



**УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

В.В. Шевченко, В.Ф. Головко

**АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ
ВАГОНІВ**

НАВЧАЛЬНИЙ ПОСІБНИК

Харків 2008

ББК 65.9(2)21
УДК 681.3.06

В.В. Шевченко, В.Ф. Головка Автоматизоване проектування вагонів:
Навчальний посібник. – Харків: УкрДАЗТ, 2008. - 214 с.

У навчальному посібнику розглядаються основи автоматизованого проектування технічних об'єктів. Викладається структура систем автоматизованого проектування, принципи розроблення математичних моделей і розв'язання задач аналізу і синтезу технічних об'єктів. Значна увага приділена оптимальному проектуванню конструкцій. Наводяться приклади розв'язання завдань автоматизованого проектування, у тому числі завдань геометричного моделювання і розрахунку на міцність деталей рухливого складу.

Навчальний посібник призначений для студентів навчальних закладів залізничного транспорту, а також може бути корисним науковим і інженерно - технічним працівникам, діяльність яких пов'язана із проектуванням, ремонтом та експлуатацією рухливого складу.

Іл. 127, табл. 7, бібліогр.: 19 назв.

ISBN 978-966-7593-86-5

*Рекомендовано Міністерством освіти і науки України як навчальний
посібник для студентів вищих навчальних закладів залізничного
транспорту
№ 1.4/18-Г-795 від 23.05.07.*

Рецензенти:

професори І.Г. Міренський (Харківська національна академія
міського господарства),
В.Г. Маслієв (Національний технічний університет
«Харківський політехнічний інститут»),
Б.О. Перепелиця (Національний технічний університет
«Харківський політехнічний інститут»)

© Шевченко В.В., Головка В.Ф.
© УкрДАЗТ, 2008

ЗМІСТ

Вступ	5
1. Системи автоматизованого проектування	7
1.1. Принципи й задачі проектування	7
1.2. Структура САПР	10
1.3. Класифікація САПР	12
2. Математичні моделі технічних об'єктів	14
2.1. Вимоги, які пред'являються до математичних моделей	14
2.2. Класифікація математичних моделей	16
2.3. Типові математичні моделі технічних систем	19
3. Задачі аналізу і синтезу в системах автоматизованого проектування	29
3.1. Класифікація задач аналізу і синтезу	29
3.2. Математична постановка типових задач аналізу	30
3.3. Задачі структурного синтезу в системах автоматизованого проектування	32
3.3.1. Формальне подання структури системи	32
3.3.2. Методи й алгоритми структурного синтезу	34
3.3.3. Автоматизація конструкторського й технологічного проектування	47
4. Методи оптимізації технічних об'єктів	51
4.1. Задачі оптимального проектування	51
4.2. Методи розв'язання задач оптимального проектування	56
4.2.1. Основи теорії оптимізації	56
4.2.2. Методи одновимірного пошуку	63
4.2.3. Методи багатовимірного пошуку	68
4.3. Лінійне програмування	72
4.4. Нелінійне програмування	75
4.5. Застосування комп'ютера для пошуку екстремуму функції	80

5.	Геометричне моделювання на базі системи AutoCAD	90
5.1.	Вікно графічного редактора	90
5.2.	Системи координат	93
5.3.	Креслення об'єктів	98
5.4.	Лінійні перетворення об'єктів	101
5.5.	Методи формування плоских геометричних об'єктів	106
5.6.	Об'єктна прив'язка	110
5.7.	Метричні задачі	117
5.8.	Проставляння розмірів	119
5.9.	Приклад розроблення плоских моделей	122
6.	Розрахунок на міцність в MSC/NASTRAN	127
6.1.	Основні етапи підготовки	127
6.2.	Інтерфейс програми	128
6.3.	Створення точок, прямих ліній, дуг, окружностей і сплайнів	132
6.4.	Створення поверхонь	139
6.5.	Розрахунок важеля гальмової передачі вагона	142
7.	Імітаційне моделювання	167
7.1.	Поняття імітаційної моделі	167
7.2.	Системи масового обслуговування	171
7.3.	Система моделювання GPSS World	174
7.4.	Оператори мови GPSS World	177
7.5.	Розроблення моделі в GPSS World	182
7.6.	Моделювання роботи ділянки цеху	190
7.7.	Моделювання ремонту колісних пар вагонів	199
	Список літератури	212

ВСТУП

Пішли в минуле часи, коли робочими інструментами конструктора були кульман, олівець і гумка. Сучасні технології базуються на системах автоматизованого проектування (САПР), які дозволяють істотно знизити час, трудомісткість і вартість розроблення, підвищити якість проектування за рахунок автоматизації виконання креслень і необхідних розрахунків.

Автоматизоване проектування використовує засоби обчислювальної техніки й дозволяє вирішувати такі задачі, які важко було вирішувати традиційними ручними способами. Це задачі оптимального проектування конструкцій, використання таких сучасних методів дослідження напруженого стану конструкцій, як метод кінцевих елементів.

Автоматизація проектування дозволяє підвищити якість як змісту, так і зовнішнього оформлення проектів. У першому випадку завдяки розв'язанню оптимізаційних задач, а також можливості застосування сучасних методів розрахунку на міцність (заснованих на використанні складного математичного апарата) вдається покращити техніко - економічні характеристики проєктованих машин, наприклад, зменшити матеріаломісткість, габаритні розміри, підвищити надійність і т.п.

Технічний об'єкт, який проєктується, може характеризуватися декількома критеріями якості. У цьому випадку звичайно на основі часткових критеріїв формується узагальнений критерій.

Скорочення строків проектування при використанні САПР досягається завдяки прискоренню виконання розрахункових операцій і креслення рисунків, скороченню строків пошуку необхідної інформації внаслідок наявності в пам'яті системи інформаційної бази, що містить всі необхідні дані.

У цей час поряд з використанням САПР конструкцій застосовуються САПР технологічних процесів. За структурою, принципами побудови й стадіями розроблення ці системи багато в чому аналогічні. Наявність системи автоматизованого проектування виробу створює передумови для більш ефективної організації САПР технологічних процесів його виготовлення, оскільки результати проектування виробу, що зберігаються в

пам'яті ЕОМ, можуть використовуватися для проектування технологічного процесу його виготовлення.

Системи автоматизованого проектування вперше були створені в авіаційній і автомобільній галузях промисловості, для яких найбільш важливі швидке освоєння нових моделей машин, поліпшення якості проектів, скорочення строків їхнього розроблення.

Автоматизоване проектування вагонів почалося на початку 80-х років.

Розроблення системи автоматизованого проектування вантажних вагонів (САПР «Вагон») почалося з автоматизації розрахунків (дослідницьких, міцності, надійності), а також зі статистичної обробки результатів випробувань на ЕОМ. Програмне забезпечення створювалося провідними вченими в галузі залізничного транспорту й фахівцями вагонобудівних заводів. З початку 90-х років почалося виконання текстової й графічної документації.

Впровадження комп'ютерної техніки привело до зміни технології проектування. Застосовується параметричне й твердотільне моделювання геометрії, імітаційне моделювання кінематики і динаміки, а також аналіз міцності конструкцій методом кінцевих елементів. У процесі проектування програмні комплекси послідовно передають дані, отримані на кожному етапі проектування.

У навчальному посібнику розглядаються основні компоненти системи автоматизованого проектування. Докладно викладаються методи оптимального проектування конструкцій.

Навчальний посібник призначений для студентів вищих навчальних закладів, які навчаються за фахом „Рухомий склад і спеціальна техніка залізничного транспорту”, а також може бути корисним для інженерно-технічних працівників залізничного транспорту й вагонобудування.

1. СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

1.1. Принципи і задачі проектування

Створення будь-якого об'єкта починається з його проектування. У процесі проектування розробляють опис об'єкта, що ще не існує, але вже задуманий і буде створений у майбутньому. Проектування включає комплекс робіт з досліджень, розрахунків і конструювання виробу. Ціль проектування полягає в отриманні опису об'єкта проектування, необхідного для створення нового виробу або реалізації нового процесу, що задовольняє задані вимоги. Проектування - творчий процес, що може бути реалізований тільки при активній участі людини.

Автоматизоване проектування має передумовою виконання проектних процедур й операцій з використанням ЕОМ. При автоматизованому проектуванні можливий такий розподіл функцій між людиною і ЕОМ, при якому людина в основному вирішує задачі творчого характеру, а ЕОМ - задачі, що припускають формалізований опис у вигляді алгоритмів.

При проектуванні складних систем використовується блоково - ієрархічний підхід. Його суть заключається в розчленовуванні системи на ієрархічні рівні. На найвищому рівні використовується найменш деталізований опис загальних властивостей об'єкта. Далі система розчленовується на окремі блоки, які утворюють наступний ієрархічний рівень. Зі збільшенням числа рівнів ступінь деталізації зростає.

Кожен ієрархічний рівень має своє поняття системи та елементів. Наприклад, вагон на верхньому рівні ієрархії може розглядатися як система. Елементами цієї системи є кузов, візок, автогальмове устаткування й ін. На наступному рівні візок розглядається як система, елементами якої є надресорна балка, бічні рами, колісні пари, букси, ресорне підвішування. Ще більш низький рівень дозволяє розглядати буксу як систему, елементами якої будуть корпус, підшипники й т.д. Такий поділ триває доти, поки не залишаться елементи, які подальшому розчленовуванню не підлягають. Ці елементи називаються базовими елементами.

Властивості системи можна розділити на окремі групи. Така декомпозиція приводить до появи різних аспектів опису властивостей. Звичайно виділяють функціональний, конструкторський і технологічний аспекти. Розв'язання задач, пов'язаних із цими аспектами, називають функціональним, конструкторським і технологічним проектуванням.

Функціональний аспект характеризує основні принципи функціонування об'єкта. Для відображення цих властивостей використовуються принципи, структурні, функціональні, кінематичні схеми і супровідні їхні документи.

Конструкторський аспект опирається на результати функціонального проектування й пов'язаний з визначенням геометричних форм об'єктів і їхнім взаємним розташуванням у просторі.

Технологічний аспект пов'язаний з описом методів і засобів виготовлення проектного об'єкта.

При проектуванні складних систем виділяють кілька стадій: передпроектні дослідження, розроблення технічного завдання й технічної пропозиції, ескізного, технічного й робочого проектів, випробування й впровадження.

Стадія передпроектних досліджень або стадія науково-дослідних робіт (НДР) пов'язана з пошуком принципів можливостей створення системи. Результатом цієї роботи є технічна пропозиція.

Технічна пропозиція передбачає:

- вишукування або розроблення принципів побудови, розроблення структурної схеми, вибір технічних засобів;
- математичне моделювання проектного системи;
- аналіз результатів і порівняння з технічним завданням.

Стадія ескізного проектування передбачає:

- уточнення структурної схеми;
- уточнення складу підсистем, їх функцій, основних характеристик;
- аналіз характеристик технічних засобів, проведення оптимізації, скорочення номенклатури технічних засобів;

- моделювання (математичне, імітаційне, фізичне);
- виготовлення експериментальних зразків.

Стадія технічного проектування передбачає поглиблення й деталізацію отриманих результатів в умовах, близьких до експлуатаційних.

На стадії робочого проектування створюється сукупність документів для виготовлення й випробування дослідного зразка.

Етапом проектування називають складову частину процесу проектування, що зводиться до формування всіх необхідних описів об'єкта, що відносяться до одного аспекту або ієрархічного рівня. Проектною процедурою називають складову частину етапу проектування. Виконання проектної процедури закінчується проектним рішенням. Проектна процедура ділиться на більш дрібні частини, які називаються проектними операціями. Як приклад проектних процедур можна навести вибір типової конструкції, оформлення креслення виробу. Прикладами проектних операцій можуть служити креслення типового графічного зображення (зубчастого зачеплення, рамки креслення й т.п.), розв'язання системи алгебраїчних рівнянь.

Послідовність розв'язання задачі кожного ієрархічного рівня може бути різною. Якщо спочатку розв'язуються задачі високого ієрархічного рівня, а потім задачі більш низького рівня, то таке проектування називають спадним (проектування «зверху вниз»).

При висхідному проектуванні (проектуванні «знизу нагору») спочатку розробляються елементи, з яких пізніше формується система.

Кожний із цих видів проектування має позитивні й негативні сторони. При спадному проектуванні характеристики елементів ще не визначені і, отже, носять можливий характер. При висхідному проектуванні, навпаки, властивості елементів визначаються раніше системи і, отже, вимоги до них мають можливий характер. На практиці звичайно поєднують обидва підходи.

В остаточному вигляді проект являє собою повний комплект конструкторської й технологічної документації, оформленої відповідно до діючих стандартів (ЕСКД, ЕСТД).

1.2. Структура САПР

Системою автоматизованого проектування (САПР) називають організаційно - технічну систему, що складається з комплексу засобів автоматизації проектування, який взаємодіє з підрозділами проектної організації і виконує автоматизоване проектування.

Як будь-яку складну систему, САПР можна подати у вигляді сукупності окремих частин - підсистем. Підсистеми САПР тісно пов'язані з організаційною структурою проектної організації й призначені для виконання певних функцій. Залежно від характеру виконуваних функцій підсистеми діляться на ті, що проектують, і ті, що обслуговують. До тих, що проектують, відносяться підсистеми, що виконують проектні процедури і операції. Вони орієнтовані на виконання певного етапу (стадії) проектування або групи безпосередньо зв'язаних проектних завдань. Прикладами можуть служити підсистеми проектування деталей і складальних одиниць, проектування технологічних процесів механічної обробки.

До обслуговуючих відносяться підсистеми, які забезпечують працездатність підсистем, що проектують. Прикладами обслуговуючих підсистем є підсистема інформаційного пошуку, підсистема документування, підсистема графічного введення - виведення.

Підсистеми, що проектують, діляться на проектно - залежні (об'єктні) і проектно - незалежні (інваріантні). Проектно - залежними називають підсистеми, які реалізують проектні процедури для конкретного об'єкта автоматизованого проектування. До проектно - незалежних відносяться підсистеми, що виконують функції керування і обробки інформації незалежно від особливостей об'єкта, проект якого розробляється.

Компоненти різних підсистем САПР, об'єднані за функціональними ознаками, створюють такі засоби забезпечення: технічне, математичне, програмне, інформаційне, лінгвістичне, методичне, організаційне.

Технічне забезпечення САПР являє собою комплекс технічних засобів, призначених для розв'язання задач автоматизованого проектування.

Технічні засоби в САПР вирішують такі задачі [2]: введення вихідних даних опису об'єкта проектування; відображення введеної інформації з метою її контролю і редагування; перетворення інформації (зміна форми подання даних, трансляції, виконання арифметичних і логічних операцій і т.п.); зберігання інформації; відображення підсумкових і проміжних результатів розв'язання; оперативного спілкування проектувальника із системою.

Компоненти технічного забезпечення САПР включають пристрої обчислювальної й організаційної техніки, засоби передачі даних, множині, вимірювальні й інші пристрої, що забезпечують функціонування САПР.

Математичне забезпечення САПР [1] включає методи й засоби, що дозволяють будувати математичні моделі об'єктів конкретної САПР, вирішувати задачі їхнього оптимального проектування й т.п. В математичне забезпечення входять елементи, що використовуються в різних САПР. До них належать методи розв'язання алгебраїчних рівнянь, чисельні методи розв'язання диференціальних рівнянь, пошук екстремуму, а також алгоритми, на основі яких розробляється програмне забезпечення САПР. Крім того, в математичне забезпечення входять специфічні математичні моделі, характерні для даної предметної галузі.

Програмне забезпечення САПР являє собою сукупність програм і програмної документації, необхідних для автоматизованого проектування.

Програмне забезпечення поділяється на загальносистемне й спеціальне (прикладне).

У загальносистемне програмне забезпечення входять програми, необхідні для організації функціонування технічних засобів. Загальносистемне програмне забезпечення пов'язане з технічними засобами й не відображає специфіки САПР.

Спеціальне (прикладне) програмне забезпечення включає програми, необхідні для виконання проектних процедур, і звичайно створюється у вигляді пакетів прикладних програм. Кожен пакет прикладних програм використовується на певному етапі процесу проектування або обслуговує групу однотипних завдань на різних етапах проектування.

Інформаційне забезпечення САПР становлять різні дані, необхідні для виконання автоматизованого проектування. В інформаційне забезпечення входять документи з описом стандартних процедур, типових проектних рішень, типових елементів, матеріалів.

Основу інформаційного забезпечення САПР становлять банки даних. Банк даних складається з бази даних і системи керування базою даних. База даних - це самі дані, розміщені на машинних носіях. Дані можуть бути подані як у вигляді алфавітно-цифрової, так й у вигляді графічної інформації. Система управління базою даних (СУБД) являє собою сукупність програмних засобів, що забезпечують функціонування банку даних. За допомогою СУБД відбувається пошук необхідних даних за запитом користувача, введення нових даних, видалення невикористовуваних даних, захист від перекручування й несанкціонованого доступу й т.п.

Лінгвістичне забезпечення САПР становлять мови програмування, призначені для опису процедур автоматизованого проектування й проектних рішень.

Методичне забезпечення САПР включає документи, що визначають порядок експлуатації системи автоматизованого проектування. Документи методичного забезпечення мають в основному інструктивний характер.

Організаційне забезпечення САПР являє собою сукупність положень, інструкцій, наказів, документів, що регламентують організаційну структуру підрозділів (у тому числі, штатний розклад), кваліфікаційні вимоги й інші документи, що визначають взаємодію підрозділів проектної організації з комплексом засобів автоматизованого проектування.

1.3. Класифікація САПР

Класифікація САПР, що проектується, дає можливість оцінити її науково - технічний рівень, тривалість розроблення, витрат матеріальних, трудових, фінансових й інших ресурсів.

Класифікація САПР визначається стандартом.

Стандарт регламентує такі ознаки класифікації САПР: тип об'єкта проектування; різновид об'єктів проектування; складність об'єктів проектування; рівень автоматизації проектування; комплексність автоматизації проектування; характер проектних документів, що випускають; число рівнів у структурі технічного забезпечення.

За типом об'єкта проектування САПР діляться на 9 груп: 1) САПР виробів машинобудування; 2) САПР виробів приладобудування; 3) САПР технологічних процесів у машинобудуванні і приладобудуванні; 4) САПР об'єктів будівництва; 5) САПР технологічних процесів у будівництві; 6) САПР програмних виробів; 7) САПР організаційних систем. Інші групи (8 і 9) - резервні.

Класифікація за різновидом об'єкта проектування стандартами не регламентується. Потрібна вказівка й кодування документів відповідно до діючої в галузі системи позначень.

Складність об'єктів проектування оцінюється числом складових частин. Виділяють САПР: 1) простих об'єктів із числом складових частин до 10^2 ; 2) об'єктів середньої складності ($10^2 \div 10^3$); 3) складних об'єктів ($10^3 \div 10^4$); 4) дуже складних об'єктів ($10^4 \div 10^6$); 5) об'єктів дуже високої складності (число складових частин понад 10^6). Складовою частиною виробу, що проектується, є деталь, складовою частиною технологічного процесу - елементарна технологічна операція.

За рівнем автоматизації системи ділять на три групи. Виділяють системи проектування: 1) низькоавтоматизованого проектування (число автоматизованих проектних процедур до 25%); 2) середньоавтоматизованого проектування (25% \div 50%); 3) високоавтоматизованого проектування (понад 50%).

Класифікацію за комплексністю автоматизованого проектування виконують за числом автоматизованих етапів проектування. Розрізняють САПР: 1) одноетапні; 2) багатоетапні; 3) комплексні. В одноетапній САПР автоматизований один етап проектування, у комплексній САПР - всі етапи.

Залежно від характеру проектних документів, що випускають, встановлено 5 класифікаційних груп. Розрізняють САПР, що випускають документи: 1) на паперових стрічках або

аркушах; 2) на магнітних носіях; 3) на фотоносіях (мікрофільми, мікрофіші); 4) комбіновані. П'ята група - резервна.

За числом проектних документів, що випускають, САПР ділять на системи малої продуктивності (до 10^3 документів у рік у перерахуванні на формат А4), середньої продуктивності ($10^3 - 10^4$) і високої продуктивності ($10^5 - 10^6$).

2. МАТЕМАТИЧНІ МОДЕЛІ ТЕХНІЧНИХ ОБ'ЄКТІВ

2.1. Вимоги, які пред'являються до математичних моделей

Вивчення різних процесів за допомогою моделей називається моделюванням. Моделювання дозволяє імітувати роботу системи, приймати попереднє рішення про вибір її характеристик, фактично не маючи реального об'єкта.

Поширені два способи моделювання: фізичне й математичне.

При фізичному моделюванні використовуються фізичні моделі. Між реальним об'єктом і фізичною моделлю повинні бути встановлені співвідношення подібності (геометричної, силової й т.п.).

При математичному моделюванні процеси в об'єкті, який досліджується, його моделі мають різний зміст, але описуються однаковими математичними співвідношеннями. Математична модель - це сукупність математичних залежностей, що приблизно описують процес, який досліджується. При побудові математичних моделей використовуються закони фундаментальної науки: фізики, хімії, економіки й ін.

У процесі побудови моделі системи виконуються такі операції: 1) система розбивається на елементи; 2) встановлюються фізичні закони, на основі яких функціонує система; 3) встановлюється зв'язок між окремими елементами системи.

Найчастіше використовується метод прямої аналогії, що базується на наступних принципах.

1. Система, що моделюється, розбивається на окремі підсистеми. Всі підсистеми функціонують на основі одного фізичного принципу.

2. Стан кожної підсистеми характеризується безліччю фазових змінних. Фазові змінні - величини, що характеризують стан об'єкта. Вони також називаються змінними стану. У якості незалежних змінних можуть виступати час, частота, просторові координати й т.п. Фазові змінні є залежними змінними. Прикладами фазових змінних є швидкість, сила, напруги, деформації й т.п.

3. Структура кожної підсистеми може бути подана великою кількістю елементів і зв'язків їх один з одним.

4. Властивості елемента визначаються його математичною моделлю, що і виражає взаємозалежність між фазовими змінними. Вирази, які визначають таку взаємозалежність, називаються компонентними рівняннями.

5. Зв'язок елементів один з одним визначається топологічними рівняннями. Сукупність топологічних і компонентних рівнянь утворить математичну модель.

До математичних моделей пред'являється ряд вимог [3].

Універсальність математичної моделі визначає ступінь повноти властивостей реального об'єкта. Залежно від призначення математична модель відбиває лише деякі властивості. Наприклад, математична модель, призначена для дослідження коливань вагона, не відбиває його теплотехнічних властивостей. Однак така модель повинна давати можливість проведення аналізу більш-менш численної групи однотипних об'єктів.

Точність математичної моделі відбиває ступінь збігу значень параметрів реального об'єкта з параметрами, отриманими при використанні математичної моделі. Кількісне оцінення ступеня збігу результатів викликає утруднення з наступних причин. По-перше, реальний об'єкт і його модель, як правило, характеризується деяким числом параметрів. У результаті виникає необхідність подання векторної оцінки точності деяким скалярним критерієм. По-друге, модель може багаторазово використовуватися для аналізу різних варіантів об'єкта. Характер прояву властивостей об'єкта може залежати від особливостей взаємозв'язку об'єкта із зовнішнім середовищем. Тому показник точності математичної моделі буде залежати від конкретних умов

функціонування. По-третє, параметри реального об'єкта визначаються в результаті експериментів. Похибка експерименту може бути порівнянна з похибкою математичної моделі.

Нехай об'єкт характеризується m вихідними параметрами. Позначимо параметри реального об'єкта $y_j, j = 1, 2, \dots, m$, а значення тих же параметрів, отриманих за допомогою моделі, - y_{mj} . Введемо вектор відносних похибок $\varepsilon = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_m$ де $\varepsilon_j = (y_j - y_{mj})/y_j$. Для оцінки точності можна використати m -норму (максимальний за абсолютним значенням елемент вектора)

$$\varepsilon_{\max} = \max_{j \in [1:m]} |\varepsilon_j|.$$

Можна використати l -норму, тобто

$$\varepsilon_{\text{ср}} = \sqrt{\sum_{j=1}^m \varepsilon_j^2}.$$

Адекватність математичної моделі являє здатність моделі відображати властивості реального об'єкта із заданою точністю. Як правило, адекватність моделі має місце лише в обмеженій галузі зміни зовнішніх змінних.

