$$L(x^*) = 4 + 5 = 9$$

Приклад 2. Розглянемо задачу лінійного програмування. Задано цільову функцію $L(x) = x_1 + x_2$. Потрібно максимізувати цільову функцію $L(x) = x_1 + x_2 \to max$, враховуючи задані обмеження:

$$-x_1 + 2x_2 \leq 6 \quad (1)$$

$$3x_1 + 5x_2 \geq 4 \quad (2)$$

$$x_1 \geq 0, x_2 \geq 0$$

Розв'язання:

Цей приклад одержано з попереднього, де відкинуто друге непряме обмеження. З Рис. 4 випливає, що $max\ L(x) = \infty\ \ x \in D$.

Зауважимо, що, взагалі кажучи, з необмеженості області D не випливає необмеженість зверху (знизу) цільової функції у допустимій області.

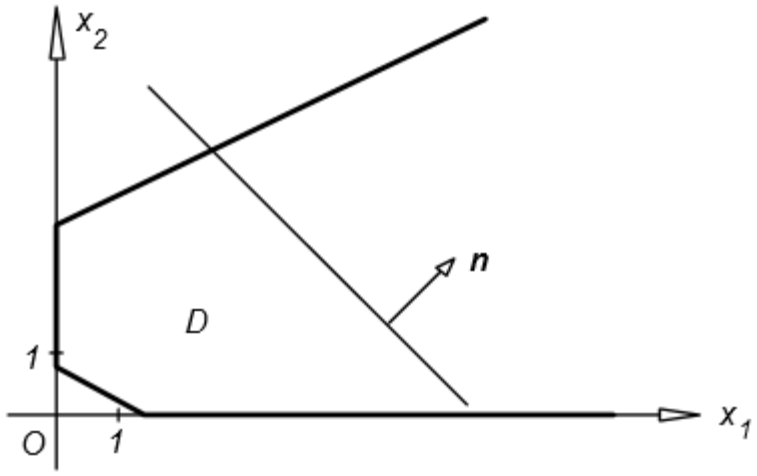


Рис. 4

Приклад 3. Розглянемо задачу лінійного програмування. Задано цільову функцію $L(x) = 9x_1 + 4x_2$. Потрібно максимізувати цільову функцію $L(x) = 9x_1 + 4x_2 \to max$, враховуючи задані обмеження:

$$-x_1 + 2x_2 \leq 6 \quad (1)$$

$$9x_1 + 4x_2 \leq 56 \quad (2)$$

$$3x_1 + 5x_2 \geq 4 \quad (3)$$

$$x_1 \geq 0, x_2 \geq 0$$

Розв'язання:

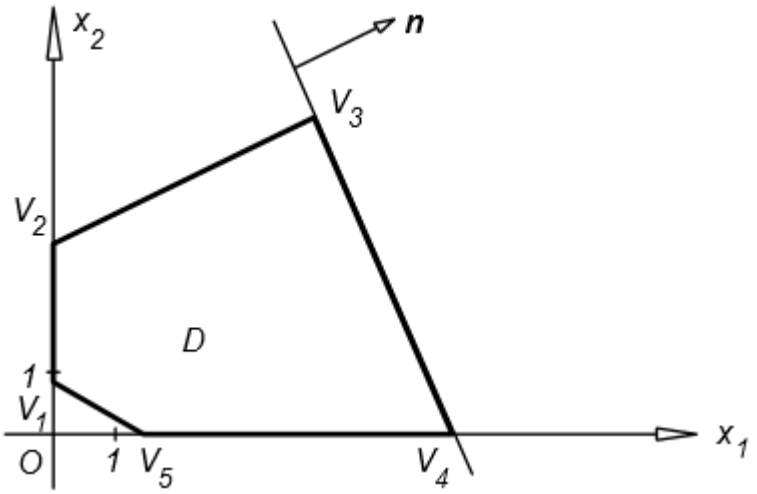


Рис. 5

Цей приклад відрізняється від першого лише цільовою функцією. Аналізуючи Рис. 5, приходимо до висновку, що максимального значення, рівного 56, цільова функція досягає у будь-якій точці сторони V_3V_4 п'ятикутника D.

Питання для самоперевірки:

1. Сформулюйте загальну задачу лінійного програмування.
2. Які існують форми задачі лінійного програмування?
3. В чому суть графічного методу розв'язання задачі лінійного програмування?

Література:

1. Гавриш П.А., Васильєва Л.В. Математичне моделювання систем і процесів : навч. посіб.. Краматорськ : ДДМА, 2007. 100 с.

2. Струченков В.И. Методы оптимизации / Струченков В.И.: - Учебное пособие для вузов. – М.: - Екзамен. – 2005. – 256 с.

3. Пантелеев А.В. Методы оптимизации в примерах и задачах / Пантелеев А.В., Летова Т.А.: - Серия: Прикладная математика для вузов. – М.: Высшая школа. – 2005. – 544 с.

Лекція 4. Методи розв’язання оптимізаційних задач (2 години)

4.1. Симплекс-метод

4.2. Транспортна задача

4.1. Симплекс-метод

Загальним методом розв’язування задач лінійного є симплекс-метод (метод послідовного поліпшення плану), основна ідея якого полягає в тому, щоб збільшуючи значення вільної змінної, збільшити значення цільової функції, але при цьому залишити невід’ємними базисні змінні.

Процес розв’язання задачі симплекс-методом має ітераційний характер: однотипні обчислювальні процедури (ітерації) повторюються у певній послідовності доти, доки не буде отримано оптимальний план задачі або з’ясовано, що його не існує.

Отже, симплекс-метод – це ітераційна обчислювальна процедура, яка дає змогу, починаючи з певного опорного плану, за скінченну кількість кроків отримати оптимальний план задачі лінійного програмування

Симплекс-метод – універсальний, розв’язування ЗЛП від більше ніж 2-х змінних. Симплекс-метод застосовується для ЗЛП в канонічній формі. Алгоритм розв’язування задачі лінійного програмування симплекс-методом складається з п’яти етапів:

1. Визначення початкового опорного плану задачі лінійного програмування. Побудова симплексної таблиці.

2. Перевірка опорного плану на оптимальність за допомогою оцінок. Якщо всі оцінки задовольняють умову оптимальності, то визначений опорний план є оптимальним планом задачі. Якщо хоча б одна з оцінок не задовольняє умову оптимальності, то переходять до нового опорного плану або встановлюють, що оптимального плану задачі не існує.

3. Перехід до нового опорного плану задачі виконується визначенням ключового елемента та розрахунком нової симплексної таблиці.

4. Повторення дій починаючи з п. 2.

Розглянемо докладніше кожний з етапів алгоритму.

1. Визначення першого опорного плану починають із запису задачі лінійного програмування в канонічній формі, тобто у вигляді обмежень-рівнянь з невід'ємними правими частинами. Якщо в умові задачі присутні обмеження-нерівності, то перетворення їх на рівняння виконується за допомогою додаткових змінних, які вводяться до лівої частини обмежень типу «≤» зі знаком «+», а до обмежень типу «≥» – зі знаком «–». У цільовій функції задачі додаткові змінні мають коефіцієнт нуль.

Після зведення задачі до канонічного вигляду виписують матрицю коефіцієнтів. Знаходять лінійно незалежний набір стовпчиків, що створює стовпці одиничної матриці. Ці змінні називаються базисними (y_1, y_2, y_3), інші – небазисні (x_1, x_2). Будують першу симплекс-таблицю: стовпчики матимуть назви небазисних змінних (x_1, x_2), рядки – назви базисних змінних

(y_1, y_2, y_3) у тій послідовності, у якій вони складають одиничну матрицю. Значення у таблиці відповідають значенням небазисних стовпчиків у матриці А. В останньому стовпчику виписують вектор b (стовпчик вільних членів). Останній рядок – це буде рядок оцінок. Далі потрібно розрахувати оцінки і значення цільової функції. Для цього виписують біля кожної змінної її коефіцієнти в цільовій функції. Далі кожну з оцінок обчислюють по правилу: скалярний добуток лівого вектора на вектор x_1 (для наступної оцінки буде вектор x_2 і т.д.) і мінус верхнє значення. У такий спосіб отримують початковий опорний план задачі лінійного програмування. Подальший обчислювальний процес та перевірку опорного плану на оптимальність подають у вигляді симплексної таблиці.

2. Перевіряють опорний план на оптимальність згідно з наведеною далі теоремою.