Економічність математичної моделі оцінюється насамперед витратами обчислювальних ресурсів (витратами машинного часу, пам'яті).

2.2. Класифікація математичних моделей

Математичні моделі розрізняються за рядом ознак [1].

За характером властивостей, що відображаються, моделі діляться на *функціональні* й *структурні*.

Функціональні моделі використовуються для відображення процесів функціонування об'єкта. Вони звичайно мають форму системи рівнянь. Залежно від фізичної природи об'єкта моделі можуть бути механічними, електричними, тепловими й ін.

Структурні математичні моделі призначені для відображення структурних властивостей і діляться на топологічні і геометричні моделі.

Топологічна модель відображає тільки склад елементів і способи з'єднання елементів один з одним.

На рис. 2.1 показані схеми двох конструкцій. Для конструкції, показаної на рис. 2.1, а, можна побудувати топологічну модель у вигляді списку стержнів і вказівки вузлів, за допомогою яких утворюється стержень. Наприклад, стержень *a* з'єднує вузли 1 й 2.

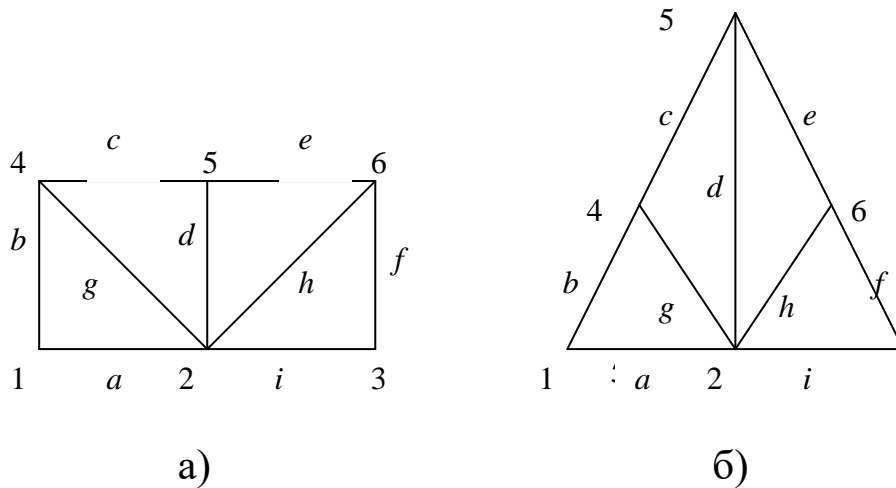


Рис. 2.1

Неважно помітити, що топологічна модель для конструкцій *a* й *б* одна і та ж.

Топологічна модель може бути наведена у вигляді графа, таблиці, матриці, списків і т.п.

Геометрична математична модель, крім відомостей про структуру об'єкта, містить додаткові відомості про форму елементів.

Розрізняють моделі елементів і моделі систем.

Нехай вектор \mathbf{W} , елементами якого є змінні w_k , характеризує зв'язок об'єкта із зовнішнім середовищем, а вектор \mathbf{U} , елементами якого є u_p , характеризує внутрішній стан об'єкта. Частина змінних w_k , що відображають зовнішні впливи, можна об'єднати у вектор вхідних змінних $\mathbf{W}_{\text{вх}}$. Інша частина змінних w_k визначається характером процесів в об'єкті й може бути об'єднана у вектор $\mathbf{W}_{\text{вих}}$. Вектор вихідних змінних $\mathbf{W}_{\text{вих}}$ залежить як від $\mathbf{W}_{\text{вх}}$, так і від \mathbf{U} . Цю залежність можна навести в явній формі

$$\mathbf{W}_{\text{вих}} = \mathbf{F}_1(\mathbf{W}_{\text{вх}}, \mathbf{U})$$

або у вигляді системи алгебраїчних і трансцендентних рівнянь

$$F_2(W_{\text{вх}}, W_{\text{вих}}, U) = 0.$$

Така модель називається *статичною* моделлю об'єкта.

Модель може включати похідні змінних за часом:

$$F_3(W, d/dt, U, d/dt) = 0.$$

Тоді модель об'єкта являє систему диференціальних рівнянь і називається *динамічною* моделлю.

Якщо змінні можуть набувати тільки окремих, ізольованих значень, то математична модель називається *дискретною моделлю*.

У випадку використання безперервних змінних модель називається *безперервною моделлю*.

При проектуванні складних систем використовується блочно-ієрархічний підхід. На кожному рівні використовується своє поняття системи й елемента системи.

Залежно від ступеня деталізації математичні моделі діляться на моделі, створені на мікро-, макро- і метарівні.

На мікрорівні модель відбиває фізичні процеси, що протікають у безперервному суцільному просторі й часі. У цьому випадку модель описується диференціальними рівняннями в часткових похідних. Такі моделі використовуються для розрахунку полів механічних напруг і деформацій, електричних потенціалів, тисків, температур і т.п. На практиці такі моделі мають обмежене застосування внаслідок складності розв'язання систем диференціальних рівнянь.

На макрорівні використовують дискретизацію простору. Це приводить до переходу від розподіленої до зосередженої моделі. У цьому випадку модель описується системою звичайних диференціальних рівнянь.

При великій кількості елементів і відповідно порядку системи рівнянь виникають труднощі розв'язання задач на основі моделей макрорівня й здійснюється перехід до наступного ієрархічного рівня.

На метарівні системи розглядаються як складні пристрої й комплекси. Тут можуть використатися різноманітні математичні моделі. На метарівні в більшості випадків математичні моделі представляються системами звичайних диференціальних рівнянь.

2.3. Типові математичні моделі технічних систем

Для багатьох систем різної фізичної природи форма топологічних рівнянь однакова. Таку подібність рівнянь називають аналогією.

Розглянемо приклади компонентних і топологічних рівнянь.

Електричні системи. Фазовими змінними в електричних системах є струми й напруги.

Компонентні рівняння елементів мають такий вигляд:

$$U=RI; I=C(d/dt); U=L(d/dt),$$

де I - струм;

U – напруга, яка дорівнює різниці потенціалів на кінцях (полюсах) елемента;

R - опір;

C - ємність;

L - індуктивність.

Елементи, з'єднані між собою, утворять структуру, у якій є розгалужені (двополюсні елементи) і вузли. У системі можуть утворюватися контури.

Топологічні рівняння для вузлів і контурів системи мають такий вигляд:

$$\sum_k I_k = 0;$$

$$\sum_i U_i = 0.$$

Ці рівняння називаються законами Кірхгофа. Перше рівняння виражає перший закон Кірхгофа: сума струмів у вузлі

дорівнює нулю. Власне кажучи, це рівняння є рівнянням рівноваги (балансу).

Друге рівняння виражає другий закон Кірхгофа. Воно записується для контурів і може бути названо рівнянням сумісності.

Механічні поступальні системи - це системи, у яких реалізується поступальний рух. Елементами такої системи є:

елементи маси, які характеризують інерційні властивості;
елементи гнучкості, що характеризують пружні властивості;
елементи механічного опору, що характеризують втрати механічної енергії.

У механічній системі ці властивості можуть бути безперервно розподілені по об'єму, тобто кожна ділянка механічної системи може одночасно мати ці властивості.

У якості фазових змінних можуть використовуватися або сили й швидкості, або сили й переміщення.

Компонентним рівнянням маси є рівняння другого закону Ньютона

$$P = d(m)/dt,$$

де m – маса елементарної ділянки;

P - сила;

V - швидкість.

Елементами гнучкості можуть бути, наприклад, пружини, стержні. Якщо на такий елемент діють тільки повздовжні сили й інтерес являють тільки повздовжні переміщення, то компонентні рівняння елементів гнучкості одержують на основі закону Гука

$$\sigma = E (\Delta l / l),$$

де σ - механічна напруга, що діє в напрямку повздовжньої осі елемента;

l - довжина елемента;

Δl - зміна довжини елемента (деформація) під впливом напруги σ ;
 $\Delta l / l$ - відносна деформація;
 E - модуль пружності (модуль Юнга).

Напругу можна навести в такому вигляді:

$$\sigma = P / F,$$

де P - сила,
 F - площа поперечного перерізу розглянутої ділянки.

Після диференціювання за часом отримаємо

$$d/dt = (EF / l)(dl / dt)$$

або

$$\lambda_m(d/dt) = V,$$

де $\lambda_m = l/(EF)$ – гнучкість;
 $V = dl/dt$ - швидкість.

Величина $1 / L_m$, зворотна гнучкості, називається твердістю пружного елемента.

Величина λ_m є аналогом індуктивності в електричних системах.

Якщо сила тертя залежить від швидкості, з'являються елементи механічного опору R . Вони є аналогами електричного опору.

Топологічні рівняння механічної поступальної системи мають такий вигляд:

$$\sum_k P_k = 0;$$

$$\sum_i V_i = 0.$$

Перше рівняння є рівнянням рівноваги сил, що діють на розглянуте тіло. Це рівняння виражає принцип Даламбера і є аналогом першого закону Кірхгофа.

Друге топологічне рівняння виражає принцип складання швидкостей. Відповідно до цього принципу сума абсолютної, відносної й переносної швидкостей дорівнює нулю. Це рівняння є аналогом другого закону Кірхгофа.

Механічні пружні системи являють собою окремих випадок механічної поступальної системи.

Такими системами є різноманітні спорудження, проектування яких здійснюється методами будівельної механіки. Основним завданням розрахунку є визначення параметрів конструкцій, що задовольняють вимоги міцності, твердості й стійкості. Вибір параметрів здійснюється на основі розрахунку зусиль і деформацій у конструкції, яка проектується.

Компонентні рівняння елементів пружної механічної системи є рівняння, що зв'язують сили й деформації. У випадку лінійного зв'язку між силами й деформаціями компонентні рівняння утворюються на основі закону Гука.

Елементи компонентної моделі отримують шляхом поділу суцільного середовища на ділянки. Компонентні в цьому випадку зв'язують більш ніж дві фазові змінні. Наприклад, при розтяганні (стиску) відносно повздовжнє подовження (стиск) супроводжується його відносним поперечним звуженням (розширенням). Деформація розтягання в напрямку однієї з координатних осей, наприклад осі x ,

$$\Delta l / l = (1/E)(\sigma - \nu \sigma_y - \nu \sigma_z),$$

де $\nu = (\Delta d : d) / (\Delta l : l)$ – коефіцієнт Пуассона;

σ_y й σ_z - напруги, що діють у напрямку осей y й z .

Топологічні рівняння пружної механічної системи включають рівняння рівноваги сил і моментів, а також рівняння спільності деформацій.

Механічні обертальні системи. Така система є аналогом механічної поступальної системи. Наприклад, переміщенню x відповідає кут повороту φ , поступальній швидкості v відповідає кутова швидкість ω , силі P - обертальний момент M . Параметрам

механічної поступальної системи m , λ_m й R відповідають аналогічні параметри механічної обертальної системи: J - момент інерції щодо тієї осі, навколо якої є обертання з кутовою швидкістю ω ; λ_{ep} – обертальна гнучкість; R_{ep} - обертальний опір.

Компонентні рівняння механічної обертальної системи мають вигляд

$$M = J (d\omega / dt);$$

$$\lambda_{ep} (d / dt) = \omega_1 - \omega_2;$$

$$M R_{ep} = \omega_1 - \omega_2,$$

де ω_1 й ω_2 - кутові швидкості кінців елементарної ділянки.

Топологічні рівняння механічної обертальної системи також подаються у вигляді принципу Даламбера для обертальних систем і закону складання швидкостей щодо даної осі:

$$\sum_k M_k = 0; \quad \sum_i \omega_i = 0.$$

Гідравлічні й пневматичні системи. Для гідравлічних систем основними фазовими характеристиками є потік рідини (витрата) g і тиск P . При цьому змінні g й P є аналогами електричного струму й напруги.

Для ділянки трубопроводу можна записати рівняння Нав'є-Стокса в наближеній формі

$$d/dt = -\rho^{-1}(P/l) - 2(a/\rho)V,$$

де V - швидкість;

ρ - щільність;

a - коефіцієнт лінеаризованого в'язкого тертя в трубопроводі;

P – різниця тисків на кінцях розглянутої ділянки;

l – довжина ділянки.

Витрата

$$g = \rho S V,$$

де S - площа поперечного перерізу трубопроводу.

Приймаючи $\rho = \text{const}$, отримаємо

$$dg/dt = (S/l) P - 2(a/\rho)g.$$

Позначимо $\lambda_2 = l/S$ й $R_2 = 2al/(\rho S)$. Тоді

$$P = \lambda_2 (dg/dt) + R_2 g .$$

Якщо $\rho \neq \text{const}$, то

$$P = \lambda_2 (dg/dt) + R_2 g - l(dp/dt) .$$

Для труби круглого поперечного перерізу радіусом r і довжиною l

$$R_2 = 8\nu l / (\pi r^4),$$

де ν - кінематична в'язкість рідини.

Вираз для R_2 і компонентне рівняння для гідравлічного опору справедливі тільки для ламінарного потоку рідини.

Топологічні рівняння гідравлічної системи аналогічні за формою рівнянням розглянутих систем:

$$\sum_k g_k = 0; \quad \sum_i P_i = 0.$$

Рівняння виражають рівність нулю суми потоків у будь-якому вузлі системи і суми тисків уздовж будь-якого контуру системи.

Теплові системи. Основними фазовими змінними цих систем є температура T і тепловий потік g . Як елемент можна розглядати ділянку твердого тіла. Кожна ділянка характеризується середньою теплоємністю C_T . Збільшення кількості теплоти в елементі пов'язане зі збільшенням температури таким співвідношенням:

$$dQ = C_T d.$$

Тоді

$$d/dt = C_T d/dt$$

або

$$g = C_T d/dt.$$

Кожна ділянка твердого тіла має теплопровідність. Відповідно до закону Фур'є щільність теплового потоку пропорційна градієнту температури, причому коефіцієнтом пропорційності є коефіцієнт теплопровідності. Тоді

$$g = (\lambda S/l)(T_1 - T_2) = (T_1 - T_2)/R_T,$$

де S - площа поперечного перерізу;

l - довжина ділянки;

λ - коефіцієнт теплопровідності;

$T_1 - T_2$ - різниця температур.

Величина

$$R_T = l/(\lambda S)$$

являє собою тепловий опір ділянки.

Якщо ділянка твердого тіла контактує з рідким або газовим середовищем, то тепловий опір визначається тепловіддачею через конвекцію. У цьому випадку умова теплообміну підкоряється закону Ньютона

$$J_Q = a_{\text{конв}}(T_1 - T_2),$$

де $a_{\text{конв}}$ - коефіцієнт тепловіддачі через конвекцію;

T_1 - температура рідини (газу) в удалині від твердого тіла;

T_2 - температура, яку мало б тверде тіло на своїй поверхні при відсутності теплообміну.

При цьому тепловий опір

$$R_T = 1 / (S a_{\text{конв}}).$$

Таким чином, має місце аналогія між механічними, гідравлічними, пневматичними, тепловими й електричними системами. Аналогами електричної напруги є тиск, температура, швидкість. Аналогами електричного струму є сили й потоки рідини, газу, теплоти.

Топологічне рівняння рівноваги виступає як аналог першого закону Кірхгофа

$$\sum_{k=1}^n I_k = 0. \quad (2.1)$$

Це рівняння потоків у вузлах з'єднаних елементів.

Топологічне рівняння спільності деформацій виступають як аналог другого закону Кірхгофа

$$\sum_{k=1}^m U_k = 0. \quad (2.2)$$

Це рівняння спільності тисків, температур або швидкостей (переміщень) у будь-якому замкнутому контурі.

У рівняннях (2.1) і (2.2) I_k й U_k - фазові змінні, стосовні k -ї галузі; n - число розгалуженостей, що підходять до розглянутого вузла; m - число галузей у розглянутому контурі.

Системи масового обслуговування. Ці системи призначені для виконання потоку заявок або вимог, що надходять на вхід системи у випадкові моменти часу. Система масового обслуговування складається з деякого числа каналів обслуговування, у якості яких можуть виступати верстати, під'їзні колії, робочі бригади, технологічні агрегати, випробувальні стенди і т.д. Виконання заявок, які надходять, відбувається протягом деякого (випадкового) часу, після чого канал звільняється і готовий прийняти наступну заявку. Послідовність заявок, що надходять на вхід системи, утворить безперервний потік заявок.

Системи масового обслуговування бувають двох типів: системи з відмовами й системи з очікуванням. У системі масового обслуговування з відмовами заявка, що надійшла в той момент, коли всі канали зайняті, одержує відмову й не обслуговується. У системі з очікуванням кожна заявка, що надійшла в систему, коли в ній немає вільних каналів, стає в чергу і очікує, поки не звільниться який-небудь канал обслуговування.

Найважливішим вихідним параметром системи масового обслуговування є продуктивність системи (пропускна здатність) - середнє число заявок, які система може обслужити в одиницю часу. Крім того, до вихідних характеристик системи відносять середній час очікування заявок у черзі, імовірність обслуговування кожної із заявок, середній час простою каналів, середнє число зайнятих каналів.

Модель системи масового обслуговування містить у собі моделі вхідних потоків, пристроїв, черг, пам'яті й вузлів.

Побудова моделі вхідного потоку зводиться до вибору параметрів потоку заявок. Найчастіше використовується найпростіший потік, що володіє властивостями стаціонарності, ординарності й відсутності післядії.

Властивість стаціонарності означає незалежність імовірності P_n надходження певної кількості заявок n протягом заданого інтервалу Δt від розміщення цього інтервалу на осі часу.

Потік називають ординарним у випадку неможливості одночасного надходження двох заявок або більше на вхід системи.

Відсутність післядії означає, що ймовірності різних (непересічних) інтервалів не залежать одна від одної. Число заявок n за час Δt у найпростішому потоці підкоряється закону Пуассона

$$P_n(\Delta t) = \frac{(\lambda \Delta t)^n}{n!} e^{-\lambda \Delta t},$$

де λ - параметр розподілу;

e – основа натурального логарифма.

Параметр розподілу λ називається інтенсивністю (щільністю) потоку. Величина параметра λ визначає середній час τ між моментами надходження заявок $\tau = 1/\lambda$ і дисперсію інтервалів між надходженнями заявок, рівну $D = 1/\lambda$. Значення τ і D використовуються для вироблення значень моментів надходження заявок t_i при моделюванні процесів масового обслуговування на ЕОМ.

Математична модель каналу обслуговування являє собою закон розподілу часу обслуговування. Як математична модель каналу обслуговування найчастіше використовується експонентний закон із щільністю розподілу $P(t) = \mu e^{-\mu t}$, де μ - інтенсивність обслуговування. Середній час обслуговування дорівнює $1/\mu$, а дисперсія часу обслуговування дорівнює $1/\mu^2$.

Модель черги дозволяє описати порядок обслуговування заявок, що надходять на вхід системи. У теорії масового обслуговування цю модель називають дисципліною черги. Найчастіше черга підкоряється правилу, відповідно до якого послідовність обслуговування заявок відповідає порядку, у якому заявки надходять у чергу. Таке правило обслуговування називають ПЕРППО: «першим прийшов - першим обслуговується». Використовуються й інші варіанти дисципліни черги, серед яких можна вказати правило «останнім прийшов - першим обслуговується» (ПОСППО). Можна використовувати дисципліну черги, обумовлену правилом випадкового відбору вимог (СОТ).

Модель пам'яті являє собою алгоритм, за яким для кожної заявки розраховується необхідний обсяг пам'яті, необхідної для зберігання інформації.

Призначенням моделі вузлів є відображення існуючих між елементами системи інформаційних зв'язків. Моделі вузлів необхідні для зміни параметрів заявок при переході від пристрою до пристрою, для розмноження, об'єднання заявок і т.п.

Оскільки вхідний потік заявок, час обслуговування й інші характеристики процесу обслуговування підкоряються статистичним закономірностям, у цілому модель системи масового обслуговування - модель статистична.

3. ЗАДАЧІ АНАЛІЗУ І СИНТЕЗУ В СИСТЕМАХ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ

3.1. Класифікація задач аналізу й синтезу

Розрізняють задачі аналізу й синтезу. Аналіз припускає вивчення властивостей технічних об'єктів. Синтез технічних об'єктів ставить своєю задачею створення нових варіантів. При синтезі створюється опис об'єкта, при аналізі - визначаються властивості й виконується дослідження працездатності за його описом.

Класифікація типових задач аналізу й синтезу показана на рис. 3.1 [2].

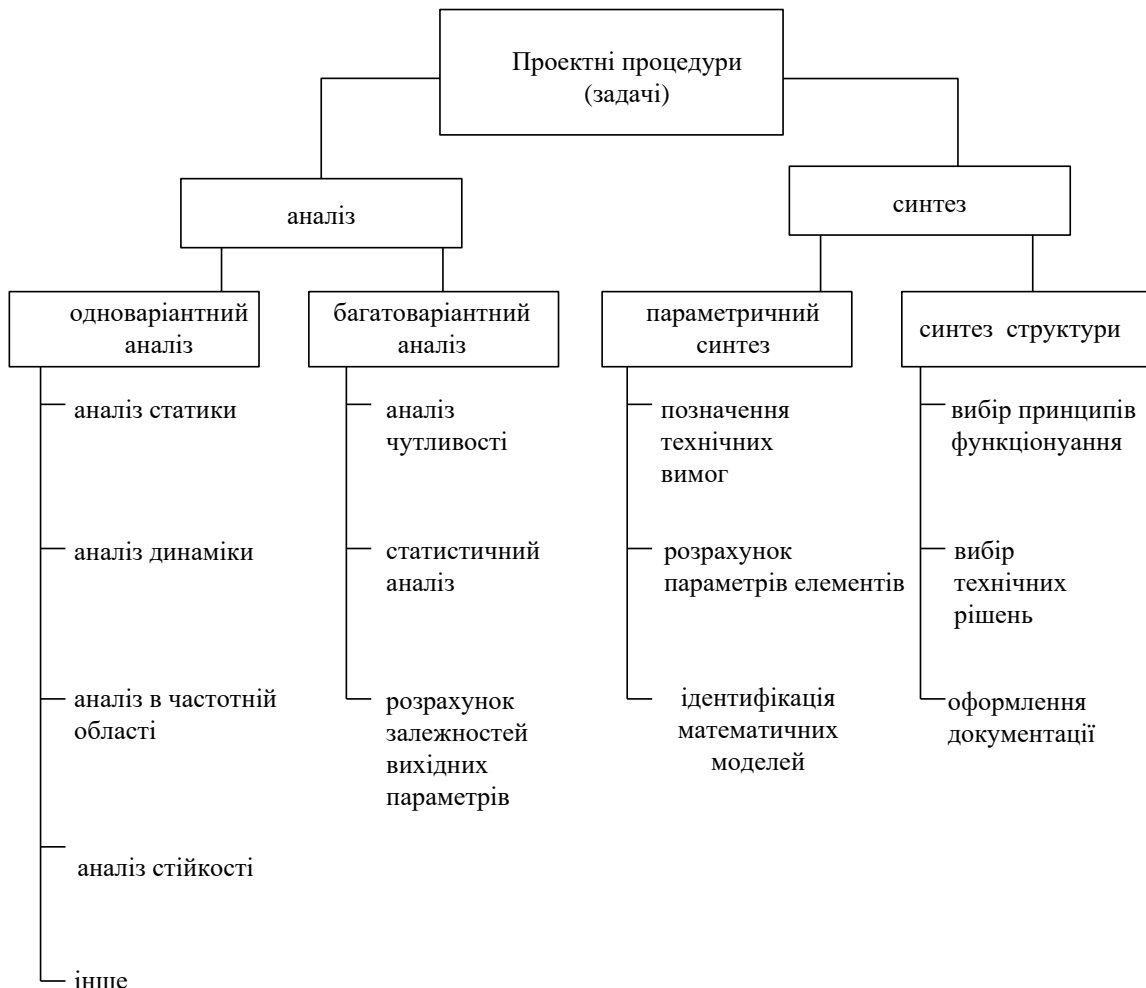


Рис. 3.1

Задача одноваріантного аналізу зводиться до визначення реакції об'єкта на зовнішнє збурювання. Зрозуміло, проектувальника цікавлять параметри об'єкта, при яких об'єкт буде задовольняти задані вимоги. Проте розв'язання задачі аналізу дає проектувальникові винятково важливу інформацію.

При багатоваріантному аналізі виконується дослідження властивостей об'єкта в деякій області простору внутрішніх параметрів.

Задача синтезу поділяється на задачу структурного і параметричного синтезу. Структурний синтез полягає у визначенні структури об'єкта, тобто у визначенні набору елементів і зв'язків між ними.

Метою параметричного синтезу є визначення числових значень параметрів елементів при заданих структурі і умовах працездатності.

3.2. Математична постановка типових задач аналізу

Розв'язання задач аналізу на ЕОМ вимагає застосування чисельних методів. Інженерові-проектувальникові важливо знати ці методи, щоб використовувати програми для ЕОМ, які реалізують той або інший метод розв'язання задачі.

Аналіз статичних станів об'єкта може бути виконаний шляхом інтегрування рівнянь

$$\mathbf{F}(\mathbf{V}) = 0. \quad (3.1)$$

Інший підхід до розв'язання задачі (3.1) заснований на застосуванні ітераційних методів розв'язання.

Можливий також підхід, при якому задача аналізу статички уявляється як екстремальна задача. При цьому можна побудувати цільову функцію, що має мінімум у точці розв'язання задачі \mathbf{V}^* . Такою функцією може бути функція, яка дає суму квадратів відхилів:

$$\varphi(\mathbf{V}) = \sum_{i=1}^n [f_i(\mathbf{V})]^2,$$

де $f_i(\mathbf{V})$ - i - а компонента вектора $\mathbf{F}(\mathbf{V})$;
 n - порядок системи (3.1).

Аналіз динамічних процесів функціонування систем проводиться шляхом розв'язання систем звичайних диференційних рівнянь. У загальному випадку система диференційних рівнянь у неявному вигляді може бути подана так:

$$\mathbf{F}(d/dt, \mathbf{U}, \mathbf{W}, t) = 0, \quad (3.2)$$

де $\mathbf{V} = (\mathbf{U}, \mathbf{W})$ - вектор фазових змінних.

Передбачається, що початкові умови для рівнянь (3.2) задані.

Для розв'язання системи рівнянь (3.2) найчастіше використовуються чисельні методи інтегрування.

Для деяких технічних систем вихідні параметри визначаються на основі аналізу частотних характеристик. Як приклад можна навести системи автоматичного регулювання й керування.

Диференційні рівняння, що описують процеси в таких системах, мають вигляд

$$\mathbf{A}d/dt + \mathbf{B}\mathbf{U} + \mathbf{C}\mathbf{W} + \mathbf{D}\mathbf{U}_{ex}(t) = 0, \quad (3.3)$$

де \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} – матриці коефіцієнтів;

$\mathbf{U}_{ex}(t)$ - вектор-функція зовнішнього вхідного впливу.

Задаючись синусоїдальним вхідним впливом і виконавши для рівняння (4.3) перетворення Фур'є, отримаємо систему рівнянь із комплексними коефіцієнтами:

$$j\omega\mathbf{A}\tilde{\mathbf{U}} + \mathbf{B}\tilde{\mathbf{U}} + \mathbf{C}\tilde{\mathbf{W}} + \mathbf{D} = 0,$$

де $\tilde{\mathbf{U}}$ і $\tilde{\mathbf{W}}$ - функції \mathbf{U} й \mathbf{W} , перетворені за Фур'є.

Аналіз чутливості проводиться з метою з'ясування впливу вхідних параметрів на вихідні параметри системи.

Нехай x_i - вектор внутрішніх і зовнішніх параметрів системи, y_j - вихідні параметри, де $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$...

Чутливість кількісно оцінюють за допомогою матриці \mathbf{A} коефіцієнтів чутливості, елементи якої рівні $a_{ij} = \partial y_j / \partial x_i$. Помітимо, що j - у рядок матриці чутливості \mathbf{A}_j є градієнтом функції $y_j(\mathbf{X})$, тобто $\mathbf{A}_j = \mathbf{grad} y_j(\mathbf{X}) = (\partial y_j / \partial x_1, \dots, \partial y_j / \partial x_2, \dots, \partial y_j / \partial x_n) \dots$

3.3. Задачі структурного синтезу в системах автоматизованого проектування

3.3.1. Формальне подання структури системи

Універсальним засобом наочного подання різноманітних систем служить граф.

Графом називають сукупність вершин, з'єднаних ребрами. Можна описати граф у такому вигляді: $\Gamma = (B, P, I)$, де B – множина вершин (вузлів); P – множина ребер; I – покажчик способів з'єднання ребер один з одним.

Граф, у якому ребра мають певні напрямки, називається спрямованим графом (орієнтованим або орграфом). В орієнтованому графі ребра називаються дугами. Пара вершин може з'єднуватися двома або більше ребрами (дугами). Дуга (або ребро) може починатися й закінчуватися в одній і тій же вершині. Така дуга (або ребро) називається петлею.

Вершини, з'єднані ребром або дугою, називаються суміжними. Ребра, що мають загальну вершину, також називаються суміжними.

У графі, показаному на рис. 3.2,а, множина ребер $P = \{a, b, c, d, e, f, g\}$, а множина вершин $B = \{1, 2, 3, 4, 5\}$.

Частину графа, що включає деякі вузли й ребра графа, називають підграфом. На рис. 3.2, б показаний підграф $\Gamma' = (B', P', I')$, у якому $B' = \{1, 2, 3\}$, $P' = \{a, b, c\}$. Якщо в підграф входять всі вершини графа, тобто $B' = B$, то підграф називають суграфом.

Послідовність ребер, у якій сусідні ребра інцидентні одній і тій же вершині, називається маршрутом. Наприклад, послідовність ребер (a, h, g, e) визначає маршрут (рис. 3.2,а). Маршрут називається ланцюгом, якщо всі його ребра різні, і простим ланцюгом, якщо різні всі його вершини. Замкнутий ланцюг називається циклом. Довжина маршруту дорівнює кількості ребер у порядку їх проходження. Довжина найкоротшого простого ланцюга, що з'єднає вузли U_i й U_j , називається відстанню $d(U_i, U_j)$ між U_i й U_j . Найбільша відстань називається діаметром графа.

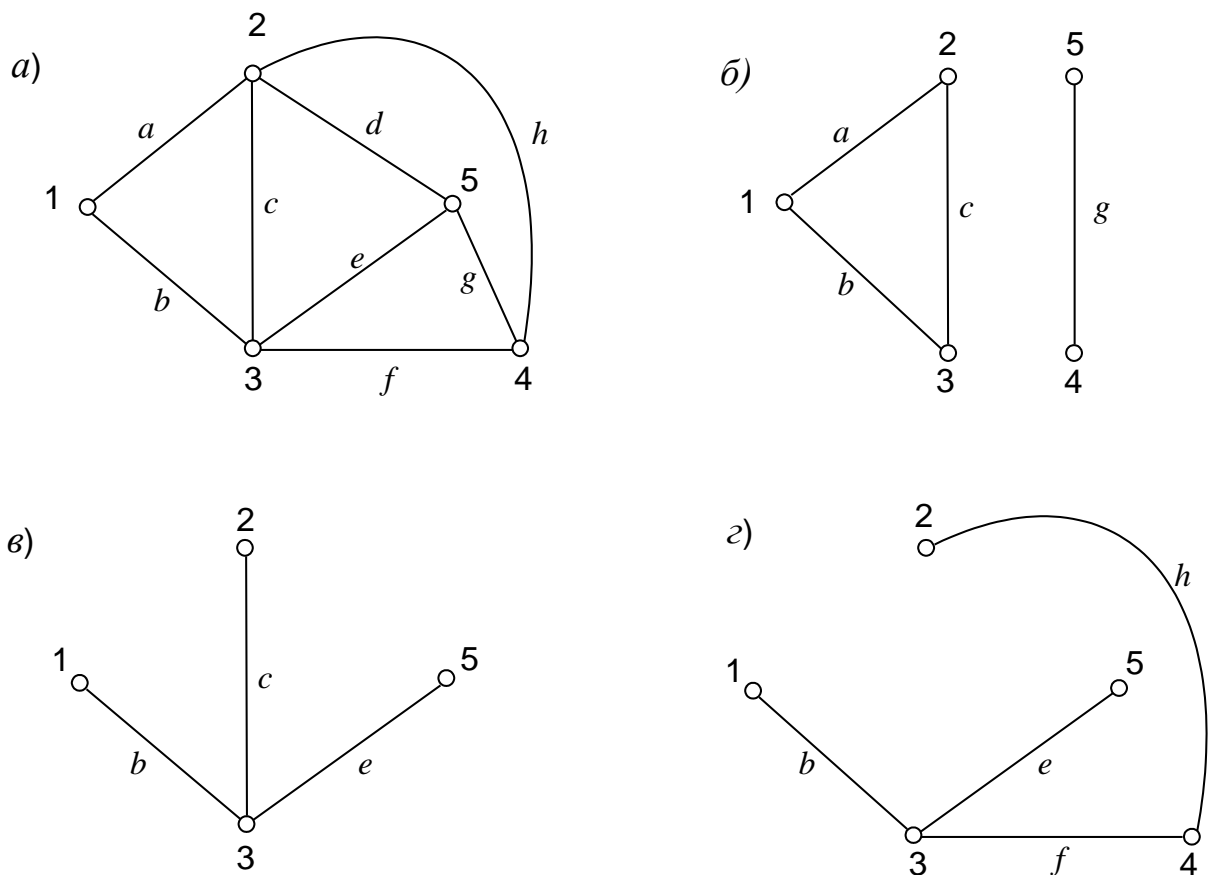


Рис. 3.2

Граф називається зв'язним, якщо будь-яка пара його вершин з'єднана маршрутом. Це можна записати в такій формі: «якщо $\forall x, y \in Y (\exists s(x, y))$, то граф зв'язний». Тут використовується форма запису, прийнята в теорії множин. Цей запис читається так: «якщо для будь-яких вершин x і y , що належать множині Y , існує

маршрут s , то граф зв'язний». Граф, показаний на рис. 3.2,а – зв'язний, граф на рис 3.2, б - незв'язний.

Зв'язний підграф без циклів називається деревом зв'язного графа. На рис. 3.2,в, г наведені приклади дерев для графа рис. 3.2, а.

Покриваючим (фундаментальним) деревом (кістяком) графа називають зв'язний суграф без циклів. Інакше кажучи, дерево є покриваючим (фундаментальним), якщо його розгалуження охоплюють всі вершини й не утворюють при цьому циклів.

Ребра графа, що ввійшли в дерево, називають гілками дерева.

На рис. 3.2,г показане фундаментальне дерево (кісту) графа. Для одного і того ж самого графа можна вибрати кілька фундаментальних дерев.

Крім наочного графічного подання для опису структури графа використовуються матриця інцидентності й матриця суміжності.

Нехай V_1, V_2, \dots, V_n – вершини графа $\Gamma(V, P)$, а P_1, P_2, \dots, P_m – його ребра.

Матрицею суміжності, що відповідає графу Γ , називається матриця R , у якій елемент R_{ij} дорівнює числу ребер (дуг), що з'єднують вершини V_i й V_j (що йдуть із V_i в V_j), $R_{ij} = 0$, якщо відповідні вершини не суміжні. Матриця суміжності – квадратна матриця порядку n , де n - число вершин графа Γ .

У табл. 3.1 наведена матриця суміжності для графа рис. 3.2, а.

Таблиця 3.1

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	1	1
3	1	1	0	0	1
4	0	1	1	0	1
5	0	1	1	1	0

У матриці інцидентності A графа Γ елемент $a_{ij} = 1$, якщо вершина U_i інцидентна ребру P_j , і $a_{ij} = 0$, якщо вершина U_i і ребро P_j не інцидентні.

У табл. 3.2 наведена матриця інцидентності для графа рис. 3.2, *a*.

Таблиця 3.2

Вершини	Ребра							
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
1	1	1	0	0	0	0	0	0
2	1	0	1	1	0	0	0	1
3	0	1	1	0	1	1	0	0
4	0	0	0	0	0	1	1	1
5	0	0	0	1	1	0	1	0

Якщо граф спрямований, то знак одиниці вказує напрямок дуг: +1 відповідає рядку вузла, до якого спрямоване ребро, -1 рядку другого вузла. Всі інші елементи матриці приймають рівними нулю.

Графи використовуються для відображення структури при проектуванні систем. Вершини графа відповідають елементам системи, а ребра відображають зв'язок елементів одного з одним. Ребрам можуть ставитися у відповідність значення параметрів, які називаються вагою. Як вага може використовуватися, наприклад, довжина. У цьому випадку граф називають зваженим графом.

Використовується також інший підхід, при якому ребрами зображують елементи, а вершинами - зв'язки між елементами.

Дерева зв'язного графа зручно використати для зображення ієрархічної структури системи (рис. 3.3). Вершини дерева звичайно ділять на рівні (яруси). На рис. 3.3 рівні позначені римськими цифрами I - IV.

Єдину вершину верхнього рівня називають кореневою вершиною (коренем дерева). Кінцеві (висячі) вершини дерева називають листами.

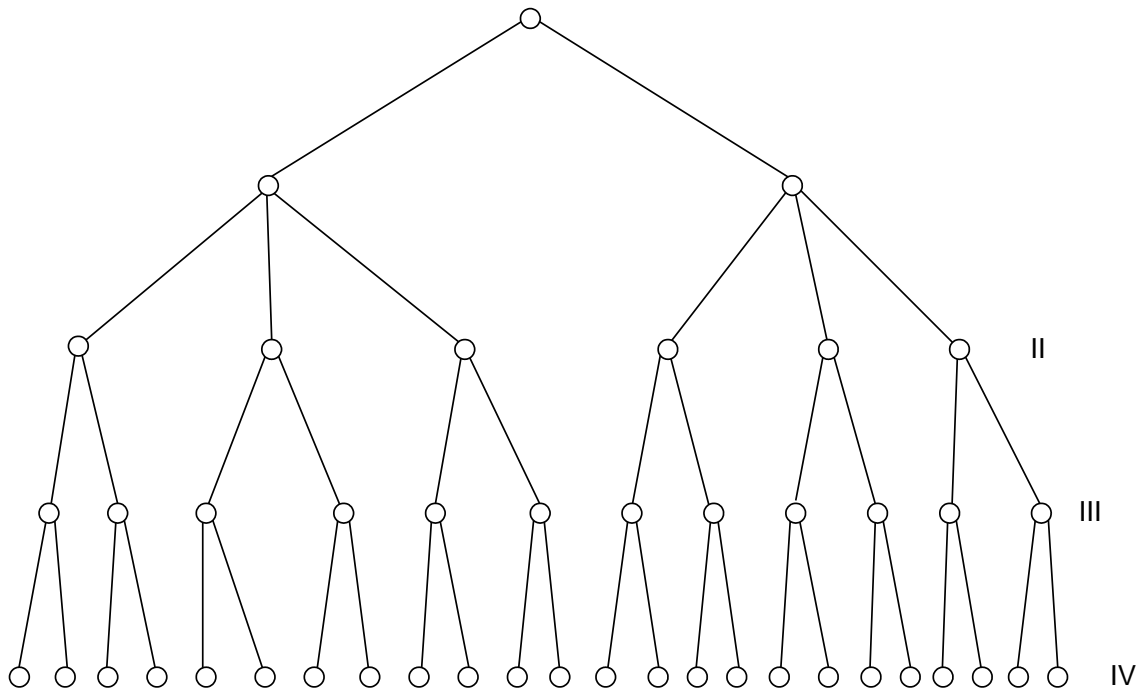


Рис. 3.3

3.3.2. Методи й алгоритми структурного синтезу

Використовується кілька підходів до розв'язання задачі структурного синтезу: 1) перебір; 2) послідовний синтез; 3) трансформація описів різних аспектів.

Класифікація алгоритмів структурного синтезу наведена на рис. 3.4.

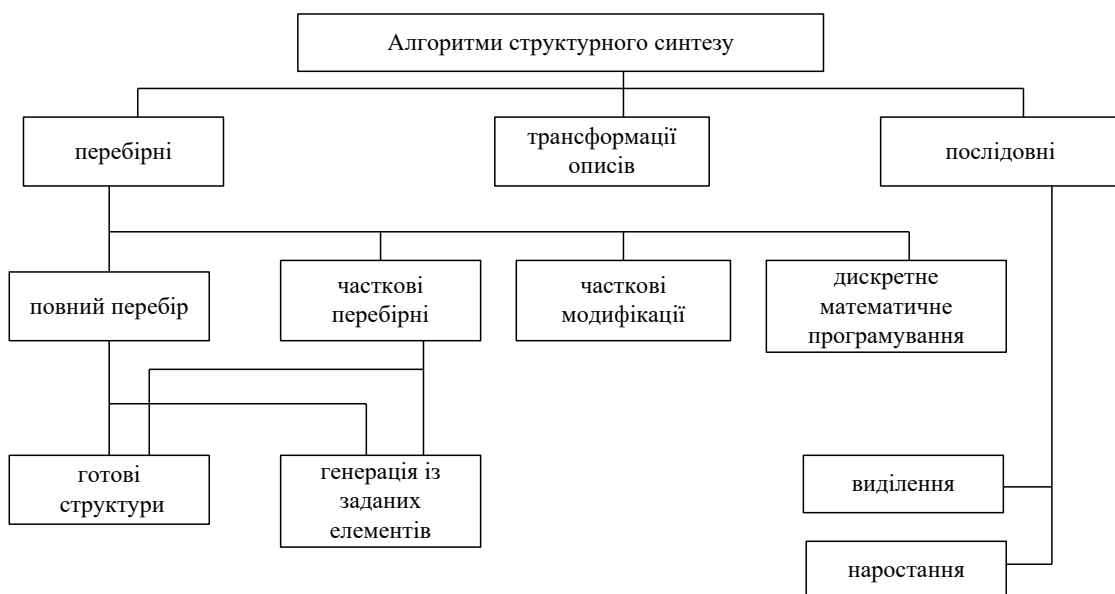


Рис. 3.4

Алгоритми перебору ґрунтуються на оцінці варіантів готових структур. Варіанти структур створюються заздалегідь і зберігаються в базі даних або генеруються із заданого набору елементів.

При використанні перебірних алгоритмів виконуються такі операції:

- 1) вибір або генерація чергового варіанта;
- 2) оцінення варіанта;
- 3) прийняття рішення про подальші дії.

Повний перебір. Для знаходження оптимального розв'язання цілочислової задачі застосовують методи, у яких враховується те, що число можливих розв'язань будь-якій цілочислової задачі є кінцевим. Так, у задачі з змінними x_1 і x_2 , що задовольняють граничні умови $0 \leq x_1 \leq 5$, $0 \leq x_2 \leq 6$, число можливих розв'язань $N=5 \cdot 6=30$. Можна розглянути всі можливі сполучення x_1 і x_2 , вибрати з них ті, які задовольняють обмеження, і з їх числа вибрати найкраще в значенні прийнятого критерію розв'язання.

Такий метод є дуже трудомістким. Зі збільшенням числа змінних трудомісткість різко зростає.

Метод гілок і меж. У методі гілок і меж не розглядаються всі можливі альтернативи.

Розглянемо метод гілок і меж на такому прикладі:

$$F(x_1, x_2) = 6x_1 + 4x_2 \rightarrow \max;$$

$$5x_1 + 2x_2 \leq 23;$$

$$6x_1 + 4x_2 \leq 38;$$

$$x_1, x_2 - \text{цілі.}$$

Розв'яжемо задачу без накладення умов цілочисельності. Назвемо її задачею 1.

У результаті розв'язання цієї задачі як безперервної знайдемо оптимальне розв'язання $x_1^1 = 2$; $x_2^1 = 6.5$; $F_1 = 38$, де верхній індекс у змінних і нижній індекс у цільовій функції відповідає номеру задачі. У цьому розв'язанні значення змінної не задовольняє вимоги цілочисельності.

Складемо дві задачі із граничними умовами, що виключають можливість одержання $x_2 = 6.5$.

Задача 2

$$F(x_1, x_2) = 6x_1 + 4x_2 \rightarrow \max;$$

$$5x_1 + 2x_2 \leq 23;$$

$$6x_1 + 4x_2 \leq 38;$$

$$x_1 \geq 0; \quad x_2 \leq 6$$

Задача 3

$$F(x_1, x_2) = 6x_1 + 4x_2 \rightarrow \max;$$

$$5x_1 + 2x_2 \leq 23;$$

$$6x_1 + 4x_2 \leq 38;$$

$$x_1 \geq 0; \quad x_2 \geq 7$$

У результаті розв'язання цих задач отримані такі значення:

$$x_1^2 = 2.2; \quad x_2^2 = 6; \quad F = 37.2; \quad x_1^3 = 1.666; \quad x_2^3 = 7; \quad F = 38.$$

При розв'язанні наступних задач (рис. 3.5) накладаються вимоги на граничну умову $x_1^2 = 2.2$; $x_1^3 = 1.666$.

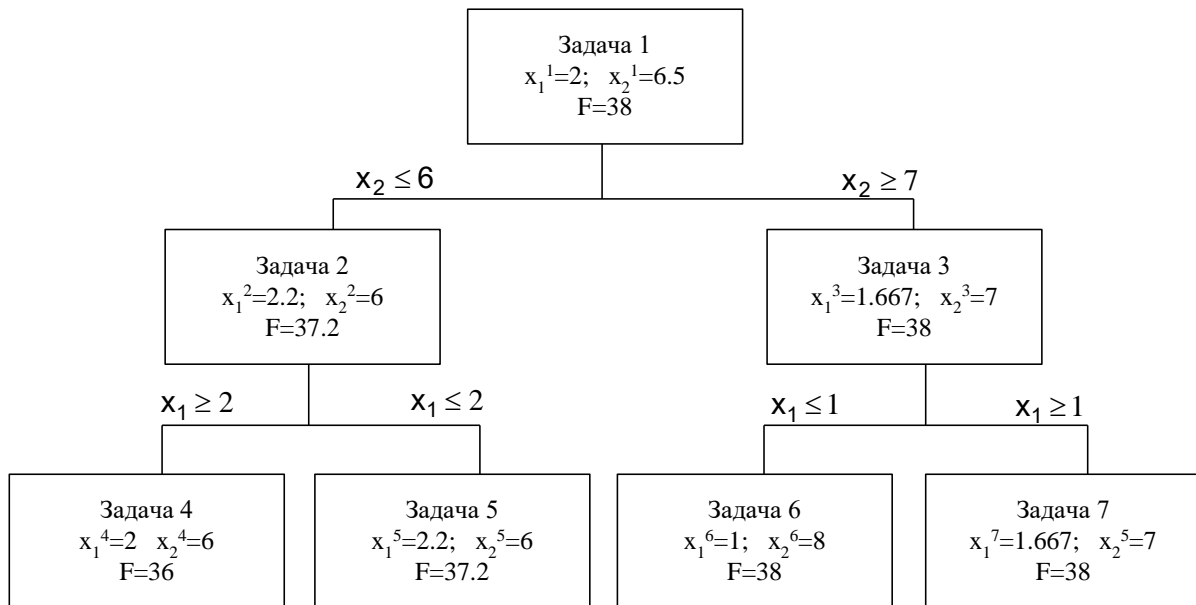


Рис. 3.5

Результати розв'язань наступних задач наведені в табл. 3.3.

Таблиця 3.3

Задача	x_1	x_2	F
1	2	6.5	38
2	2.2	6	37.2
3	1.667	7	38
4	2	6	36
5	2.2	6	37.2
6	1	8	38
7	1.667	7	38

З табл. 3.3 видно, що оптимальним є розв'язання задачі 6. Існує також цілочислове розв'язання задачі 4. Однак воно дає менше значення цільової функції.

Дискретне математичне програмування. Цей алгоритм застосовується в тих випадках, коли задачу синтезу структури можна сформулювати як задачу дискретного математичного програмування.