Теорема (ознака оптимальності опорного плану). Опорний план задачі лінійного програмування є оптимальним, якщо всі оцінки невід'ємні для задачі на max або недодатні для задачі на min.

Якщо для побудови опорного плану було використано метод штучного базису, необхідною умовою оптимальності є також вимога, щоб у процесі розв'язування задачі всі штучні змінні були виведені з базису і дорівнювали нулю. Якщо умова оптимальності не виконується, то потрібно виконати перехід до наступного, нового опорного плану задачі.

3. Перехід від одного опорного плану до іншого виконується за допомогою Жорданових перетворень. Спочатку визначають ключовий елемент (x_{rs}), що знаходиться на перетині головного рядка і головного стовпчика:

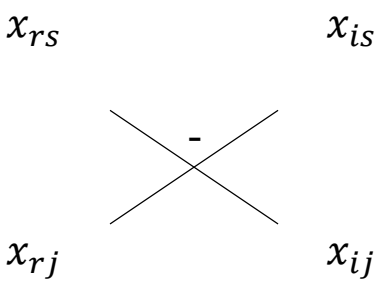
1) головним стовпчиком буде стовпчик з найбільшою по модулю від'ємною оцінкою;

2) для визначення головного рядка складають відношення правих частин до додатніх елементів головного стовбця; і головним вибирають рядок де симплексне відношення найменше.

Алгоритм Жорданових перетворень

1. Поміняти місцями позначення головного рядка і головного стовпця: $x_1 \leftrightarrow y_1$
2. Ключовий елемент замінити на обернений: $x_{rs} \rightarrow \frac{1}{x_{rs}}$
3. Усі елементи головного рядка поділити на ключовий елемент: $\div\, x_{rs}$
4. Усі елементи головного стовпця діляться на x_{rs} і міняють знак на протилежний: $\div\, (-x_{rs})$
5. Наступні елементи $x_{ij}^{'}$ обчислюють за методом прямокутників:

$x_{ij}^{'} = \frac{x_{rs} * x_{ij} - x_{rj} * x_{is}}{x_{rs}}$ або $x_{ij}^{'} = x_{ij} - \frac{x_{rj} * x_{is}}{x_{rs}}$



Далі ітераційний процес повторюють доти, доки не буде визначено оптимальний план задачі.

4.2. Транспортна задача

Постановка транспортної задачі:

$$C(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \to \min \ - \text{цільова функція (сумарні витрати)}$$

$$\begin{cases} \sum_{j=1}^{m} x_{ij} \le a_i \\ \sum_{i=1}^{n} x_{ij} \le b_j \\ x_{ij} \ge 0, \\ i = \overline{1,n}\,, j = \overline{1,m} \end{cases} \quad - \text{обмеження}$$

n – кількість вузлів виробництва $A_i, i = \overline{1,n}$ a_i – потужність виробленої електроенергії у вузлі A_i,

m – кількість пунктів споживання $B_j, j = \overline{1,m}$ b_j – потреба вузла споживання B_j.

Матриця витрат при передачі:

$$C = \begin{pmatrix} c_1 & c_{12} & \dots & c_{1m} \\ \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{pmatrix}$$

x_{ij} – кількість одиниць електроенергії, яку передають із вузла генерування у вузол споживання $A_i \to B_j$.

Транспортна таблиця має вигляд (Таблиця 2):

Таблиця 2. Транспортна таблиця

c_{11}	c_{12}	$\dots$	c_{1m}	a_1
c_{21}	c_{22}	$\dots$	c_{2m}	a_2
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
c_{n1}	c_{n2}	$\dots$	c_{nm}	a_n
b_1	b_2	$\dots$	b_m	a_i / b_j

Транспортна задача буває двох типів: *відкрита* і *закрита* (збалансована).

Умова балансу:

$$\sum_{i=1}^{n} a_i = \sum_{j=1}^{m} b_j$$

Якщо умова балансу порушена, то задачу потрібно збалансувати.

1) якщо

$$\sum_{i=1}^{n} a_i > \sum_{j=1}^{m} b_j ,$$

то потрібно ввести фіктивний магазин b_0:

$$b_0 = \sum_{i=1}^{n} a_i - \sum_{j=1}^{m} b_j ;$$

2) якщо

$$\sum_{i=1}^{n} a_i < \sum_{j=1}^{m} b_j ,$$

то потрібно ввести фіктивний склад a_0:

$$a_0 = \sum_{j=1}^{n} b_j - \sum_{i=1}^{n} a_i.$$

Вартість перевезення c_{ij} при цьому вважають = 0.

Для знаходження опорного розв'язку транспортної задачі використовують метод «північно-західного кута» і метод мінімальної вартості (найменшого елемента).

1. Метод «північно-західного кута». Суть методу – на кожному кроці обирається верхня ліва клітинка і вписується в неї $x_{ij} = \min(\mathrm{a_i}, \mathrm{b_j})$.

2. Метод мінімізації витрат (найменшого елемента). Суть методу – на кожному кроці обирається клітинка з найменшим значенням c_{ij} і вписується в неї $x_{ij} = \min(a_i, b_j)$.

Для перевірки оптимальності розв'язання транспортної задачі використовують метод потенціалів. Суть методу – необхідно розставити потенціали (змінні u_i, v_j), для яких опорний розв'язок буде оптимальним тоді, коли виконуються умови:

$$\forall(i,j)_{\text{базисні}} \qquad u_i + v_j = c_{ij} \tag{1}$$

$$\forall(i,j)_{\text{небазисні}} \qquad u_i + v_j \le c_{ij} \tag{2}$$

Іноді останню умову описують за допомогою оцінок

$$\Delta_{ij} = u_i + v_j - c_{ij} \le 0$$

Якщо ці умови виконуються, то оптимальний розв'язок знайдено.

Питання для самоперевірки:

1. Опишіть симплексний метод розв'язування задач лінійного програмування.
2. Математична постановка транспортної задачі.
3. Які існують методи побудови початкового опорного плану транспортної задачі? В чому суть кожного з них?
4. В чому суть методу потенціалів для знаходження оптимального плану транспортної задачі?

Література:

1. Струченков В.И. Методы оптимизации / Струченков В.И.: - Учебное пособие для вузов. – М.: - Екзамен. – 2005. – 256 с.
2. Пантелеев А.В. Методы оптимизации в примерах и задачах / Пантелеев А.В., Летова Т.А.: - Серия: Прикладная математика для вузов. – М.: Высшая школа. – 2005. – 544 с.

Лекція 5. Багатокритеріальні задачі (2 години)

5.1. Задачі багатокритеріальної оптимізації

5.2. Методи вирішення задач багатокритеріальної оптимізації

5.3. Метод послідовних поступок

5.1. Задачі багатокритеріальної оптимізації

У практичній діяльності часто зустрічаються задачі, що полягають у пошуку найкращого (оптимального) рішення при наявності різних незвідних один до одного критеріїв оптимальності. Наприклад, прийняття рішення про будівництво лінії електропередач в об'їзд міста повинно враховувати такі чинники, як виграш міста в цілому з міркувань екології, програш окремих підприємств і фірм і багато інших. Якщо такого роду задачі розв'язуються методами математичного програмування, то говорять про задачі багатокритеріальної оптимізації. Ці задачі можуть носити як лінійний, так і нелінійний характер. Методи вирішення таких задач розглянемо на прикладі лінійних багатокритеріальних оптимізаційних задач.

Задачі багатокритеріальної оптимізації виникають у тих випадках, коли є кілька цілей, які не можуть бути відображені одним критерієм (наприклад, вартість і надійність). Потрібно знайти точку області допустимих рішень, яка мінімізує або максимізує всі такі критерії. Якщо в подібного роду задачах мова йде не про різнорідні критерії деякої системи, а про зіставлення однорідних критеріїв

різних її підсистем (наприклад, галузі, групи населення і т. п.), то ці задачі називаються задачами векторної оптимізації.

Деякі часткові критерії можуть суперечити один одному, інші діють в одному напрямку, треті – індиферентні, байдужі один до одного. Тому процес вирішення багатокритеріальних задач неминуче пов'язаний з експертними оцінками як самих критеріїв, так і взаємин між ними.