Приклад 3.1

Нехай об'єкт проектування складається з 3 ланок, а кожна ланка може бути реалізована декількома варіантами (рис. 3.6).



Рис. 3.6

Об'єкт характеризується двома параметрами: P - імовірністю безвідмовної роботи протягом заданого часу й C – вартістю.

Кожний варіант ланок, що входять в об'єкт проектування, характеризується цими ж параметрами. Значення параметрів ланок наведені в табл. 3.4.

Таблиця 3.4

Варіант	Ланка		
	1	2	3
1	$p_{11}=0.95$ $c_{11}=15$	$p_{12}=0.99$ $c_{12}=30$	$p_{13}=0.95$ $c_{13}=15$
2	$p_{21}=0.99$ $c_{21}=30$		$p_{23}=0.99$ $c_{23}=25$
3	$p_{31}=0.90$ $c_{31}=20$		

Параметри об'єкта проектування визначаються параметрами ланок:

$$P = p_1 p_2 p_3;$$

$$C = c_1 + c_2 + c_3.$$

Задача полягає у виборі такого варіанта реалізації кожної ланки, при якому об'єкт проектування має мінімальну вартість за

умови, що ймовірність безвідмовної роботи протягом заданого часу не менш заданого значення, рівного 0.93.

Введемо булеві змінні δ_{ik} , де k - номер ланки, i - номер варіанта.

Приймемо, що

$\delta_{ik}=1$, якщо для k - ї ланки прийнятий i - й варіант;

$\delta_{ik}=0$ у протилежному випадку.

Кожна ланка може бути реалізована тільки одним варіантом:

$$\delta_{11} + \delta_{21} + \delta_{31} = 1;$$

$$\delta_{12} = 1;$$

$$\delta_{13} + \delta_{23} = 1.$$

Вартість кожної ланки залежно від варіанта дорівнює:

$$Z_1 = 15 \delta_{11} + 30 \delta_{21} + 20\delta_{31};$$

$$Z_2 = 30 \delta_{12};$$

$$Z_3 = 15 \delta_{13} + 25 \delta_{23}.$$

Імовірність безвідмовної роботи об'єкта:

$$p_1 = 0.95 \delta_{11} + 0.99 \delta_{21} + 0.90\delta_{31};$$

$$p_2 = 0.99 \delta_{12};$$

$$p_3 = 0.95 \delta_{13} + 0.99 \delta_{23}.$$

Сформулюємо задачу оптимізації:

$$Z = 15 \delta_{11} + 30 \delta_{21} + 20\delta_{31} + 30 \delta_{12} + 15 \delta_{13} + 25 \delta_{23} \rightarrow \min$$

при умовах:

$$(0.95 \delta_{11} + 0.99 \delta_{21} + 0.90\delta_{31}) (0.99 \delta_{12}) (0.95 \delta_{13} + 0.99 \delta_{23}) \geq 0.93;$$

$$\delta_{11} + \delta_{21} + \delta_{31} = 1;$$

$$\delta_{12} = 1;$$

$$\delta_{13} + \delta_{23} = 1;$$

$$\delta_{11} \geq 0; \quad \delta_{11} \leq 1; \quad \delta_{11} - \text{ціле};$$

$$\delta_{21} \geq 0; \quad \delta_{21} \leq 1; \quad \delta_{21} - \text{ціле};$$

$$\delta_{31} \geq 0; \quad \delta_{31} \leq 1; \quad \delta_{31} - \text{ціле};$$

$$\delta_{12} \geq 0; \quad \delta_{12} \leq 1; \quad \delta_{12} - \text{ціле};$$

$$\delta_{13} \geq 0; \quad \delta_{13} \leq 1; \quad \delta_{13} - \text{ціле};$$

$$\delta_{23} \geq 0; \quad \delta_{23} \leq 1; \quad \delta_{23} - \text{ціле}.$$

Це задача нелінійного цілочисельного програмування

Розв'яжемо задачу за допомогою надбудови **Excel**, що називається **Пошук розв'язання**.

У меню **Сервіс** виберемо **Пошук розв'язання**. Якщо команда **Пошук розв'язання** відсутня, то для її встановлення необхідно вибрати команду **Сервіс, Надбудови, Пошук розв'язання**.

На робочій сторінці введемо в комірки A1, A4, A6 необхідні пояснення. Значення змінних будуть перебувати в комірках B3:G3.

В комірку H4 введемо цільову функцію

$$=15*B3 + 30*C3 + 20*D3 + 30*E3 + 15 *F3 + 25 *G3,$$

в комірки A7:A21 введемо ліві частини обмежень

$$=(0.95 *B3 + 0.99 *C3 + 0.90*D3)(0.95 *F3 + 0.99 G3)* + 0.99 E3$$

$$=B3+C3+D3$$

$$=F3+G3$$

$$=B3$$

$$=B3$$

$$=C3$$

$$=C3$$

$$=D3$$

$$=D3$$

=E3
 =E3
 =F3
 =F3
 =G3
 =G3

а в комірки C7:C21 - праві частини обмежень.

Робоча сторінка набуває вигляду, показаного на рис. 3.7.

Виберемо команду **Сервіс, Пошук розв'язання** й заповнимо діалогове вікно **Пошук розв'язання** (рис. 3.7).

The screenshot shows the Microsoft Excel Solver interface. The Solver Parameters dialog box is open, with the following settings:

- Set Objective:** \$G\$4 (Target Value To Of: To Max Of)
- By Changing Variable Cells:** \$B\$3:\$G\$3
- To Change Variable Cells:** Make Unconstrained Variables Non-Negative
- Subject to the Constraints:**
 - \$B\$7:\$B\$21 <= \$C\$7:\$C\$21
 - \$B\$7:\$B\$21 <= 1
 - \$B\$8:\$B\$21 <= 1
 - \$B\$10:\$B\$21 <= 0
 - \$B\$11:\$B\$21 <= 1
 - \$B\$12:\$B\$21 <= 0
 - \$B\$13:\$B\$21 <= 1
 - \$B\$14:\$B\$21 <= 0
 - \$B\$15:\$B\$21 <= 1
 - \$B\$16:\$B\$21 <= 0
 - \$B\$17:\$B\$21 <= 1
 - \$B\$18:\$B\$21 <= 0
 - \$B\$19:\$B\$21 <= 1
 - \$B\$20:\$B\$21 <= 0
 - \$B\$21:\$B\$21 <= 1
- Change Variable Cells:** Make Unconstrained Variables Non-Negative
- Load/Save:** Load Solver Load/Save Defaults Save Solver Load/Save Defaults
- Help:** Solver Help
- Options:** Make Unconstrained Variables Non-Negative Select a Solving Method Select a GRG Nonlinear engine Select an LP Simplex LP engine Select an Evolutionary engine
- Help:** Solver Help
- Options:** Make Unconstrained Variables Non-Negative Select a Solving Method Select a GRG Nonlinear engine Select an LP Simplex LP engine Select an Evolutionary engine

Рис. 3.7

У поле **Встановити цільову комірку** необхідно ввести адресу комірки \$H\$4 з функцією, для якої знаходиться мінімальне значення.

У поле **Рівної** встановити перемикач у положення мінімального значення.

У полі **Змінюючи комірки** вказуються комірки, які повинні змінюватися в процесі пошуку розв'язання. У нашому випадку варто ввести діапазон \$B3:\$G3.

Обмеження, що накладають на змінні задачі, відображаються в полі **Обмеження** (рис 3.8). Засіб пошуку розв'язання допускає обмеження у вигляді рівностей, нерівностей, а також дозволяє ввести вимогу цілочисельності змінних.

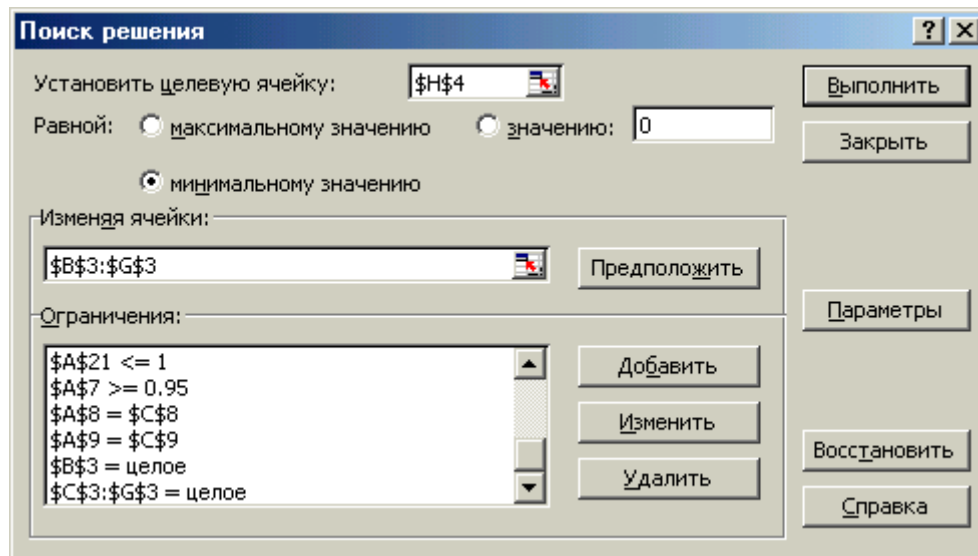


Рис. 3.8

Для уведення обмежень натисніть кнопку **Додати** діалогового вікна **Пошук розв'язання**. Після цього в діалоговому вікні **Додавання обмеження** (рис. 3.9) заповніть поля.

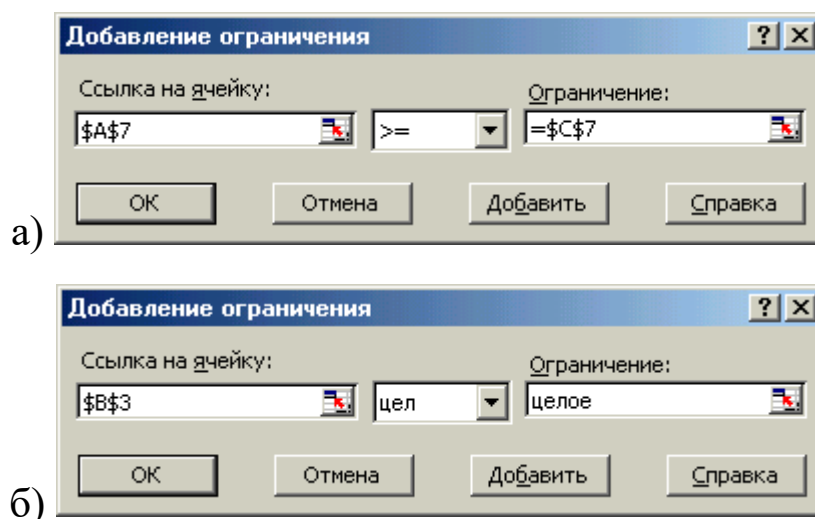


Рис. 3.9

Для введення наступного обмеження натисніть кнопку **Додати** в діалоговому вікні **Додавання обмеження** й заповніть поля.

Натисніть кнопку **Параметри** діалогового вікна **Пошук розв'язання**. У діалоговому вікні, що **відкрилося**, **Параметри пошуку розв'язання** (рис. 3.10) можна змінювати параметри й вибирати метод пошуку розв'язання.

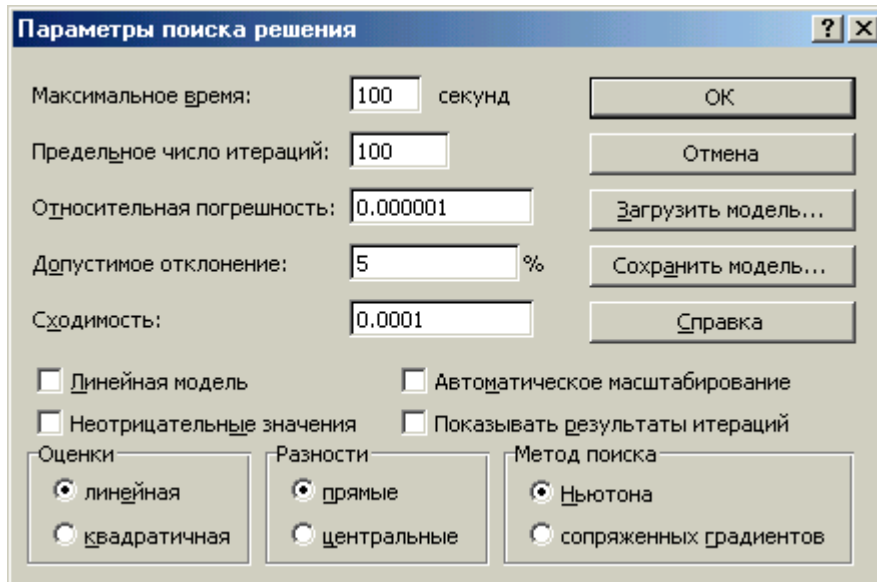


Рис. 3.10

Для нашого прикладу можна залишити параметри, які прийняті за замовчуванням. Натиснемо кнопку **ОК** і закриємо діалогове вікно **Параметри пошуку розв'язання**.

У діалоговому вікні **Пошук розв'язання** натиснемо кнопку **Виконати**. У діалоговому вікні, що **відкрилося**, **Результати пошуку розв'язання** з'явиться повідомлення про результати розв'язання (рис. 3.11).

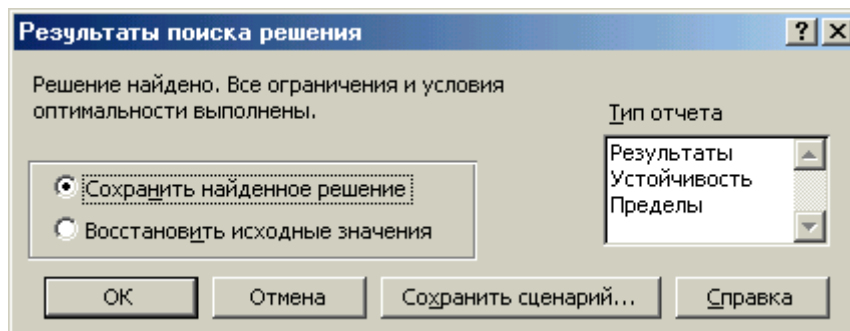


Рис. 3.11

На рис. 3.11 видно, що існують звіти трьох типів: результати, стійкість і межі.

Звіт про результати розв'язання показаний на рис. 3.12.

Целевая ячейка (Минимум)					
Ячейка	Имя	Исходно	Результат		
\$H\$4	Целевая функция	0	85		
Изменяемые ячейки					
Ячейка	Имя	Исходно	Результат		
\$B\$3	d11	0	0		
\$C\$3	d21	0	1		
\$D\$3	d31	0	0		
\$E\$3	d12	0	1		
\$F\$3	d13	0	0		
\$G\$3	d23	0	1		
Ограничения					
Ячейка	Имя	Значение	формула	Статус	Разница
\$A\$7	Ограничения	0.970299	\$A\$7>=0.95	не связан.	0.020299
\$A\$8	Ограничения	1	\$A\$8=\$C\$8	не связан.	0
\$A\$9	Ограничения	1	\$A\$9=\$C\$9	связанное	0
\$A\$10	Ограничения	0	\$A\$10>=0	связанное	0

Рис. 3.12

Зі звіту видно, що в результаті розв'язання отримані такі значення змінних: $\delta_{21} = \delta_{12} = \delta_{23} = 1$; $\delta_{11} = \delta_{31} = \delta_{13} = 0$. Отже, перший блок необхідно виконувати за другим варіантом, третій блок - теж за другим варіантом. Зі звіту видно, що вартість об'єкта $C=85$, імовірність безвідмовної роботи протягом заданого часу $P=0.970299$.

Використовуються також і інші методи і алгоритми структурного синтезу: метод І-АБО дерева, алгоритми послідовного синтезу.

3.3.3. Автоматизація конструкторського й технологічного проектування

На рис. 3.13 наведена класифікація задач конструкторського проектування [5].

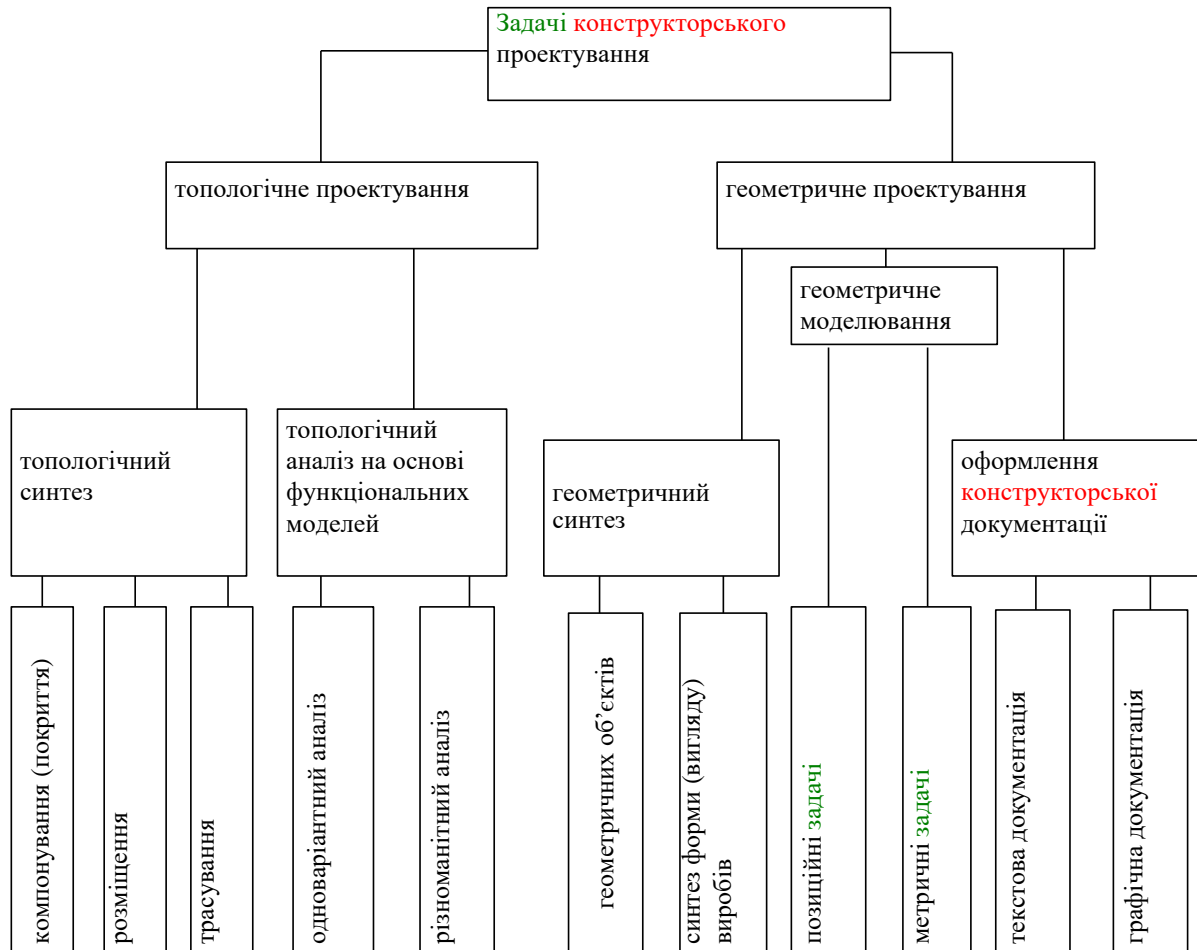


Рис. 3.13

Задачі конструкторського проектування діляться на дві групи: задачі топологічного проектування й задачі геометричного проектування.

Група задач топологічного проектування розглядає синтез структури (топології) конструкції з врахуванням її функціональних характеристик.

Задачі геометричного проектування призначені для визначення суто геометричних параметрів конструкції.

Задачі топологічного проектування. До числа задач топологічного проектування відносяться задачі компоунання, розміщення й трасування.

Задача компоунання конструктивних елементів вищого ієрархічного рівня з елементів нижчого рівня є розповсюдженою задачею проектування. Це найбільш трудомістка частина конструкторського проектування. Спочатку вирішують ескізну частину задачі компоунання, коли за функціональною схемою розробляють загальну конструкцію вузла. Потім на основі ескізного компоунання розробляють робоче компоунання з більш детальним проробленням конструкції вузла.

Задача розміщення полягає у визначенні оптимального просторового розташування об'єктів. Прикладами можуть бути розміщення технологічного устаткування в цеху, розроблення генеральних планів депо і вагоноремонтних заводів.

Задача трасування полягає у визначенні геометрії з'єднання елементів. Прикладами можуть бути задачі про прокладення залізниць, повітропроводів, трубопроводів.

У більшості випадків задачу топологічного проектування можна подати як задачу дискретного математичного програмування.

Геометричне моделювання. Деталі й вузли конструкції можуть мати найрізноманітнішу форму й характеризуватися різними властивостями.

Сукупність властивостей, що однозначно визначають форму геометричного об'єкта, називають геометричною моделлю об'єкта. Геометричну модель можна уявити у вигляді сукупності рівнянь ліній і поверхонь, графіками, списками, таблицями, описаними спеціальними графічними мовами. Теоретичною основою створення геометричних моделей є аналітична геометрія, теорія множин, диференціальна геометрія, теорія графів, алгебра логіки.

При геометричному проектуванні геометричні моделі застосовуються для опису геометричних властивостей об'єкта конструювання (форми, розташування в просторі); розв'язання

геометричних задач (позиційних і метричних); перетворення форми й положення геометричних об'єктів; введення графічної інформації; оформлення конструкторської документації.

До *позиційних задач* геометричного моделювання відносяться визначення приналежності точки замкнутої площини або тривимірної області, визначення перетинання або торкання плоских або об'ємних тіл (деталей) у процесі їхнього руху, оцінення мінімальних або максимальних відстаней і т.п.

До *метричних задач* геометричного моделювання відносяться задачі обчислення площ, об'ємів, маси, моментів інерції, центрів мас і т.п.

Приклад 3.2

Обчислити площу, момент інерції й моменти опору поперечного перерізу хребтової балки піввагона (рис. 3.14).

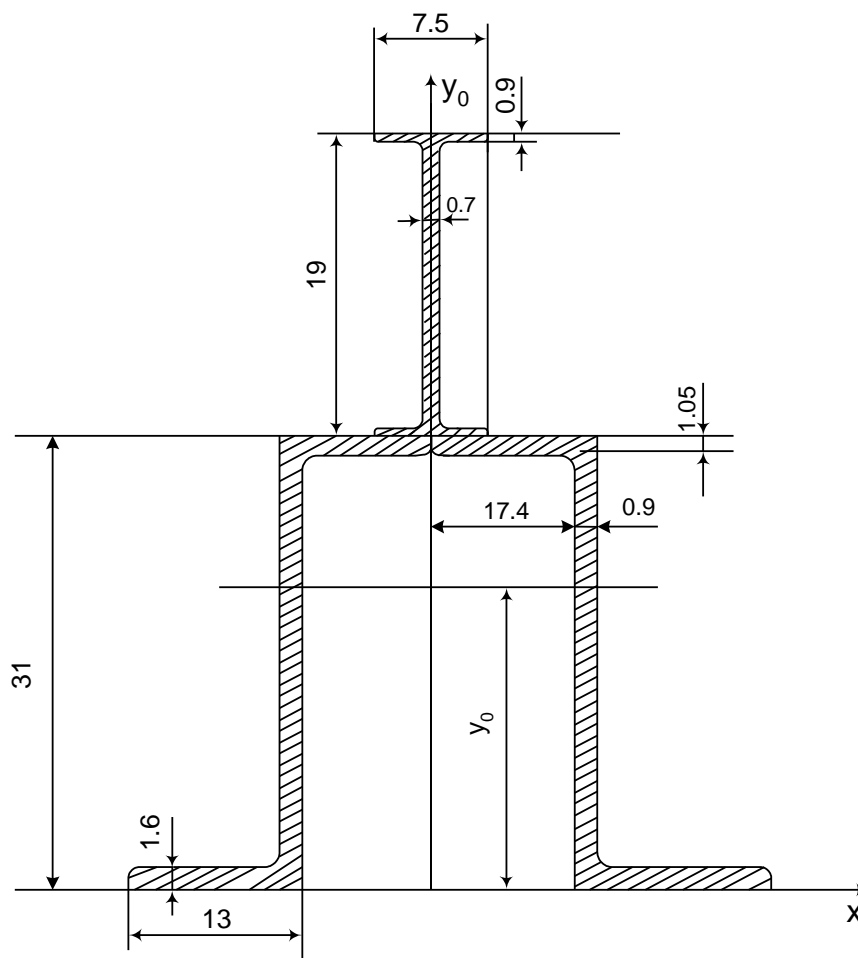


Рис. 3.14

Для визначення моменту інерції розіб'ємо всю фігуру площею F на окремі елементи площею F_1, F_2, \dots, F_n з координатами y_1, y_2, \dots, y_n . Через центр ваги окремих елементів проводять осі, паралельні заданій осі, і обчислюють моменти інерції щодо осі $I_{x1}, I_{x2}, \dots, I_{xn}$.