5.2. Методи вирішення задач багатокритеріальної оптимізації

Відомий ряд методів вирішення задач багатокритеріальної оптимізації:

– оптимізація одного, визнаного найбільш важливим, критерію, інші критерії при цьому відіграють роль додаткових обмежень;

– упорядкування заданої множини критеріїв і послідовна оптимізація за кожним з них;

– зведення багатьох критеріїв до одного введенням експертних вагових коефіцієнтів для кожного з критеріїв таким чином, що більш важливий критерій отримує більш високу вагу.

В ідеальному випадку можна вести пошук такого рішення, яке належить перетину множин оптимальних рішень всіх однокритеріальних задач. Однак такий перетин зазвичай виявляється порожньою множиною, тому доводиться розглядати так звану переговорну множину ефективних рішень (оптимальних за Парето). Критерій оптимальності італійського економіста Ст. Парето застосовується при вирішенні таких задач, коли

оптимізація означає поліпшення одних показників при умові, щоб інші не погіршувалися.

Множину допустимих розв'язків, для яких неможливо одночасно поліпшити всі частинні показники ефективності (тобто покращити хоча б один з них, не погіршуючи інших), прийнято називати областю Парето, або областю компромісів, а розв'язки, що їй належать – ефективними, або оптимальними за Парето.

У загальному випадку ефективні рішення не еквівалентні один одному, так що про два оптимальних за Парето розв'язки не можна сказати, який з них краще. Тому при вирішенні багатокритеріальних задач необхідне додаткове вивчення ефективних розв'язків. Для цього можна було б сформулювати деякий критерій і оптимізувати його на множині ефективних розв'язків. Однак при цьому виникають значні труднощі у зв'язку з тим, що, як правило, область компромісів не є опуклою, і отримана задача в загальному випадку буде завданням неопуклого програмування. Звичайний підхід полягає в прагненні «згорнути» частинні критерії в один узагальнений скалярний критерій, оптимізація якого призводить до оптимального рішення задачі в цілому. Формулювання відповідного узагальненого критерію в залежності від конкретних умов якраз і є основним питанням, яке вивчається в багатокритеріальної оптимізації.

В деяких випадках замість одного узагальненого критерію і вирішення однієї відповідної задачі скалярної оптимізації пропонується розглядати послідовність узагальнених критеріїв і

послідовність завдань скалярної оптимізації. На жаль, багато з описаних в літературі подібних процедур не завжди призводять до ефективних рішень.

5.3. Метод послідовних поступок

Метод послідовних поступок вирішення задач багатокритеріальної оптимізації застосовується у разі, коли частинні критерії можуть бути впорядковані у порядку зменшення їх важливості. Припустимо, що всі частинні критерії максимізуються і пронумеровані в порядку зменшення їх важливості. Знаходимо максимальне значення Z_1^*, першого по важливості критерію в області допустимих розв'язків, шляхом вирішення задачі однокритеріальної

$$Z_1(\bar{X}) \to \max,$$
$$\bar{X} \in Q.$$

Потім, виходячи з практичних міркувань і прийнятої точності, призначається величина допустимого відхилення, > 0 (економічно виправданої поступки) критерію Z, і знаходиться максимальне значення другого критерію Z'_2 за умови, що значення першого критерію не повинно відхилятися від свого максимального значення більш ніж на величину допустимої поступки, тобто вирішується задача:

$$Z_2(\bar{X}) \to \max,$$
$$Z_1(\bar{X}) \geq Z_1^* - \delta_1,$$
$$\bar{X} \in Q.$$

Знову призначається величина поступки $\delta_2 > 0$ за другим критерієм, яка разом з першою поступкою використовується для знаходження умовного максимуму третього частинного критерію:

$$
\begin{aligned}
&Z_3(\overline{X}) \to \max,\\
&Z_1(\overline{X}) \geq Z_1^* - \delta_1,\\
&Z_2(\overline{X}) \geq Z_2^* - \delta_2,\\
&\overline{X} \in Q.
\end{aligned}
$$

Аналогічні процедури повторюються до тих пір, поки не буде виявлено максимальне значення останнього по важливості критерію Z_m за умови, що значення кожного з перших m -1 частинних критеріїв відрізняється від відповідного умовного максимуму не більше ніж на величину допустимої поступки за даним критерієм. Отримане на останньому етапі рішення вважається оптимальним. Слід зауважити, що цей метод не завжди призводить до ефективного вирішення.

Питання для самоперевірки:

1. Охарактеризуйте багатокритеріальну оптимізацію.
2. В чому суть методу послідовних поступок?