Момент інерції кожного елемента щодо осі x

$$I_1 = I_{x1} + y_1^2 F_1; I_2 = I_{x2} + y_2^2 F_2; \dots; I_n = I_{xn} + y_n^2 F_n.$$

Момент інерції всієї фігури

$$I_x = I_1 + I_2 + \dots + I_n,$$

тобто

$$I_x = \sum_1^n I_n = \sum_1^n (I_{xn} + y_n^2 F_n) = \sum_1^n I_{xn} + \sum_1^n y_n^2 F_n.$$

Ордината центра ваги

$$y_0 = \frac{\sum F_i y_i}{F}.$$

Момент інерції щодо центральної осі

$$I_{xc} = \sum_1^n I_{xn} + \sum_1^n F_i y_n^2 - y_0^2 F,$$

або



$$I_{xc} = \sum_1^n I_{xn} + \sum_1^n F_i y_n^2 - y_0 \frac{\sum F_i y_i}{F} F.$$

Таким чином,

$$I_{xc} = \sum_1^n I_{xn} + \sum_1^n F_i y_n^2 - y_0 \sum_1^n F_i y_i.$$

Обчислення наведені в табл. 3.5.

Таблиця 3.5

Номер	Елемент перетину	F см ²	y	Fy	Fy ²	I _{x0}
1		132.4	15.3	2025.72	30993.516	21044
2		26.99	40.3	1087.697	43834.1891	1507.5
Σ		159.39		3113.417	74827.7051	22551.5

Ордината центра ваги

$$y_0 = \frac{\sum F_i y_i}{F} = \frac{3113.417}{159.32} = 19.542 \text{ см.}$$

Момент інерції щодо центральної осі

$$I_{xc} = 22551.5 + 74827.705 - 19.542 \cdot 3113.47 = 3.654 \cdot 10^4 \text{ см}^4.$$

Моменти опору рівні

$$W_x^e = 1201 \text{ см}^3; \quad W_x^H = 1870 \text{ см}^3.$$

4. МЕТОДИ ОПТИМІЗАЦІЇ ТЕХНІЧНИХ ОБ'ЄКТІВ

4.1. Задачі оптимального проектування

Оптимальним називають таке проектування, мета якого полягає в створенні технічного об'єкта, що виконує задані функції і відповідає деяким заздалегідь встановленим вимогам.

Людина завжди прагнула створювати кращі в деякому розумінні машини й механізми. Усвідомлення людиною обмеженості наявних у її розпорядженні матеріальних, енергетичних й інших ресурсів і прагнення до створення найкращих варіантів виробів поставило проблему оптимального проектування в ряд надзвичайно актуальних проблем.

Під оптимізацією розуміють процес вибору найкращого варіанта серед деякої множини варіантів, що задовольняють задані обмеження. При цьому повинно бути чітко визначене поняття «найкращого варіанта». Оскільки критерій якості завжди

вибирає проектувальник, об'єкт може бути оптимальним тільки в розумінні даного критерію.

Розрізняють задачі структурної і параметричної оптимізації. Задача структурної оптимізації пов'язана з поліпшенням характеристик об'єкта за рахунок зміни функціонально - структурної моделі. Задача параметричної оптимізації пов'язана зі зміною керованих параметрів з метою поліпшення характеристик об'єкта при фіксованій функціонально - структурній моделі.

Математична модель оптимального проектування технічного об'єкта являє собою формалізований опис критерію якості, вимог, які пропонуються до об'єкта, й окремих параметрів.

Розроблення математичної моделі починається з визначення мети розв'язання задачі. Наприклад, при проектуванні технічного об'єкта мета може полягати в забезпеченні його мінімальної маси. Можуть бути поставлені інші цілі, такі, як максимальний коефіцієнт корисної дії, мінімальні розміри, мінімальна вартість виготовлення, максимальна надійність, максимальна вантажопідйомність, максимальна продуктивність й ін. Кожній з перерахованих цілей проектування відповідає свій критерій оптимальності (маса, продуктивність, надійність й ін.). Критерії оптимальності виражають у вигляді деякої функції $F(\mathbf{X})$, що залежить від параметрів проектування технічного об'єкта. Функцію $F(\mathbf{X})$ називають цільовою функцією (критерієм ефективності).

Розроблення математичної моделі оптимального проектування починається з вибору параметрів об'єкта, які впливають на критерій оптимальності, і визначення виду функціональної залежності критерію якості від цих параметрів. Далі визначають параметричні й функціональні обмеження для забезпечення об'єктом заданих функцій.

Розглянемо об'єкти, які характеризуються чисельними значеннями внутрішніх параметрів $\mathbf{X} = (x_1, x_2, \dots, x_n)$ і вихідних параметрів $\mathbf{Y} = (y_1, y_2, \dots, y_m)$... Ті внутрішні параметри, які є незалежними один від одного й можуть варіюватися в деяких межах, називаються керованими параметрами.

Нехай об'єкт має n керованих параметрів. Позначимо область визначення цільової функції через \mathbf{XO} . Вектор $\mathbf{X} = (x_1, x_2, \dots, x_n)$ задає координати деякої точки в області \mathbf{XO} .

У багатьох випадках задача параметричної оптимізації формулюється в термінах безперервних параметрів. Задача параметричної оптимізації може не мати обмежень. У цьому випадку екстремум цільової функції називають безумовним, а методи пошуку екстремуму називають методами безумовної оптимізації.

Якщо елементи вектора \mathbf{X} набувають тільки окремих, ізольованих (дискретних) значень, то область \mathbf{XO} є дискретною множиною точок і задача оптимізації називається задачею дискретного програмування. В окремому випадку, коли елементи вектора \mathbf{X} є цілими числами, задача оптимізації називається задачею цілочислового програмування.

Обмеження бувають прямі й функціональні.

Прямими називають обмеження вигляду

$$x_{Hi} \leq x_i \leq x_{Vi}, \quad (4.1)$$

де x_i - i -й параметр технічного об'єкта;

x_{Hi} і x_{Vi} - відповідно мінімальне й максимальне значення i -го параметра.

Сукупність обмежень (4.1) утворить n - вимірний паралелепіпед у просторі проектування.

Функціональні обмеження, які накладають на параметри об'єктів, являють собою умови зв'язку їхніх значень. Ці обмеження можуть мати вигляд

$$g_i(\mathbf{X}) \leq 0; \quad g_j(\mathbf{X}) = 0; \quad g_k(\mathbf{X}) < 0, \quad (4.2)$$

де $g_i(\mathbf{X})$, $g_j(\mathbf{X})$ і $g_k(\mathbf{X})$ – вектор-функції.

Функціональні обмеження зменшують область припустимого підпростору розв'язання й ускладнюють його форму. Як функціональні обмеження в задачах проектування можуть бути умови міцності, твердості, стійкості, вартості, обмеження об'єму й ін.

Таким чином, припустимий підпростір проектування D являє собою множину точок простору проектування R^n , які задовольняють обмеження (4.1) і (4.2).

Якщо в припустимій області D зміни керованих параметрів є тільки одне значення вектора \mathbf{X} , то проблеми вибору не виникає. У тих випадках, коли є декілька працездатних варіантів об'єкта, який проектується, для порівняння декількох варіантів і вибору серед них найкращого (у деякому розумінні) необхідно ввести критерій оптимальності, екстремальне (максимальне або мінімальне) значення якого характеризує властивість одного з найбільш важливих показників проектованого об'єкта.

Цей критерій разом з вектором керованих параметрів \mathbf{X} й описом припустимої області D утворить математичну модель задачі оптимального проектування

$$F(\mathbf{X}^*) = \min_{\mathbf{X} \in D} F(\mathbf{X}). \quad (4.3)$$

Вираз (4.3) є скороченим записом задачі оптимального проектування:

знайти значення керованих параметрів $\mathbf{X} = (x_1, x_2, \dots, x_n)$, які забезпечують мінімальне значення критерію оптимальності

$$F = F(x_1, x_2, \dots, x_n),$$

при виконанні умов працездатності об'єкта, який проектується,

$$g_i(x_1, x_2, \dots, x_n) \geq 0, \quad i = 1, 2, \dots, m;$$

$$x_{Hj} \leq x_j \leq x_{Bj}, \quad j = 1, 2, \dots, n...$$

Якщо цільова функція і (або) обмеження є нелінійними функціями, то задача (5.3) називається задачею нелінійного програмування.

Можна сформулювати більш складну задачу, у якій буде декілька показників якості об'єкта, який проектується. Такі задачі називаються багатокритеріальними задачами.

У багатокритеріальній задачі оптимального проектування потрібно визначити таке значення вектора параметрів $\mathbf{X}^* \in \mathbf{F}$, яке

забезпечувало б мінімум одночасно за всіма критеріями оптимальності.

Найчастіше в цьому випадку формується узагальнений критерій. Процес, що дозволяє звести кілька критеріїв оптимізації в один критерій, називається згорткою критеріїв.

Відомо кілька способів побудови узагальнених критеріїв.

Найбільш часто використовується зважений адитивний критерій

$$F(\mathbf{X}) = \sum_{i=1}^q \lambda_i F_i(\mathbf{X}),$$

де $F_i(\mathbf{X})$ - i -й критерій оптимальності;

λ_i - ваговий коефіцієнт.

При виборі значень вагових коефіцієнтів λ_i враховують ступінь важливості того або іншого критерію на основі експертних оцінок, досвіду або інтуїції.

Вихідні параметри можуть розділятися на дві групи. У першу групу входять параметри, значення яких y_j^+ у процесі оптимізації необхідно збільшувати (продуктивність, імовірність безвідмовної роботи й т.п.). У другу групу входять параметри, значення яких y_j^- необхідно зменшувати (витрату матеріалів, вартість і т.п.). Тоді узагальнений критерій буде мати вигляд

$$F(\mathbf{X}) = \sum_{j=1}^q \lambda_j y_j^-(\mathbf{X}) - \sum_{j=q+1}^m \lambda_j y_j^+(\mathbf{X}),$$

де λ_j - ваговий коефіцієнт.

Оскільки параметри можуть мати різну фізичну розмірність, їх нормують. Найчастіше критерії наводять у безрозмірному вигляді.

Можуть використовуватися інші способи побудови узагальнених критеріїв оптимальності.

Після побудови в першому наближенні математичної моделі оптимального проектування перед дослідником постає задача її аналізу.

Більшість задач параметричної оптимізації формулюється у вигляді задачі нелінійного програмування. Для розв'язання задач нелінійного програмування дуже важливо знати: 1) опукла або неопукла множина припустимих розв'язань задачі; 2) чи є цільова функція опуклою або увігнутою або вона не відноситься ні до одного, ні до іншого класу.

Результати аналізу математичної моделі дозволяють вибрати необхідні математичні методи оптимізації.

Після розв'язання задачі оптимізації дослідник, спираючись на досвід і інтуїцію, повинен оцінити отримані результати. При цьому може виявитися необхідним повторити розв'язання - з іншим критерієм або іншими ваговими коефіцієнтами.

4.2. Методи розв'язання задач оптимального проектування

4.2.1. Основи теорії оптимізації

Залежно від видів функції $F(\mathbf{X})$ оптимальне розв'язання \mathbf{X}^* може бути або точкою локального мінімуму, або точкою глобального мінімуму.

Вектор \mathbf{X}^* називається точкою локального (відносного) мінімуму, якщо для будь-якої точки \mathbf{X} , що належить ε -околу $d(\mathbf{X}^*, \varepsilon)$ цієї точки, функція $F(\mathbf{X})$ не набуває меншого значення:

$$F(\mathbf{X}^*) \leq F(\mathbf{X}) \text{ для всіх } \mathbf{X} \in d(\mathbf{X}^*, \varepsilon).$$

Вектор \mathbf{X}^* є точкою глобального (абсолютного) мінімуму, якщо в жодній іншій точці припустимої області D функція $F(\mathbf{X})$ не набуває меншого значення:

$$F(\mathbf{X}^*) \leq F(\mathbf{X}) \text{ для всіх } \mathbf{X} \in D.$$

Таким чином, глобальний мінімум - це найменший із всіх локальних мінімумів.

На рис. 4.1 показані точки локального (\mathbf{X}_1^* , \mathbf{X}_2^*) і глобального (\mathbf{X}_3^*) мінімумів для довільної кривої.

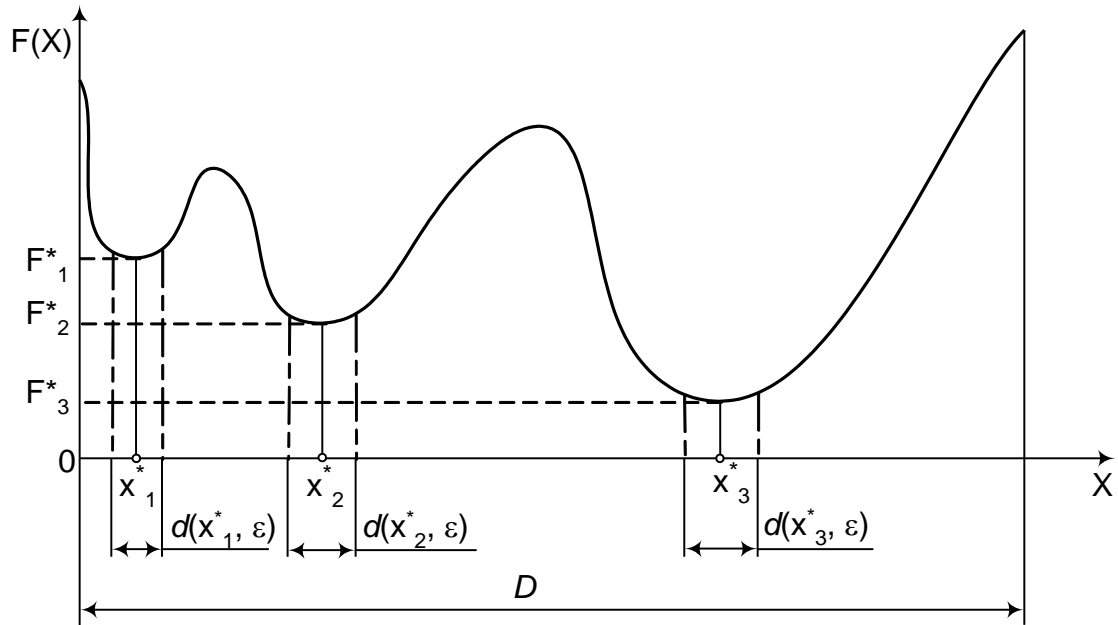


Рис 4.1

Задача параметричної оптимізації, у якій цільова функція $F(\mathbf{X})$ має кілька локальних мінімумів, називається багатоекстремальною задачею оптимізації. Цільова функція, що має один локальний екстремум, називається одноекстремальною (унімодальною).

Нехай $F(\mathbf{X})$ - скалярна функція векторного аргументу розмірності n і не менш ніж двічі диференційована. Необхідна умова того, що \mathbf{X}^* є точкою безумовного локального екстремуму, виражається таким співвідношенням:

$$\frac{\partial F(\mathbf{X})}{\partial x_i} \Big|_{\mathbf{X}=\mathbf{X}^*} = 0 \quad \text{при } i \in [1:n].$$

Необхідні умови екстремуму називають умовами

стаціонарності й часто записують у такому вигляді:

$$\nabla F(\mathbf{X}^*) = 0,$$

де $\nabla F(\mathbf{X}^*) = \left(\frac{\partial F(\mathbf{X}^*)}{\partial x_1}, \dots, \frac{\partial F(\mathbf{X}^*)}{\partial x_n} \right)$ - градієнт цільової функції в точці \mathbf{X}^* .

Символ ∇F читається: «набла» F .

Характер екстремуму в стаціонарній точці залежить від виду функції $F(\mathbf{X})$ в околі точки \mathbf{X}^* . Для цього формується матриця других похідних цільової функції за керованими параметрами (матриця Гессе). Матриця Гессе (гессіан) функції $F(\mathbf{X})$ позначається як $G(\mathbf{X})$ або $\nabla^2 F(\mathbf{X})$ і є симетричною матрицею $n \times n$ елементів вигляду

$$G(\mathbf{X}) = \begin{pmatrix} \frac{\partial^2 F(\mathbf{X})}{\partial x_1^2} & \dots & \frac{\partial^2 F(\mathbf{X})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 F(\mathbf{X})}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 F(\mathbf{X})}{\partial x_n^2} \end{pmatrix}.$$

Якщо матриця Гессе визначена позитивно, то в точці \mathbf{X}^* функція $F(\mathbf{X})$ має безумовний мінімум, якщо визначена негативно – безумовний максимум. В інших випадках точка \mathbf{X}^* є сідловою точкою.

Вважають, що дійсна симетрична матриця $A = [a_{ij}]$ є позитивно визначеною, якщо виконуються умови Сильвестра:

$$\Delta_1 = a_{11} > 0; \quad \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0; \quad \dots;$$

$$\Delta_n = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} > 0,$$

тобто дійсна симетрична матриця A позитивно визначена тоді й тільки тоді, коли всі головні діагональні мінори її визначника $\det A$ строго позитивні.

Наведемо деякі визначення.

Множину називають опуклою, якщо вона разом з будь-якими своїми точками A і B містить і всі точки відрізка AB . На рис. 4.2 наведені приклади опуклих множин точок площини. Прикладами опуклих множин у просторі можуть служити сфера, призма, піраміда й т.п.

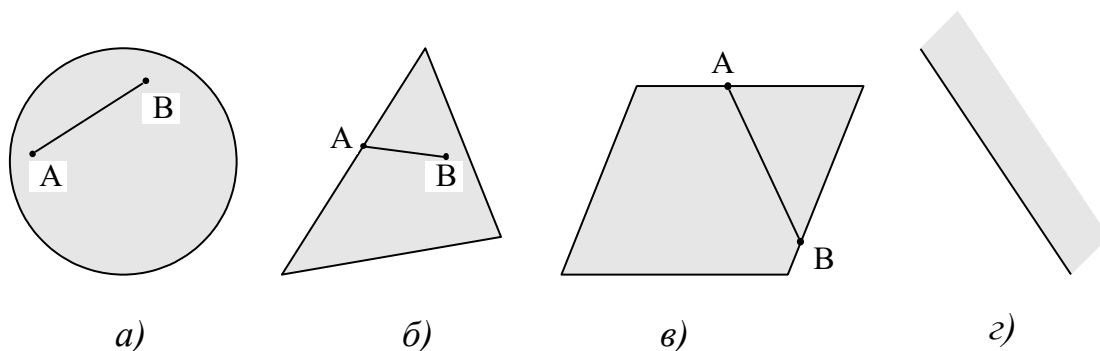


Рис. 4.2

На рис 4.3 наведені приклади неопуклих множин. У неопуклій множині можна вказати хоча б дві точки такі, що не всі точки відрізка AB належать розглянутій множині. Прикладом неопуклої множини в просторі може служити тор.

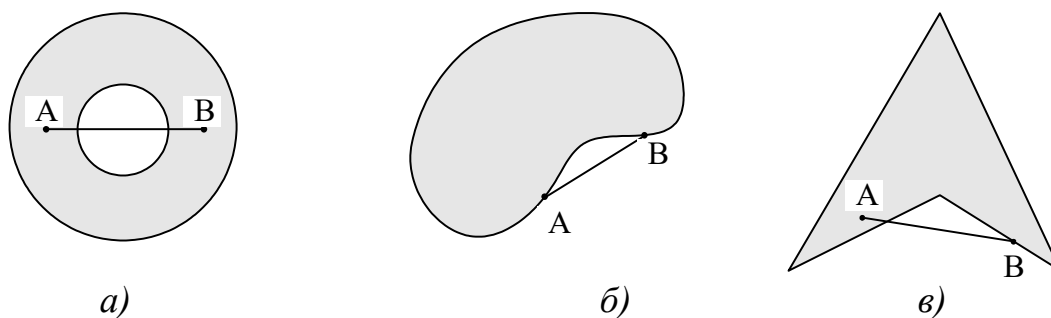


Рис. 4.3

Функція $y = f(x)$ однієї змінної називається опуклою, якщо відрізок, що з'єднує дві будь-які точки її графіка, належить графікові або розташовується вище нього (рис. 4.4,а). Функція називається увігнутою, якщо відрізок, що з'єднує дві будь-які точки графіка, належить йому або розташовується нижче нього (рис. 4.4,б).

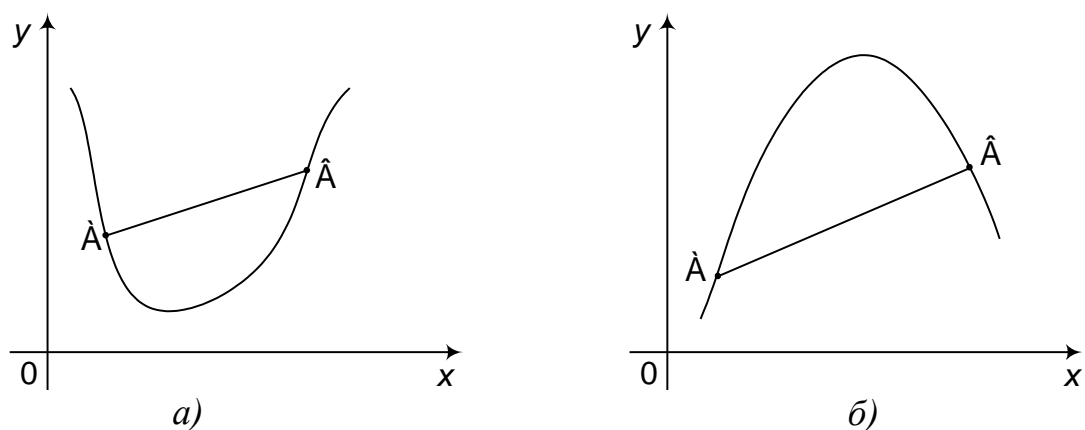


Рис. 4.4

Аналогічно визначаються поняття увігнутої і опуклої функцій декількох змінних. Кажуть, що гіперповерхня $F = F(x_1, x_2, \dots, x_n)$ опукла, якщо відрізок, що з'єднує дві її будь-які точки, лежить на поверхні або вище її. Гіперповерхня $F = F(x_1, x_2, \dots, x_n)$ увігнута, якщо відрізок, що з'єднує дві її будь-які точки, лежить на поверхні або нижче її.

Приклад 4.1

Дослідити екстремальну точку (точки) функції

$$F(X) = 2x_1^2 + 2x_2^2 - 8x_1 - 12x_2 + 60.$$

Необхідна умова мінімуму

$$\nabla F(X) = \begin{pmatrix} 4x_1 - 8 \\ 4x_2 - 12 \end{pmatrix} = 0$$

виконується в точці з координатами $x_1 = 2$, $x_2 = 3$.

Матриця Гессе

$$G(X) = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$$

позитивно визначена. Отже, у точці $(2, 3)$ функція $F(\mathbf{X}^*)$ досягає мінімуму.

Графік функції наведений на рис. 4.5,а.

Нехай потрібно досліджувати функцію

$$F(\mathbf{X}) = -2x_1^2 - 2x_2^2 - 8x_1 - 12x_2 + 60.$$

Необхідна умова мінімуму

$$\nabla F(\mathbf{X}) = \begin{pmatrix} -4x_1 - 8 \\ -4x_2 - 12 \end{pmatrix} = 0$$

виконується в точці з координатами $x_1 = -2$, $x_2 = -3$.

Матриця Гессе

$$G(\mathbf{X}) = \begin{pmatrix} -4 & 0 \\ 0 & -4 \end{pmatrix}$$

негативно визначена. Отже, у точці $(-2, -3)$ функція $F(\mathbf{X}^*)$ досягає максимуму.

Графік функції наведений на рис. 4.5,б.

Розглянемо тепер функцію

$$F(\mathbf{X}) = -2x_1^2 + 2x_2^2 - 8x_1 - 12x_2 + 60.$$

Необхідна умова мінімуму

$$\nabla F(\mathbf{X}) = \begin{pmatrix} -4x_1 - 8 \\ 4x_2 - 12 \end{pmatrix} = 0$$

виконується в точці з координатами $x_1 = -2$, $x_2 = 3$.

Матриця Гессе

$$G(\mathbf{X}) = \begin{pmatrix} -4 & 0 \\ 0 & 4 \end{pmatrix}.$$

Точка $(-2, 3)$ функція $F(\mathbf{X}^*)$ є сідловою точкою.

Графік функції наведений на рис. 4.5, в.

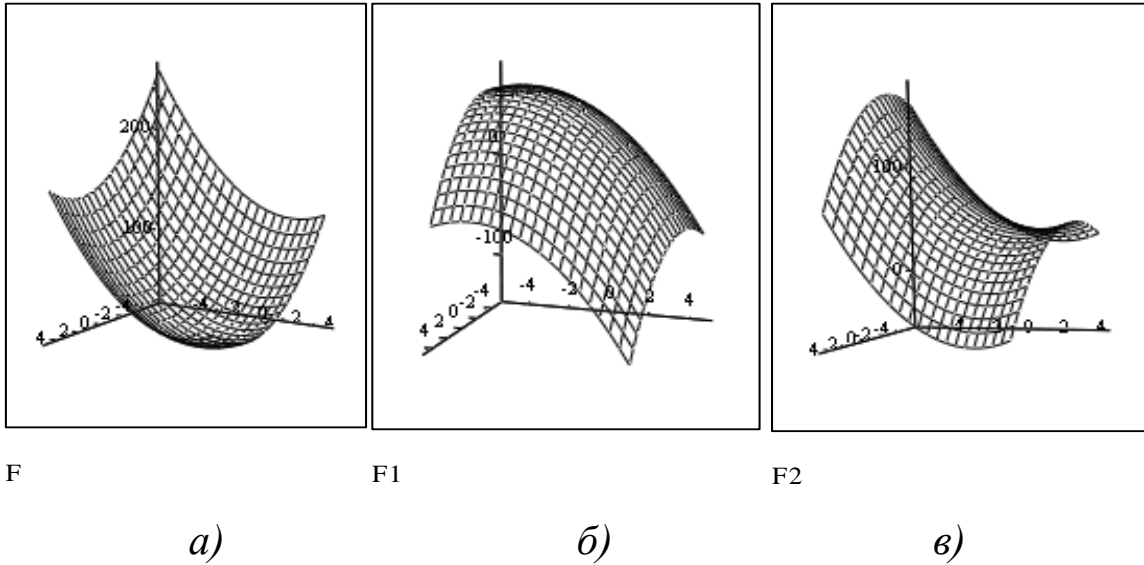


Рис. 4.5

Функцію двох змінних зручно подати у вигляді ліній рівних рівнів. Ці лінії отримані при перетинанні поверхні площиною, паралельною площині координат (x, y) . Геометричні зображення зручно використовувати для виразу характерних рис функції (наявність сідлових точок, багатоекстремальність і т.п.). Лінії рівня функції $F(x, y) = x^2 + (2y)^2$ показані на рис. 4.6,*а*. Графік функції показаний на рис. 4.6,*б*.

$$F(x, y) := x^2 + (2y)^2$$

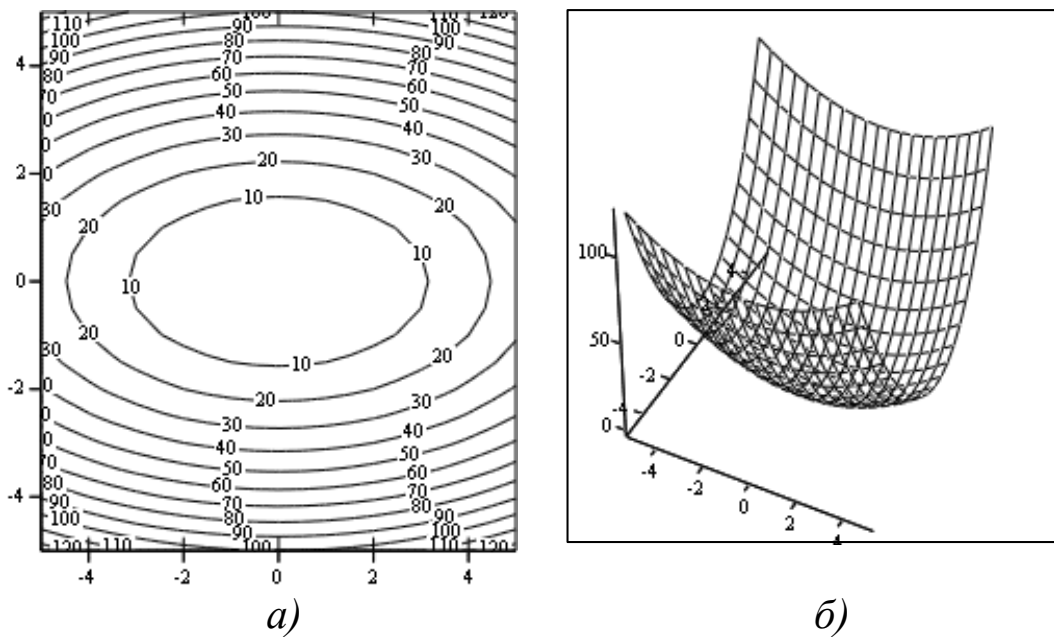


Рис. 4.6

4.2.2. Методи одновимірного пошуку

Методи одновимірного пошуку можливо розділити на дві групи: методи послідовного пошуку (методи дихотомії, Фібоначчі, золотого перетину) і методи, які використовують апроксимацію функції (методи квадратичної й кубічної інтерполяції, метод Пиявського та ін.).

Розглянемо методи послідовного пошуку. Задача одновимірного пошуку вирішується в такий спосіб. Значення змінної x перебувають в інтервалі $a \leq x \leq b$. Інтервал значень x , у якому перебуває оптимальне значення x^* , називається інтервалом невизначеності. На початку процесу оптимізації цей інтервал має довжину $b - a$. До функції не пред'являються вимоги диференційованості або безперервності. Обчисливши значення цільової функції $F(x_1)$ і $F(x_2)$ при значеннях x_1 й x_2 у зазначеному інтервалі, звужимо інтервал невизначеності:

якщо $F(x_1) < F(x_2)$, то $x^* \in [a, x_2]$;

якщо $F(x_1) = F(x_2)$, то $x^* \in [x_1, x_2]$;

якщо $F(x_1) > F(x_2)$, то $x^* \in [x_1, b]$.

Існує декілька способів звуження інтервалу невизначеності, які викладаються нижче.

Метод дихотомії (половинного ділення)

Обчислення цільової функції у двох точках інтервалу невизначеності дозволяє його звужити. Можливо в такий спосіб вибрати ці точки, що інтервал невизначеності буде мінімальним.

Пошук проводиться в такій послідовності. На k -му кроці проводиться два досліди. Значення $x_{1,k}$ й $x_{2,k}$ повинні вибиратися на відстані $\delta/2$ праворуч і ліворуч від середини інтервалу:

$$x_{1,k} = \frac{a_k + b_k}{2} - \frac{\delta}{2}; \quad x_{2,k} = \frac{a_k + b_k}{2} + \frac{\delta}{2},$$

де $\delta > 0$ – коефіцієнт, що визначає відстань між двома значеннями аргументу;
 a_k, b_k - межі інтервалу на k -му кроці.

Обчисливши $F(x_{1,k})$ і $F(x_{2,k})$ і порівнявши отримані значення, знайдемо нові межі інтервалу невизначеності (рис 5.7):

якщо $F(x_{1,k}) < F(x_{2,k})$, то $a_{k+1} = a_k, b_{k+1} = x_{2,k}$;

якщо $F(x_{1,k}) = F(x_{2,k})$, то $a_{k+1} = x_{1,k}, b_{k+1} = x_{2,k}$; (4.4)

якщо $F(x_{1,k}) > F(x_{2,k})$, то $a_{k+1} = x_{1,k}, b_{k+1} = b_k$.

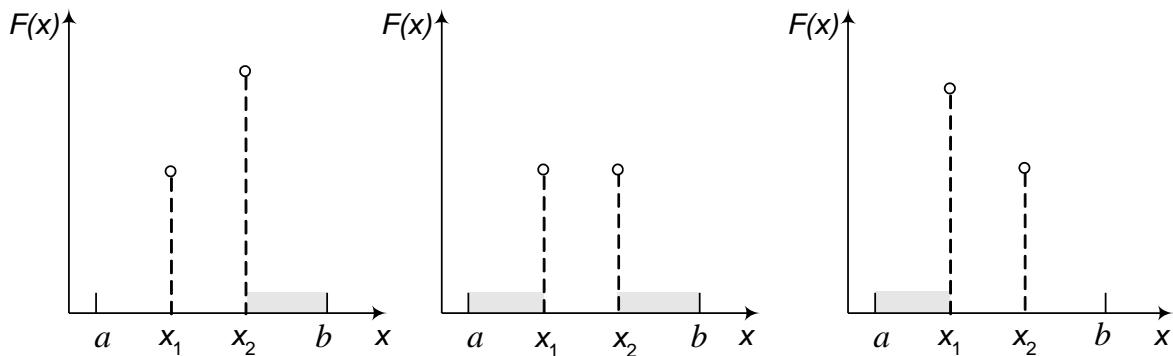


Рис. 4.7

Потім знову обчислимо координати x_1 й x_2 і продовжимо пошук.

Після n експериментів (n - парне число) екстремум буде в інтервалі довжиною $l(n) = (b - a)[2^{-n/2} + (1 - 2^{-n/2}) \delta]$. Цей інтервал можна приймати за оцінку умови припинення пошуку.

Приклад 4.2

Продемонструємо застосування методу дихотомії на прикладі мінімізації функції

$$F(x) = -30x^3 + 275x^2 - 75x + 18.$$

Як початкові точки приймемо $a = 0.1, b = 0.5$. Необхідна точність $\delta = 0.0125$.

На першому кроці відрізок ділиться навпіл точкою $x = \frac{a+b}{2}$.
 Для того, щоб визначити, з якої сторони від цієї точки перебуває мінімум, обчислимо значення функції в точках $x_1 = \frac{a+b}{2} - \frac{\delta}{2} = 0.294$ й $x_2 = \frac{a+b}{2} + \frac{\delta}{2} = 0.306$.

Значення функції в цих точках рівні $F(x_1) = 18.938$, $F(x_2) = 19.969$. Тому що $F(x_1) < F(x_2)$, то приймаємо $b = x_2$ й обчислюємо нові значення $x_1, x_2, F(x_1), F(x_2)$.

Результати обчислень наведені в табл. 4.1.

Таблиця 4.1

k	a	b	x_1	x_2	$F(x_1)$	$F(x_2)$
0	0.1	0.5	0.294	0.306	18.938	19.962
1	0.1	0.306	0.197	0.209	13.664	14.077
2	0.1	0.209	0.148	0.161	12.828	12.927
3	0.1	0.161	0.124	0.137	12.869	12.81
4	0.124	0.161	0.136	0.149	12.81	12.83
5	0.124	0.149	0.13	0.143	12.83	12.81
6	0.13	0.149	0.133	0.146	12.818	12.818
7	0.133	0.149	0.135	0.147	12.814	12.823

Графік функції на інтервалі $[0, 0.5]$ показаний рис. 4.8.

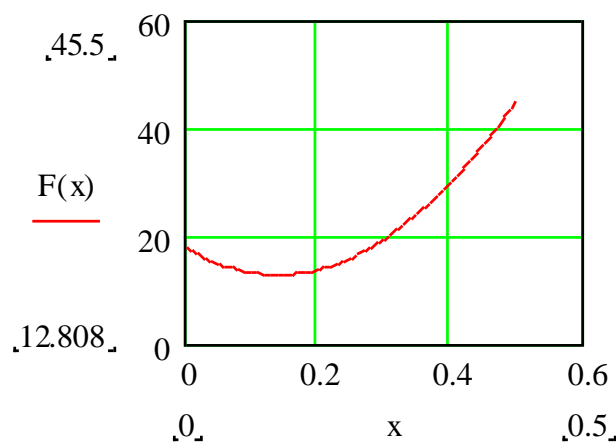


Рис. 4.8

Метод золотого перетину

У методі золотого перетину інтервал невизначеності ділиться на дві нерівні частини так, що відношення більшого відрізка до довжини всього інтервалу дорівнює відношенню довжини меншого відрізка до довжини більшого відрізка (рис. 4.8). Це правило можна записати в такому вигляді:

$$\frac{z_1}{Z} = \frac{z_2}{z_1}.$$

Крім того, $z_1 + z_2 = Z$. З першого рівняння випливає $z_1^2 = Zz_2$. Підставляючи сюди значення Z із другого рівняння й ділячи обидві частини на z_1^2 , одержимо

$$\left(\frac{z_2}{z_1}\right)^2 + \frac{z_2}{z_1} - 1 = 0.$$

Розв'язуючи це квадратне рівняння, знаходимо для позитивного кореня значення

$$\frac{z_2}{z_1} = \frac{-1 + \sqrt{5}}{2} = 0.618.$$

На рис. 4.9 показано ділення інтервалу невизначеності в цьому відношенні.

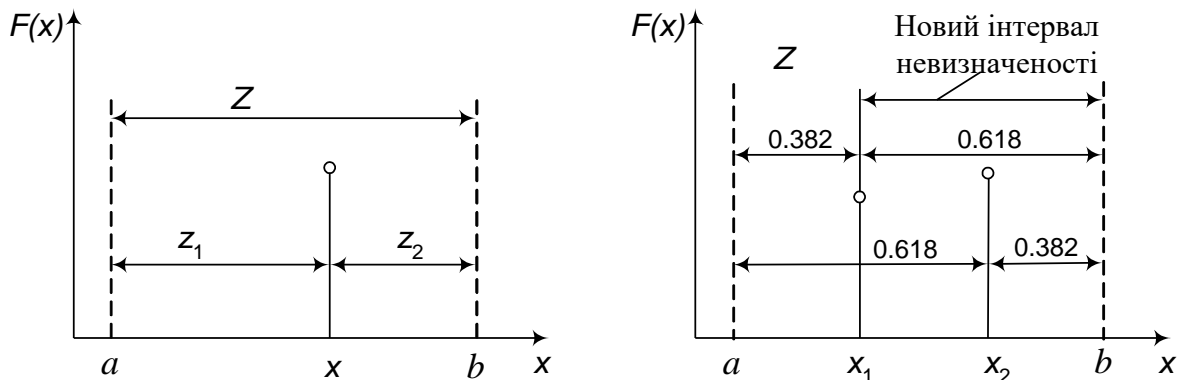


Рис. 4.9

Таким чином, у методі золотого перетину аргументи $x_{1,k}$ й $x_{2,k}$ вибираються в такий спосіб:

$$x_{1,k} = a_k + 0.382(b_k - a_k), \quad x_{2,k} = b_k - 0.382(b_k - a_k).$$

Потім у цих точках обчислюються значення цільової функції, аналізуються умови (4.4) і пошук триває.

Можливо показати, що $\frac{b_k - x_{2,k}}{x_{2,k} - a_k} = \frac{x_{2,k} - a_k}{b_k - a_k}$ й $\frac{x_{2,k} - x_{1,k}}{x_{1,k} - a_k} = \frac{x_{1,k} - a_k}{x_{2,k} - a_k}$,

тобто точки $x_{1,k}$ й $x_{2,k}$ є золотим перетином відрізка $[a_k, x_{2,k}]$ й $[a_k, b_k]$ відповідно.

Приклад 4.3

Знайдемо мінімум функції $F(x) = -30x^3 + 275x^2 - 75x + 18$ методом золотого перетину. Як початкові точки прийнемо $a = 0.1$, $b = 0.5$. Необхідна точність $\delta = 0.0125$.

Результати обчислень наведені в табл. 4.2.

Значення функції в початкових точках x_1 x_2 і дорівнюють:

$$F(x_1) = -30x_1^3 + 275x_1^2 - 75x_1 + 18; \quad F(x_1) = 18.923;$$

$$F(x_2) = -30x_2^3 + 275x_2^2 - 75x_2 + 18; \quad F(x_2) = 19.962.$$

Таблиця 4.2

k	a	b	x_1	x_2	$F(x_1)$	$F(x_2)$
0	0.1	0.5	0.253	0.347	16.39	23.8
1	01.	0.347	0.194	0.253	13.593	16.1
2	01.	0.253		0.194	12.9	13.5
3	01.	0.194	0.136	0.158	12.811	12.9
4	01	0.158	0.122	0.136	12.818	12.8
5	0.122	0.158	0.136	0.145	12.811	12.8
6	0.122	0.145	0.131	0.136	12.821	12.8
7	0.131	0.145	0.136	0.139	12.811	12.8
8	0.136	0.145	0.139	0.1414	12.808	12.8

4.2.3. Методи багатовимірного пошуку

Існує багато методів багатовимірного пошуку екстремуму функцій. Розглянемо деякі з них.

Методи дослідження функцій класичного аналізу засновані на відомих методах диференційного обчислення.

Необхідна умова існування екстремуму цільової функції $Q(\mathbf{X})$ полягає в тому, що похідна в точці екстремуму дорівнює нулю. Тоді екстремум функції $Q(\mathbf{X})$ можна знайти із системи рівнянь

$$\frac{\partial Q}{\partial x_i} = 0, \quad i = 1, 2, \dots, n \dots \quad (4.5)$$

Для того, щоб визначити, чи є розв'язок точкою мінімуму або максимуму, досліджують достатні умови існування екстремуму, згідно з яким: якщо похідна в точці екстремуму змінює знак з мінуса на плюс, то $Q(\mathbf{X}^*)$ є мінімум цільової функції; якщо значення похідної в точці екстремуму міняє знак із плюса на мінус, то $Q(\mathbf{X}^*)$ є максимум цільової функції; якщо значення похідної в точці (\mathbf{X}^*) не міняється, то в цій точці немає екстремуму.

Якщо рівняння (4.5) нелінійні, то розв'язати їх систему аналітичним способом вдається вкрай рідко. В цьому випадку використовують ЕОМ і відповідні чисельні методи.

Методи дослідження функцій класичного аналізу можна використати для розв'язання нескладних задач оптимізації без обмежень.

Метод найшвидшого спуску

Метод найшвидшого спуску вперше був запропонований Коші в 1847 році.

Напрямок градієнта в кожній точці збігається з напрямком найбільш швидкого зростання цільової функції, тобто локально найкращим є градієнтний напрямок при максимізації або антиградієнтний напрямок при мінімізації.

У процесі пошуку мінімуму функції рух від однієї точки до іншої здійснюється в напрямку найшвидшого зменшення функції:

$$x_{k+1} = x_k - \lambda^k \left(\frac{\partial F}{\partial x} \right)_k,$$

тобто за антиградієнтом, а величина кроку в обраному напрямку визначається в результаті розв'язання одновимірної задачі мінімізації F :

$$F(x_k) = \lambda^k \left(\frac{\partial F}{\partial x} \right)_k.$$

Приклад 4.4

Знайти мінімум функції $F(x_1, x_2) = x_1^2 + 25x_2^2$. Початкове наближення дорівнює $x_1=2$, $x_2=2$, точність розв'язання $\varepsilon = 0.2$. Можна записати початкові значення у векторній формі

$$\mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Значення функції в початковій точці $F(x_1, x_2) = 104$.

Часткові похідні

$$\frac{\partial}{\partial x_1} F(x_1, x_2) = 4; \quad \frac{\partial}{\partial x_2} F(x_1, x_2) = 100.$$

Отже,

$$F(\lambda) = \left[2 - \left(\frac{\partial}{\partial x_1} F(x_1, x_2) \right) \lambda^{(1)} \right]^2 + 25 \left[2 - \left(\frac{\partial}{\partial x_2} F(x_1, x_2) \right) \lambda^{(1)} \right]^2.$$

Знайдемо мінімум цієї функції за допомогою класичного методу Ейлера

$$\frac{\partial F(\lambda)}{\partial \lambda} = 250016 \lambda^2 - 10016 \lambda + 104 = 0.$$

Функція $F(\lambda^{(1)})$ має дві стаціонарні точки. Можна показати, що функція $F(\lambda)$ досягає мінімуму при $\lambda^{(1)}=0.481$.

Отже, нове значення вектора

$$\mathbf{X}^{(1)} = \mathbf{X} - \lambda^{(1)} \begin{pmatrix} \frac{\partial}{\partial x_1} F(x_1, x_2) \\ \frac{\partial}{\partial x_2} F(x_1, x_2) \end{pmatrix}; \quad \mathbf{X}^{(1)} = \begin{pmatrix} 1.92 \\ -3.072 \times 10^{-3} \end{pmatrix}.$$

Значення функції $F(x_1, x_2)=3.686$.

На другому кроці

$$F(\lambda^{(2)}) = \left[1.92 - \left(\frac{\partial}{\partial x_1} F(x_1, x_2) \right) \lambda^{(2)} \right]^2 + 25 \left[-3.072 \times 10^{-3} - \left(\frac{\partial}{\partial x_2} F(x_1, x_2) \right) \lambda^{(2)} \right]^2.$$

Часткові похідні

$$\frac{\partial}{\partial x_1} F(x_1, x_2) = \frac{300000}{7813} \quad \frac{\partial}{\partial x_2} F(x_1, x_2) = \frac{-1200}{7813}.$$

Взявши похідну за параметром λ і дорівнюючи її нулю, отримаємо

$$15.333 \lambda^2 - 14.768 \lambda + 3.786 = 0.$$

Розв'язавши це рівняння, одержимо нове найменше значення: значення $\lambda^{(2)}=0.481$.

Отже, нова точка

$$\mathbf{X}^{(2)} = \mathbf{X}^{(1)} - \lambda^{(2)} \begin{pmatrix} \frac{\partial}{\partial x_1} F(x_1, x_2) \\ \frac{\partial}{\partial x_2} F(x_1, x_2) \end{pmatrix}; \quad \mathbf{X}^{(2)} = \begin{pmatrix} 0.0717 \\ 0.071 \end{pmatrix}.$$

Значення функції в початковій точці $F(x_1, x_2)=3.689$.

Аналогічно виконаємо наступний крок.

Часткові похідні

$$\frac{\partial}{\partial x_1} F(x_1, x_2) = 0.415; \quad \frac{\partial}{\partial x_2} F(x_1, x_2) = 3.584.$$

Отже,

$$F(\lambda^{(3)}) = \left[0.071 - \left(\frac{\partial}{\partial x_1} F(x_1, x_2) \right) \lambda^{(3)} \right]^2 + 25 \left[0.071 - \left(\frac{\partial}{\partial x_2} F(x_1, x_2) \right) \lambda^{(3)} \right]^2.$$

Значення параметра дорівнює $\lambda^{(3)} = 2.031 \times 10^{-2}$.

Нові значення змінних дорівнюють

$$\mathbf{X}^{(3)} = \mathbf{X}^{(2)} - \lambda^{(3)} \begin{pmatrix} \frac{\partial}{\partial x_1} F(x_1, x_2) \\ \frac{\partial}{\partial x_2} F(x_1, x_2) \end{pmatrix}; \quad \mathbf{X}^{(3)} = \begin{pmatrix} 0.0686 \\ -1.089 \times 10^{-4} \end{pmatrix}.$$

При цьому $F(x_1, x_2) = 4.615 \times 10^{-3}$.