Література:

1. Струченков В.И. Методы оптимизации / Струченков В.И.: - Учебное пособие для вузов. – М.: - Екзамен. – 2005. – 256 с.

2. Пантелеев А.В. Методы оптимизации в примерах и задачах / Пантелеев А.В., Летова Т.А.: - Серия: Прикладная математика для вузов. – М.: Высшая школа. – 2005. – 544 с.

Змістовий модуль 3. Математичне програмне забезпечення (6 годин)

Лекція 6. Основи роботи в MathCAD (4 години)

6.1. Загальні відомості про MathCAD

6.2. Математичні вирази

6.1. Загальні відомості про MathCAD

MathCAD працює з *документами*. З погляду користувача, документ – це чистий аркуш паперу, на якому можна розміщати блоки трьох основних типів: математичні вирази, текстові фрагменти і графічні області.

Розташування нетекстових блоків у документі має принципове значення – *зліва направо і зверху вниз.*

6.2. Математичні вирази

До основних елементів математичних виразів MathCAD відносяться *типи даних, оператори, функції* і *керуючі структури.*

Оператори – елементи MathCAD, за допомогою яких можна створювати математичні вирази. До них, наприклад, відносяться символи арифметичних операцій, знаки обчислення сум, добутків, похідної, інтегралу і т.д.

Оператор визначає:

1) дію, що повинна виконуватися при наявності тих чи інших значень операндів;
2) скільки, де і які операнди повинні бути введені в оператор.

Операнд – число чи вираз, на яке діє оператор. Наприклад, у виразі **5! + 3** число **3** і вираз **5!** – операнди оператору **+** (плюс), а число **5** операнд оператору факторіал (!). Після вказівки *операндів* оператори стають блоками, що виконуються у документі.

До *типів даних* відносяться числові константи, звичайні і системні змінні, масиви (вектори і матриці) і дані файлового типу.

Константами називають пойменовані об'єкти, що зберігають деякі значення, що не можуть бути змінені. Змінні є пойменованими об'єктами, що мають деяке значення, що може змінюватися по ходу виконання програми. Тип змінної визначається її значенням; змінні можуть бути числовими, рядковими, символьними і т.д. Імена констант, змінних і інших об'єктів називають ідентифікаторами. Ідентифікатори в MathCAD являють собою набір латинських чи грецьких букв і цифр.

У MathCAD міститься невелика група особливих об'єктів, які не можна віднести ні до класу констант, ні до класу змінних, значення яких визначені одразу після запуску програми. Їх вірніше вважати системними змінними, що мають визначені системою початкові значення. Зміну значень системних змінних роблять у вкладці Вбудовані змінні діалогового вікна Math Options команди Математика ⇒ Опції.

Звичайні змінні відрізняються від системних тим, що вони повинні бути попередньо визначені користувачем, тобто їм необхідно хоча б один раз присвоїти значення. У якості оператора

присвоєння використовується знак «:=», тоді як знак «=» відведений для виводу значення чи константи змінної.

Якщо змінній присвоюється початкове значення за допомогою оператора «:=» викликається натисканням клавіші «:» (двокрапка) на клавіатурі, таке присвоєння називається локальним. До цього присвоєння змінна не визначена і її не можна використовувати. Однак за допомогою знака «≡» (клавіша «~» на клавіатурі) можна забезпечити глобальне присвоєння. MathCAD прочитує весь документ двічі зліва направо і зверху вниз. При першому проході виконуються всі дії, запропоновані глобальним оператором присвоєння (≡), а при другому – виробляються дії, запропоновані локальним оператором присвоєння (:=), і відображаються всі необхідні результати обчислень (=).

Дискретні аргументи – особливий клас змінних, який у пакеті MathCAD найчастіше заміняє керуючі структури, що називають циклами (однак повноцінною така змінна не є). Ці змінні мають ряд фіксованих значень, або цілочисельних, або у вигляді чисел з визначеним кроком, що міняються від початкового значення до кінцевого.

Дискретні аргументи значно розширюють можливості MathCAD, дозволяючи виконувати багаторазові обчислення чи цикли з повторними обчисленнями, формувати вектори і матриці.