Ітерації тривають доти, поки не будуть виконуватися нерівності

$$\left| \frac{\partial}{\partial x_1} F(x_1, x_2) \right| = 0.136 < \varepsilon; \quad \left| \frac{\partial}{\partial x_2} F(x_1, x_2) \right| = 0.00 < \varepsilon.$$

Точне значення мінімуму функції дорівнює нулю.

Метод покоординатного спуску

У методі покоординатного спуску на кожному кроці k міняється тільки одна змінна.

Значення змінної на $k+1$ кроці визначається за допомогою такого рекурентного співвідношення:

$$x_r^{k+1} = x_r^k - \frac{\left(\frac{\partial F}{\partial x_r} \right)^k}{\left(\frac{\partial^2 F}{\partial x_r^2} \right)^k}.$$

Приклад 4.5

Знайти мінімум функції $F(x_1, x_2) = x_1^2 + 25x_2^2$.

Початкове наближення дорівнює $x_1=2, x_2=2$.

У початковій точці похідні рівні:

$$\frac{\partial}{\partial x_1} F(x_1, x_2) = 4; \quad \frac{\partial^2}{\partial x_1^2} F(x_1, x_2) = 2.$$

Нове значення координати в такій точці:

$$x_1^{(1)} = x_1 - \frac{\left(\frac{\partial}{\partial x_1} F(x_1, x_2) \right)}{\left(\frac{\partial^2}{\partial x_1^2} F(x_1, x_2) \right)}; \quad x_1^{(1)} = 1.572 \times 10^{-13}.$$

Часткові похідні за координатою x_2 :

$$\frac{\partial}{\partial x_2} F(x_1, x_2) = 100; \quad \frac{\partial^2}{\partial x_2^2} F(x_1, x_2) = 50.$$

Значення координати x_2 :

$$x_2^{(1)} = x_2 - \frac{\left(\frac{\partial}{\partial x_2} F(x_1, x_2) \right)}{\left(\frac{\partial^2}{\partial x_2^2} F(x_1, x_2) \right)}; \quad x_2^{(1)} = 1.322 \times 10^{-14}.$$

При цьому $F(x_1, x_2) = 3.631 \times 10^{-13}$.

4.3. Лінійне програмування

Слово «програмування» тут розуміється як планування. Ідеї лінійного програмування з'явилися в роки другої світової війни у зв'язку з пошуком оптимальної стратегії ведення бойових операцій. Вони знайшли широке застосування в промисловості і управлінні.

Лінійне програмування є математичним апаратом розв'язання задач оптимізації, у яких цільова функція й обмеження лінійні.

Задача лінійного програмування формулюється в такий спосіб: знайти мінімальне (максимальне) значення цільової функції

$$F(x) = \sum_{j=1}^n c_j x_j,$$

при обмеженнях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m_1;$$

$$\sum_{j=1}^n a_{kj} x_j = b_k, \quad k = 1, 2, \dots, m_2;$$

$$\sum_{j=1}^n a_{lj} x_j \geq b_l, \quad l = 1, 2, \dots, m_3;$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Тут $c_j, a_{ij}, a_{kj}, a_{lj}, b_i, b_k, b_l$ - задані дійсні числа; x_j - параметри.

Приклад 4.6

Підприємство робить дві моделі A і B складальних виробів. Їхнє виробництво обмежене наявністю сировини й моделі A часом машинної обробки. Для кожного виробу моделі A потрібно 2 кг матеріалу, а для виробу моделі B – 3 кг. Підприємство може отримати від своїх постачальників до 1200 кг матеріалу за тиждень. Для кожного виробу моделі A потрібно 24 хв (0.4 год) машинного часу, а для виробу моделі B – 12 хв (0.2 год). У тиждень можна використати 160 год машинного часу.

Прибуток від продажу виробу A становить 6 грн, від продажу виробу B – 4 грн.

Скільки виробів кожної моделі варто випускати за тиждень, щоб отримувати найбільший прибуток?

Створимо математичну модель.

Позначимо через x_1 кількість випущених за тиждень виробів моделі A , а через x_2 – кількість випущених виробів моделі B .

Щотижневий прибуток

$$F = 6x_1 + 4x_2.$$

Обмеження на наявність матеріалу має такий вигляд:

$$2x_1 + 3x_2 \leq 1200.$$

Обмеження на машинний час може бути записане в такий спосіб:

$$4x_1 + 2x_2 \leq 1600.$$

Змінні x_1 й x_2 не можуть бути негативними, тобто

$$x_1 \geq 0; \quad x_2 \geq 0.$$

Отже, завдання полягає в тому, щоб знайти значення x_1 і x_2 , які доставляють максимум цільової функції

$$F = 6x_1 + 4x_2$$

при умовах

$$2x_1 + 3x_2 \leq 1200;$$

$$4x_1 + 2x_2 \leq 1600;$$

$$x_1 \geq 0;$$

$$x_2 \geq 0.$$

Для розв'язання задачі можна використати графічний метод.

Побудуємо на площині в прямокутній декартовій системі координат x_1Ox_2 область припустимих розв'язань задачі (рис. 4.9)

Кожна нерівність системи обмежень утворить на площині дві області: область можливих й область припустимих розв'язань. Система обмежень утворить на площині замкнуту опуклу область припустимих розв'язань. На рис. 4.10 ця область заштрихована.

Значення цільової функції $F(\mathbf{X}) = 6x_1 + 4x_2$ можливо визначити в будь-якій точці $\mathbf{X} = (x_1, x_2)$ області припустимих розв'язань. Лінії a , b і c на рис. 4.9 зображують прямі $6x_1 + 4x_2 = 0$, $6x_1 + 4x_2 = 3600$ й $6x_1 + 4x_2 = 4800$ відповідно. Ці прямі паралельні і є лініями рівня F зі значеннями відповідно 0, 3600 й 4800. Значення функції F зростає в міру того, як лінія рівня віддаляється від початку координат у позитивному квадранті.

Найбільшого значення функція F досягає в точці m , яка належить області припустимих розв'язань. Ця точка має координати $x_1 = 300$, $x_2 = 200$.

Таким чином, для отримання найбільшого прибутку підприємство повинне випускати в тиждень 300 виробів моделі A і 200 виробів моделі B . При цьому прибуток підприємства буде становити 2600 грн.

4.4. Нелінійне програмування

Задача нелінійного програмування формулюється таким чином:

знайти мінімальне (максимальне) значення функції

$$\min(\max) F(x_1, x_2, \dots, x_n) \quad (4.6)$$

при умовах

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) &\geq 0; \\ g_2(x_1, x_2, \dots, x_n) &\geq 0; \\ g_m(x_1, x_2, \dots, x_n) &\geq 0, \end{aligned} \quad (4.7)$$

де $F(x_1, x_2, \dots, x_n)$ і (або) $g_i(x_1, x_2, \dots, x_n)$ - нелінійні функції змінних x_1, x_2, \dots, x_n .

Нелінійне програмування відрізняється від лінійного програмування тим, що тут відсутні універсальні методи

розв'язання типу симплекс – методу в лінійному програмуванні. Це пов'язане з тим, що припустима множина розв'язань $R(x_1, x_2, \dots, x_n)$, яка визначається умовами (4.7), не є опуклою, а крім того навіть у випадку опуклості $R(x_1, x_2, \dots, x_n)$ множина його крайніх точок не кінцева. Цим викликане різноманіття методів нелінійного програмування, які розробляються лише під спеціальні класи задач.

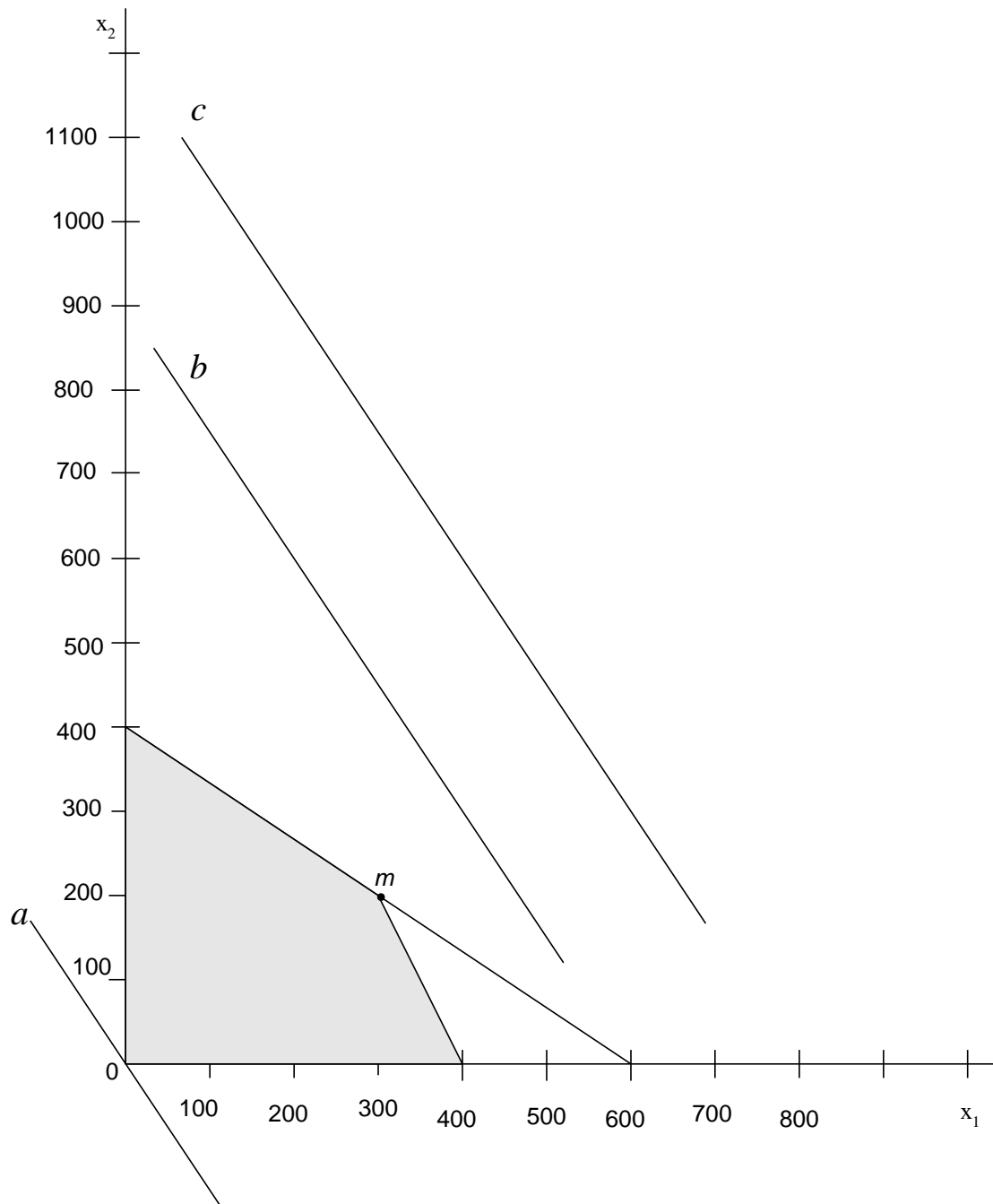


Рис. 4.10

Метод множників Лагранжа

Метод множників Лагранжа дозволяє розв'язувати задачі з обмеженнями у вигляді рівностей. Суть методу полягає в уведенні p невизначених множників λ_j ($j=1, 2, \dots, p$) і побудові функції Лагранжа

$$L = Q(\mathbf{X}) - \sum_{j=1}^p \lambda_j g_j(\mathbf{X}).$$

Для визначення значень змінних x_i ($i=1, 2, \dots, n$) розв'язують систему $n+1$ рівнянь

$$\frac{\partial L}{\partial x_i} = 0, \quad i=1, 2, \dots, n); \quad g_j(\mathbf{X})=0, \quad j=1, 2, \dots, p) \quad (4.8)$$

щодо невідомих p й λ .

Якщо (4.6) являє собою систему нелінійних рівнянь, то труднощі їх розв'язання аналогічні розв'язанню системи (4.5). Слід зазначити, що система (4.6) - лише необхідна умова існування екстремуму. Тому знайдені з її розв'язання значення змінних необхідно перевірити на екстремум аналізом похідних більш високого порядку.

Приклад 4.6

Визначити розміри a , b й h контейнера об'ємом $V=10$, що мав би мінімальну масу. Будемо вважати, що маса контейнера лінійно залежить від його площі поверхні.

Критерій оптимальності цієї задачі

$$Q = 2ab + 2ah + 2bh.$$

Обмеження на параметри a , b й h відповідно до умови задачі визначаються виразом

$$abh = V$$

або

$$abh - V = 0.$$

Згідно з методом множників Лагранжа складемо функцію

$$L(a, b, h, \lambda) = 2ab + 2ah + 2bh - \lambda(abh - V),$$

де λ - множник Лагранжа.

Часткові похідні:

$$\frac{\partial}{\partial a} L(a, b, h, \lambda) = 2b + 2h - \lambda hb = 0;$$

$$\frac{\partial}{\partial b} L(a, b, h, \lambda) = 2a + 2h - \lambda ha = 0;$$

$$\frac{\partial}{\partial h} L(a, b, h, \lambda) = 2a + 2b - \lambda ab = 0;$$

$$\frac{\partial}{\partial \lambda} L(a, b, h, \lambda) = -10 + abh = 0.$$

Розв'язавши цю систему рівнянь, отримаємо такі розміри контейнера об'ємом V :

$$a = 2.154; b = 2.154; h = 2.154; \lambda = -1.857.$$

При цьому площа поверхні контейнера $Q = 27.85$.

Аналізом похідних другого порядку необхідно показати, що знайдена точка є точкою мінімуму цільової функції.

Задача нелінійного програмування з лінійною цільовою функцією і нелінійними обмеженнями

Для більш наочного подання обмежимося прикладами, в яких використовуються функції двох змінних.

Розглянемо таку задачу: знайти максимум і мінімум функції $F=2x_1 + x_2$ при обмеженнях

$$x_1^2 + x_2^2 \leq 5;$$

$$x_1 \geq 0;$$

$$x_2 \geq 0.$$

Множина припустимих розв'язків показана на рис. 4.11.

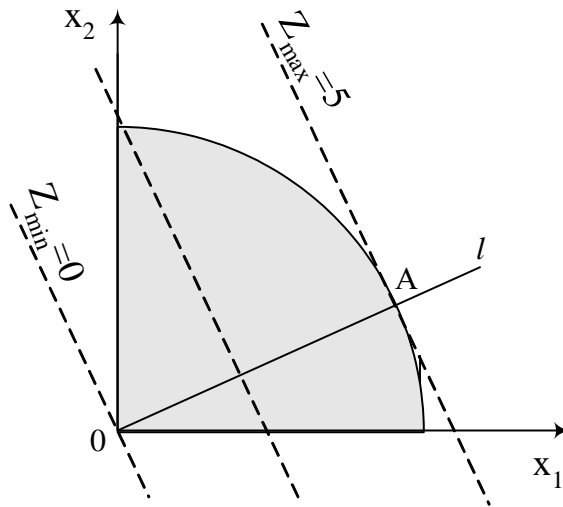


Рис. 4.11

Штрихові лінії з кутовим коефіцієнтом $k=-2$ на рис. 4.11 є лініями рівня функції $F=2x_1 + x_2$. Очевидно, що функція має мінімум у точці $0(0,0)$ і глобальний максимум у точці A торкання прямої рівня й окружності $x_1^2 + x_2^2 = 5$.

Для того, щоб знайти координати точки A , необхідно скласти рівняння прямої l і розв'язати систему рівнянь із прямої і окружності. Пряма перпендикулярна лінії рівня й, отже, її кутовий коефіцієнт k_1 дорівнює $1/2$. Пряма l проходить через точку O і має кутовий коефіцієнт $k_1 = 1/2$, а її рівняння $x_2=1/2x_1$. Розв'язуючи систему рівнянь

$$x_1^2 + x_2^2 = 5;$$

$$x_2=1/2x_1,$$

отримаємо

$$x_1 = 2; x_2 = 1.$$

Таким чином, максимум досягається в точці $A(2,1)$, значення функції $F=2$.

Задача нелінійного програмування з нелінійною цільовою функцією й нелінійними обмеженнями

Розглянемо задачу.

Знайти максимум функції

$$F = (x_1 - 4)^2 + (x_2 - 3)^2$$

при обмеженнях

$$x_1^2 + x_2^2 \leq 9;$$

$$x_1 \geq 0;$$

$$x_2 \geq 0.$$

Припустима множина розв'язань показана на рис. 4.11.

Лініями рівня функції $F = (x_1 - 4)^2 + (x_2 - 3)^2$ є окружності із центром у точці $A(4,3)$. На рис. 4.12 видно, що глобальний максимум функції F досягається в точці $O(0,0)$. При цьому максимальне значення функції $F = 25$.

4.5. Застосування комп'ютера для пошуку екстремуму функції

Для розв'язання задачі пошуку екстремуму можуть використовуватися програмні продукти Mathcad й Excel.

Mathcad є потужним універсальним засобом для розв'язання різноманітних математичних задач. Він дозволяє автоматизувати процес обчислень. Убудований графічний редактор дозволяє створювати документ, що включає текст і математичні формули.

Для розв'язання задачі пошуку екстремуму функції в Mathcad є убудовані функції Maximize й Minimize. Ці функції використовують градієнтні чисельні методи.

Функції Maximize й Minimize дозволяють знаходити локальний екстремум. Для пошуку глобального максимуму або мінімуму необхідно обчислити значення локальних екстремумів і потім знайти найбільше (найменше) значення.

Функція Maximize має такий формат:

$$\text{Maximize}(f, x_1, x_2, \dots, x_n),$$

де x_1, x_2, \dots, x_n - вектор значень аргументів, при яких цільова функція f досягає максимуму.

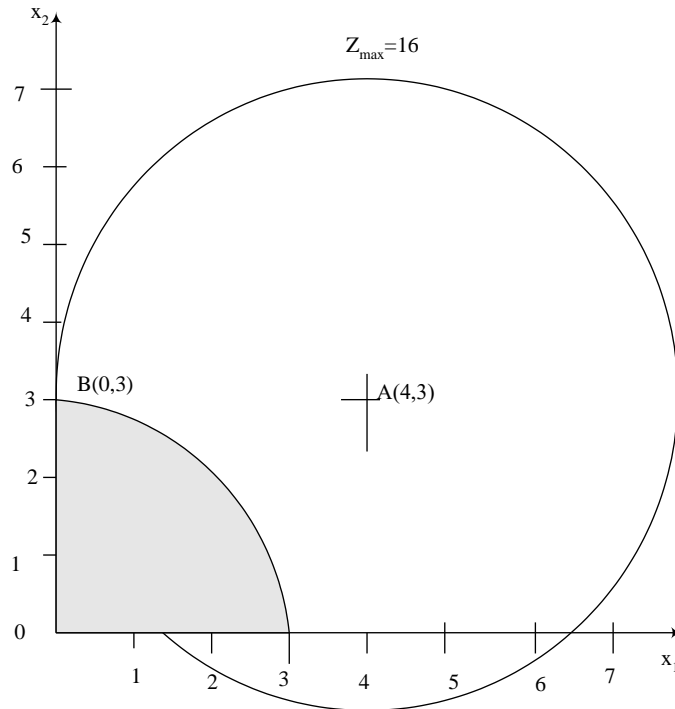


Рис. 4.12

Функція Minimize записується в такому вигляді:

$$\text{Minimize}(f, x_1, x_2, \dots, x_n),$$

де x_1, x_2, \dots, x_n - вектор значень аргументів, при яких цільова функція f досягає мінімуму.

Всім аргументам функції попередньо присвоюються початкові значення.

Приклади обчислення безумовного екстремуму функції однієї змінної наведені на рис. 4.13 і 4.14.

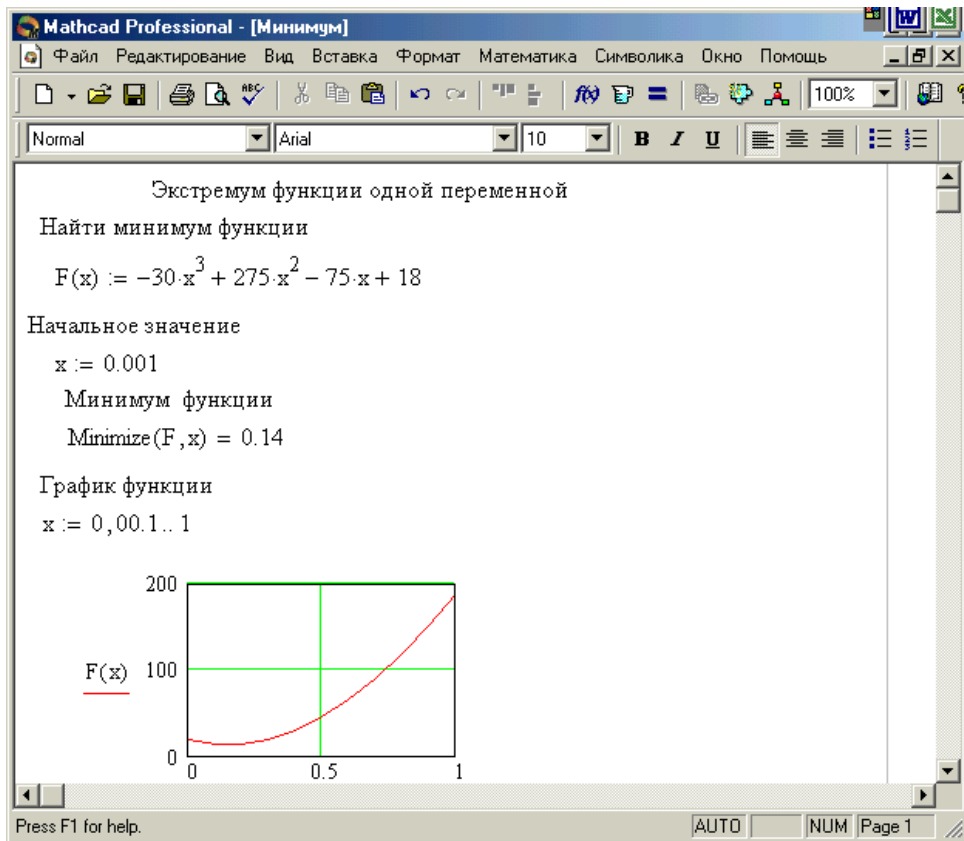


Рис. 4.13

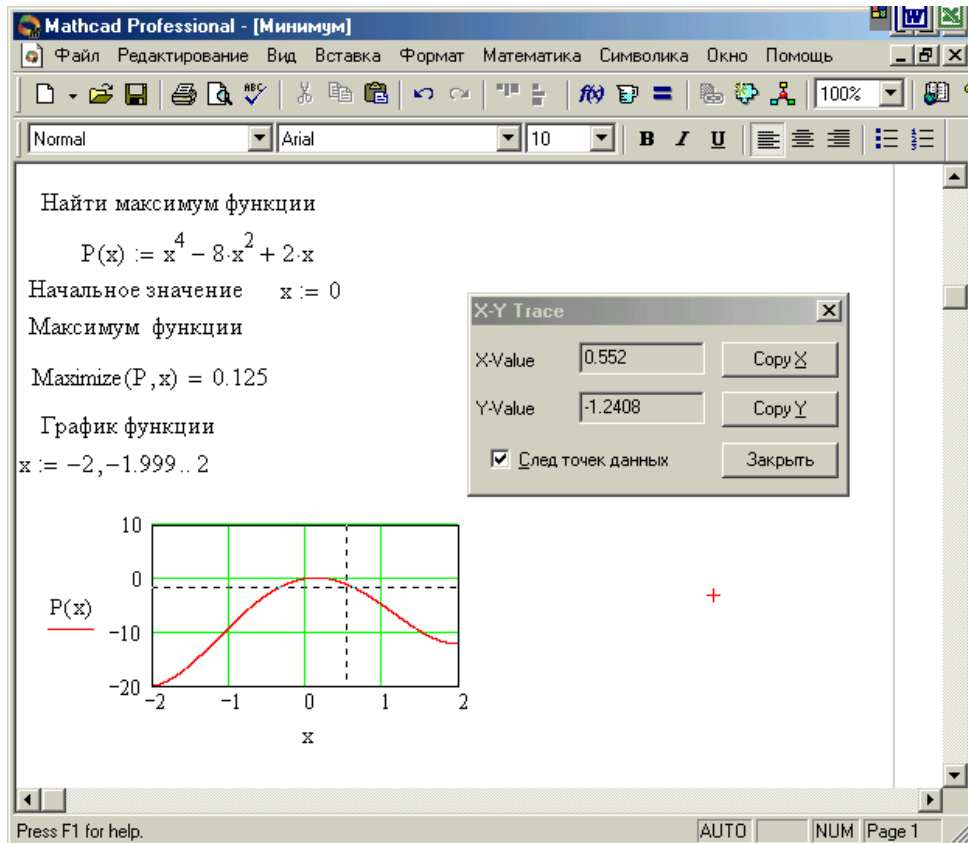


Рис. 4.14

Істотний вплив на результат чинить вибір початкового наближення, залежно від якості відповіді видаються різні локальні екстремуми. У цьому випадку чисельний метод іноді взагалі не приводить до позитивного результату.