Масив – сукупність, що має унікальне ім’я, кінцевого числа числових чи символьних елементів, впорядкованих деяким чином і

що мають визначені адреси. У пакеті MathCAD використовуються масиви двох найбільш розповсюджених типів:

– одновимірні (вектори);

– двовимірні (матриці).

Порядковий номер елемента, що є його адресою, називається *індексом*. Індекси можуть мати тільки цілочисельні значення. Вони можуть починатися з нуля чи одиниці, у відповідності зі значенням системної змінний **ORIGIN**.

Вектори і матриці можна задавати різними способами:

– за допомогою команди **Вставка ⇒ Матриця**, чи комбінації клавіш **Ctrl + M,** заповнивши масив порожніх полів для не занадто великих масивів;

– з використанням дискретного аргументу, коли має місце деяка явна залежність для обчислення елементів через їхні індекси.

Функція – вираз, відповідно до якого проводяться деякі обчислення з аргументами і визначається його числове значення.

Слід особливо зазначити різницю між аргументами і параметрами функції. Змінні, зазначені в дужках після імені функції, є її аргументами і замінюються при обчисленні функції значеннями з дужок. Змінні в правій частині визначення функції, не зазначені дужках у лівій частині, є параметрами і повинні задаватися до визначення функції.

Головною ознакою функції є повернення значення, тобто функція у відповідь на звернення до неї по імені з вказівкою її аргументів повинна повернути своє значення.

Функції в пакеті MathCAD можуть бути вбудовані, тобто завчасно введені розроблювачами, і визначені користувачем.

Текстові фрагменти являють собою частини тексту, що користувач хотів би бачити у своєму документі. Існують два види текстових фрагментів:

– текстова область призначена для невеликих частин тексту – підписів, коментарів і т.п. Вставляється за допомогою команди **Вставка ⇒ Текстова** або комбінації клавіш **Shift + "** (подвійні лапки);

– текстовий абзац застосовується в тому випадку, якщо необхідно працювати з абзацами чи сторінками. Вставляється за допомогою комбінації клавіш **Shift + Enter**.

Графічні області поділяються на три основних типи – двовимірні графіки, тривимірні графіки й імпортовані графічні образи. Двовимірні і тривимірні графіки будуються самим MathCAD на підставі оброблених даних.

Питання для самоперевірки:

1. Для чого призначене програмне забезпечення Mathcad?
2. Які математичні операції можна виконувати в Mathcad?

Література:

1. Алексеев Е. Р. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9 / Е. Р. Алексеев, О.В. Чеснокова. – М. : НТ Пресс, 2006. – 496 с..

Лекція 7. Основи роботи в MatLAB (2 години)

7.1. Загальні відомості про MatLAB

7.2. Робоче середовище MatLAB

7.3. Найпростіші обчислення

7.1. Загальні відомості про MatLAB

MATLAB (скорочення від англ. «Matrix Laboratory») – пакет прикладних програм для вирішення завдань технічної обробки даних та однойменна мова програмування, що використовується в цьому пакеті. MATLAB працює на більшості сучасних операційних систем, включаючи Linux, Mac OS, Solaris (починаючи з версії R2010b підтримка Solaris припинена) і Microsoft Windows.

Мова MATLAB є високорівневою інтерпретованою мовою програмування, що включає засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об’єктно-орієнтовані можливості та інтерфейси для програм, написаних на інших мовах програмування.

В даний час MatLab є потужним і універсальним засобом вирішення завдань, що виникають в різних областях людської діяльності. Спектр проблем, дослідження яких може бути здійснено за допомогою MatLab, охоплює: матричний аналіз, обробку сигналів і зображень, задачі математичної фізики, оптимізаційні задачі, обробку та візуалізацію даних, роботу з картографічними зображеннями, нейронні мережі, нечітку логіку та багато інших. Спеціалізовані засоби зібрані в пакети, що називаються ***ToolBox***, і

можуть бути вибірково встановлені разом з MatLab за бажанням користувача. До складу багатьох ToolBox входять програми з графічним інтерфейсом користувача, які забезпечують швидкий і наочний доступ до основних функцій. Пакет ***Simulink***, що поставляється разом з MatLab, призначений для інтерактивного моделювання нелінійних динамічних систем, що складаються зі стандартних блоків.

MatLab без проблем інтегрується з Microsoft Word і Excel. Зв'язок MatLab і Word забезпечує можливість написання в редакторі Word інтерактивних документів, так званих ***М-книг***, заснованих на спеціальному шаблоні. Користувач, який працює з М-книгою, може запускати блоки команд MatLab безпосередньо з документа Word, причому результат виконання команд відображається в М-книзі.

7.2. Робоче середовище MatLAB

Робоче середовище MatLab містить такі елементи:

- меню;
- панель інструментів з кнопками та списком;
- вікно з вкладками Launch Pad і Workspace, з якого можна отримати простий доступ до різних модулів ToolBox та до вмісту робочого середовища;
- вікно з вкладками Command History і Current Directory, призначене для перегляду і повторного виклику раніше введених команд, а також для установки поточного каталогу;
- командне вікно;

– рядок стану.

Всі основні команди набираються в командному вікні.

Вбудовані математичні функції MatLab дозволяють знаходити значення різних виразів. MatLab надає можливість керування форматом виведення результату. Команди для обчислення виразів мають вигляд, властивий всім мовам програмування високого рівня.

7.3. Найпростіші обчислення

Наберіть в командному рядку ***1+2*** та натисніть ***Enter***. В результаті в командному вікні MatLab відобразиться наступне:

```
>> 1+2
ans =
        3
>>
```

Програма MatLab спочатку обрахувала суму ***1+2***, потім записала результат в спеціальну змінну ***ans*** та вивела її значення, що рівне ***3***, в командне вікно. Нижче відповіді розташований командний рядок з курсором, який показує, що MatLab готова до подальших обчислень. Можна набирати в командному рядку нові вирази та знаходити їх значення.

Необхідний формат виведення результату визначається користувачем з меню MatLab. Виберіть в меню ***File*** пункт ***Preferences***. Для встановлення формату виведення слід переконатися, що у списку лівої панелі обраний пункт ***Command Window***. Задання формату проводиться зі списку ***Numeric format*** панелі ***Text display***. Розглянемо тільки найбільш часто

використовувані формати. Виберіть ***short*** в списку ***Numeric format***. Закрийте діалогове вікно, натиснувши кнопку ***OK***. Встановлено короткий формат з плаваючою точкою ***short*** для виведення результатів обчислень, при якому на екрані відображаються лише чотири цифри після десяткової точки.

Якщо потрібно отримати результат обчислень більш точно, то в діалоговому вікні ***Preferences*** слід вибрати в списку ***long***. Результат буде відображатися в довгому форматі з плаваючою точкою ***long*** з чотирнадцятьма цифрами після десяткової точки. Формати ***short e*** та ***long e*** призначені для виведення результату в експоненціальній формі з чотирма і п'ятнадцятьма цифрами після десяткової точки відповідно.

Задавати формат виводу можна безпосередньо з командного рядка за допомогою команди ***format***. Наприклад, для встановлення довгого з плаваючою точкою формату виводу результатів обчислень слід ввести команду ***format long e*** в командному рядку.

MatLab розрізняє прописні і малі літери. Спроба набору команди прописними буквами призведе до помилки.

Питання для самоперевірки:

1. Для чого призначене програмне забезпечення Matlab?
2. Які математичні операції можна виконувати в Matlab?

Література:

1. Алексеев Е. Р. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9 / Е. Р. Алексеев, О.В.

Чеснокова. – М. : НТ Пресс, 2006. – 496 с..

2. Паасонен В.И. Инструмент научных исследований MATLAB. Учеб. пособие. Новосиб. ун-т. Новосибирск, 2000. - 61 с.

Навчально-методичне видання

МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ МАГІСТЕРСЬКИХ ПРОГРАМ

Опорний конспект лекцій для здобувачів вищої освіти ступеня «магістр»
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»
денної та заочної форм навчання

Укладачі:
Шебаніна Олена В'ячеславівна
Домаскіна Марина Анатоліївна
Ручинська Наталія Сергіївна та ін.

Формат 60х84/16. Ум. друк. арк. 4.
Тираж 50 пр.. Зам. №

Видавничий відділ
Миколаївського національного аграрного університету
54029, м. Миколаїв, вул. Георгія Гонгадзе, 9.

Свідоцтво суб'єкта видавничої справи ДК №4490 від 20.02.2013 р.