Для більш наочного уявлення про характер поведіння функції іноді корисно побудувати її графік. Значення функції можливо побачити в будь-якій точці графіка. Для цього потрібно натиснути правою клавішею „миші” по графіку і у меню, що з’явилося, вибрати пункт **Трасування**.

Обчислення екстремуму функції багатьох змінних принципово не відрізняється від обчислення екстремуму функції однієї змінної. Приклад обчислення мінімуму функції двох змінних наведений на рис. 4.15.

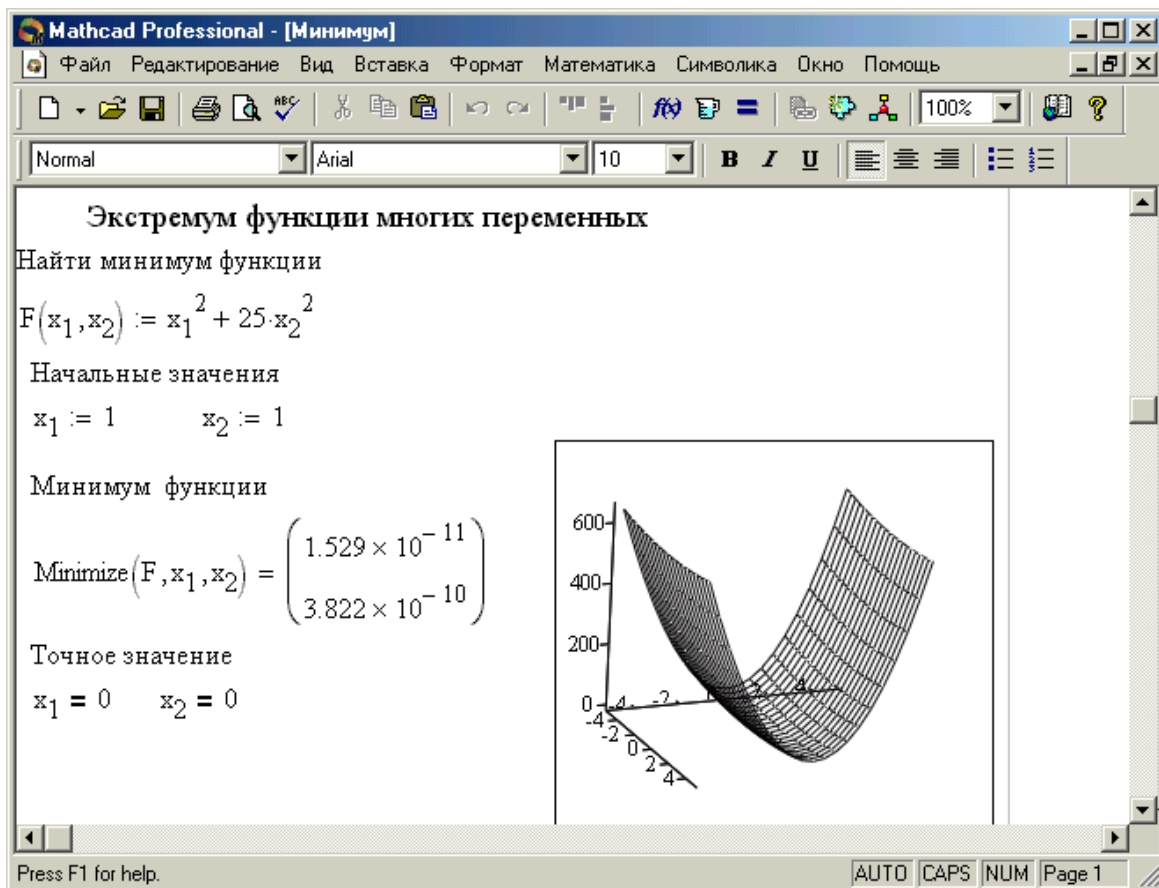


Рис. 4.15

При обчисленні умовного екстремуму функції повинні бути увімкнені в обчислювальний блок, що починається із ключового слова Given (рис. 4.16). Потім за допомогою булевих операторів записуються логічні вирази (нерівності, рівняння), що задають обмеження на значення аргументів функції, що мінімізується.

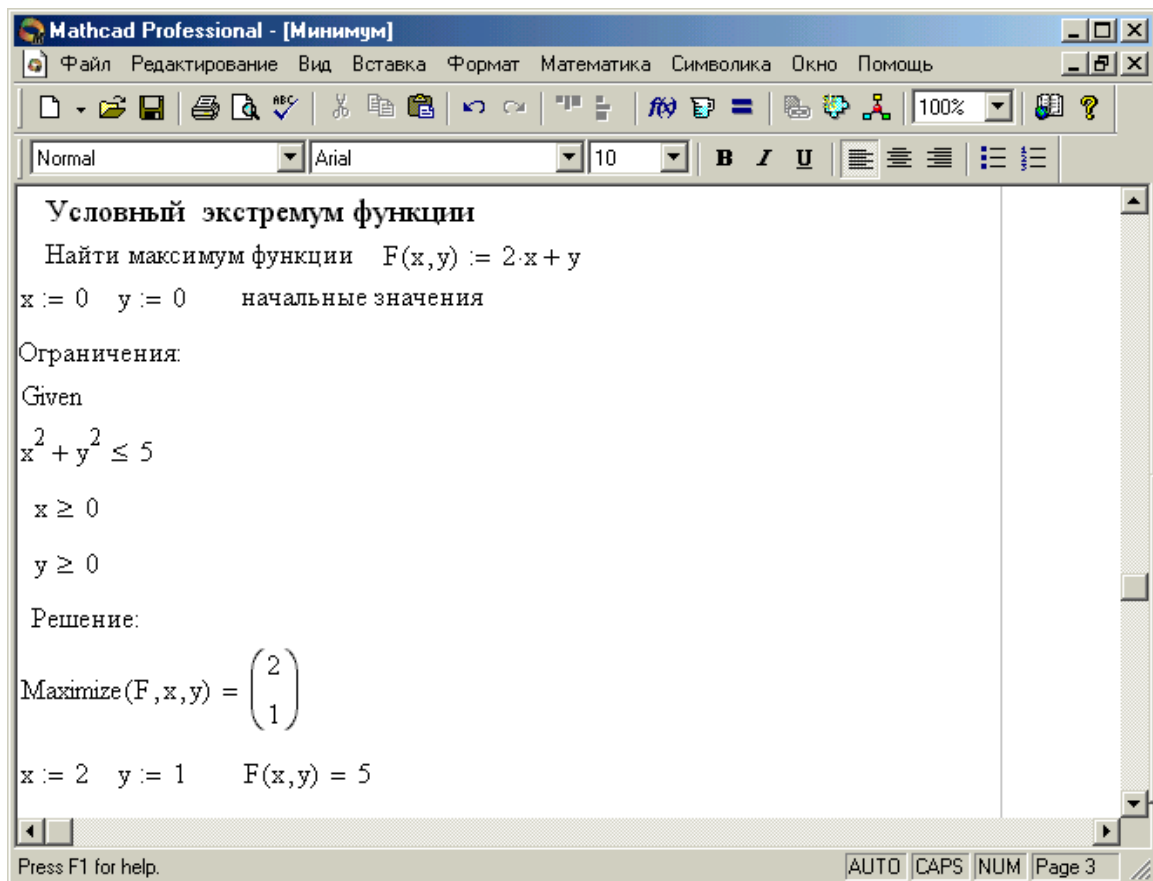


Рис. 4.16

Наприкінці блоку записується функція пошуку екстремуму. Mathcad дозволяє розв'язувати задачі лінійного програмування. Приклад розв'язання задачі, розглянутої в п. 4.3, показаний на рис. 4.17.

Для розв'язання задач пошуку екстремуму функцій можливо використовувати табличний процесор **Excel**. Розглянемо розв'язання задачі нелінійного програмування

Одна з надбудов **Excel** називається **Пошук розв'язання**. Відкрийте **Excel** і виберіть команду **Сервіс, Пошук розв'язання**. Якщо команда **Пошук розв'язання** відсутня у меню **Сервіс**, то для її встановлення необхідно вибрати команду **Сервіс,**

Надбудови, Пошук розв'язання. Якщо в діалоговому вікні **Надбудови** цієї надбудови немає, то необхідно звернутися до панелі керування, натиснути кнопку **Установлення й видалення програм** і установити надбудову **Пошук розв'язання**.

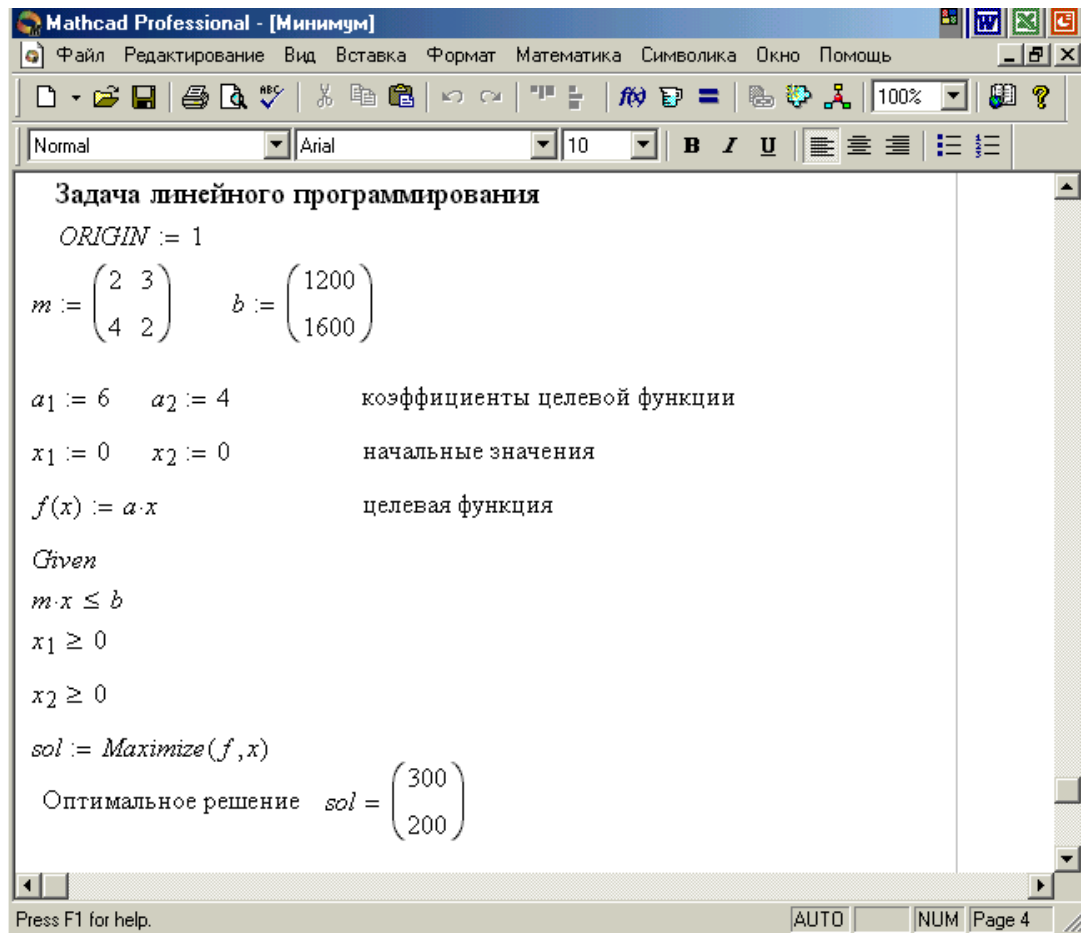


Рис. 4.17

На робочій сторінці **Excel** в комірках A3 й B3 будуть перебувати значення змінних x_1 й x_2 .

Введіть в комірку C4 цільову функцію $=(A3-4)^2+(B3-3)^2$

В комірки A7:A9 уведіть ліві частини обмежень:

$$=A3^2+B3^2.$$

A3

B3.

В комірки B:B9 введіть праві частини обмежень. Для пояснення вихідних даних і результатів у деякі комірки можна

ввести текст. У результаті робоча сторінка буде виглядати, як показано на рис. 4.18.

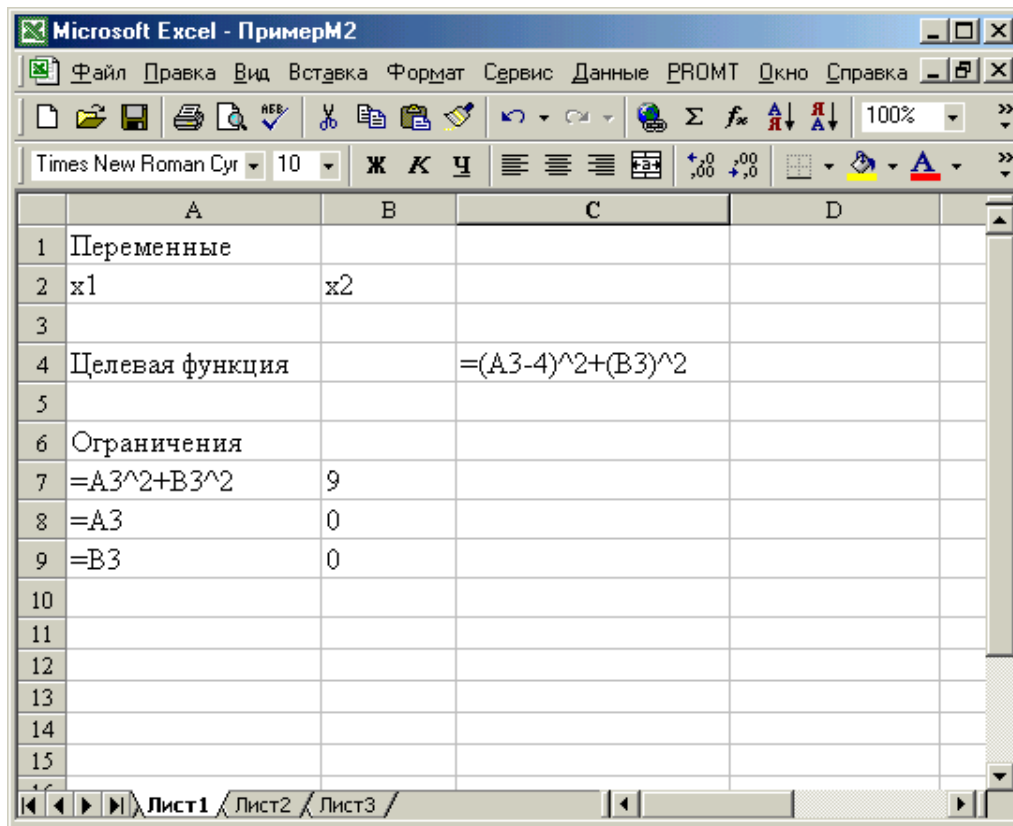


Рис. 4.18

Натисніть комірку C4. Виберіть команду **Сервіс, Пошук розв'язання** й заповніть діалогове вікно, що відкрилося, **Пошук розв'язання**.

В поле **Встановити цільову комірку** дається посилання на комірку з функцією, для якої буде знаходитися максимальне значення. У групі **Рівної** встановіть перемикач у положення **Максимальному значенню**.

Перейдіть у поле **Змінюючи комірки** й вкажіть комірку, які повинні змінюватися в процесі пошуку розв'язання, тобто комірки, відведені для змінних. Зазначимо, що в діалоговому вікні **Пошук розв'язання** варто використовувати абсолютні посилання. Абсолютне посилання залишається незмінним при зміні адреси комірки. Це значить, що в поле **Встановити цільову комірку** повинна бути вказана адреса $\$C\4 .

Для введення обмежень натисніть кнопку **Додати** в діалоговому вікні **Пошук розв'язання** й у діалоговому вікні, що **відкрилося**, **Додати** обмеження заповніть поля. Натисніть кнопку **Додати** в діалоговому вікні **Додати обмеження** й введіть другу групу обмежень. Подібним чином уведіть інші обмеження.

Після цього діалогове вікно **Пошук розв'язання** повинно виглядати, як показано на рис. 4.19.

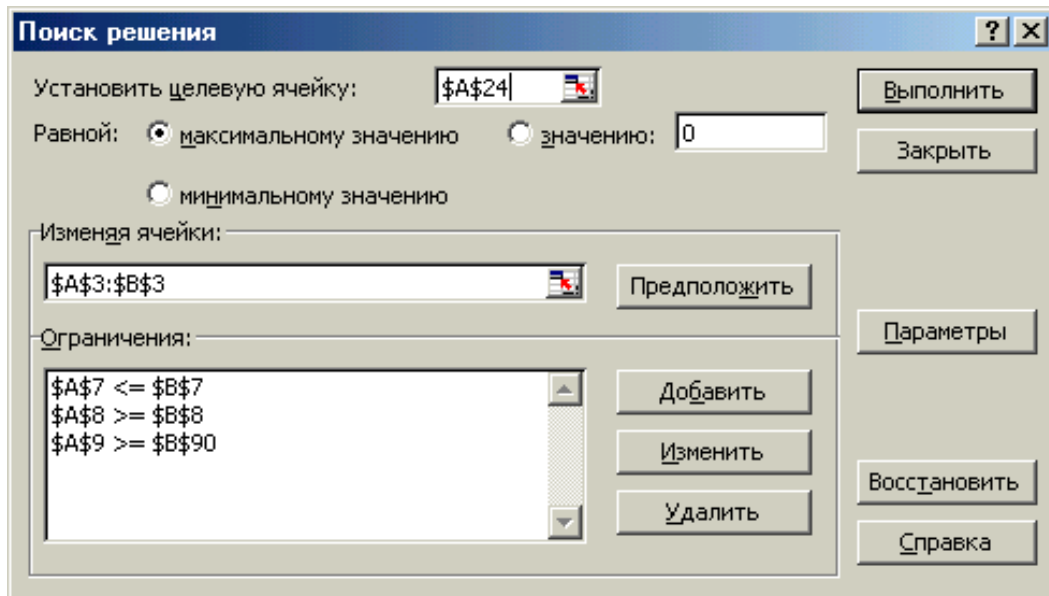


Рис. 4.19

Натисніть кнопку **Параметри** в діалоговому вікні **Пошук розв'язання** (рис. 4.19). У діалоговому вікні, що **відкрилося**, **Параметри пошуку** розв'язання (рис. 4.20) можна змінювати умови й варіанти розв'язання задачі, яка досліджується.

Для більшості задач підходять параметри, встановлені за замовчуванням. Поля **Відносна похибка** і **Допустиме відхилення** служать для задавання точності розв'язання. Рекомендується після знаходження розв'язання з параметрами, заданими за замовчуванням, повторити обчислення з більшою точністю й мінімальним припустимим відхиленням і порівняти результати. Це особливо важливо для розв'язання задач із вимогою цілочисельності змінних.

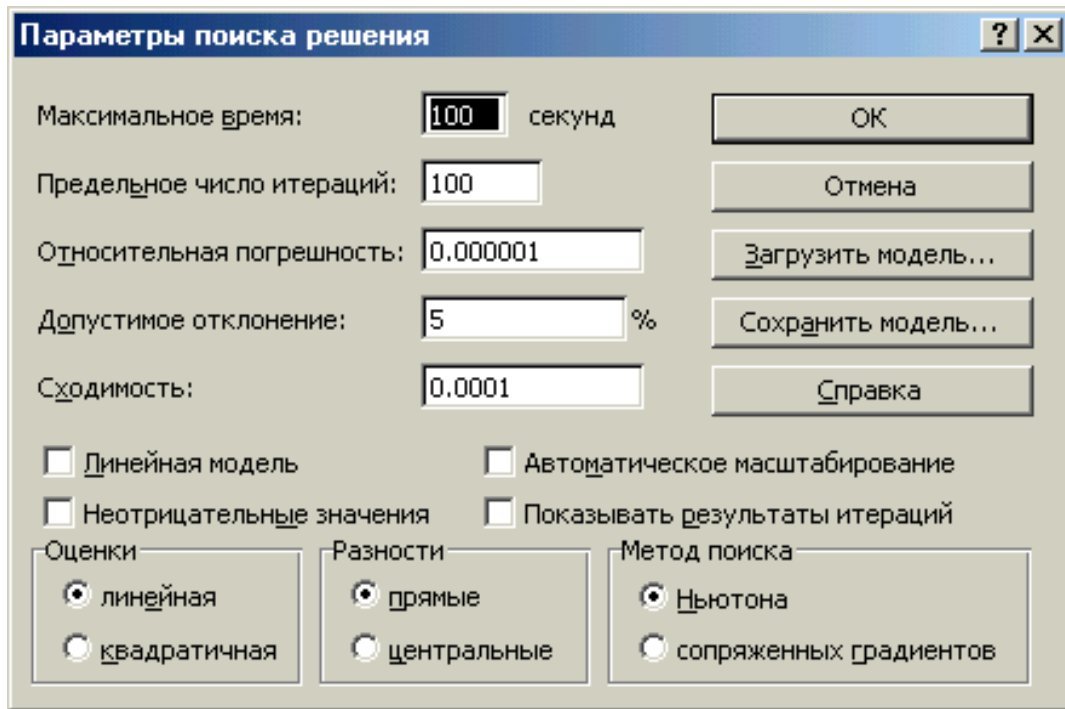


Рис. 4.20

Прапорець **Лінійна модель** служить для пошуку розв'язання лінійної задачі оптимізації або лінійної апроксимації нелінійної задачі. У випадку нелінійної задачі цей прапорець повинен бути скинутий, у випадку лінійної задачі - встановлений, тому що в противному випадку можливе отримання неправильного розв'язку.

Після встановлення параметрів варто повернутися в діалогове вікно Пошук розв'язання й натиснути кнопку **Виконати**. Відкриється вікно **Результати пошуку розв'язання**, яке повідомляє, що розв'язання знайдене (рис. 4.21).

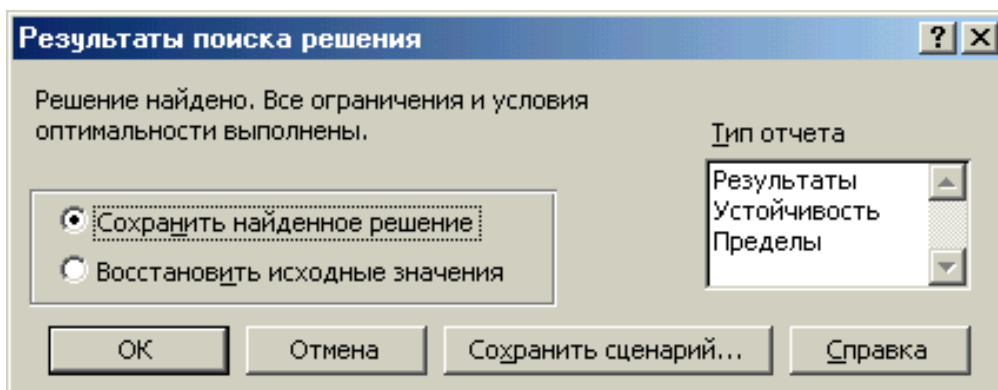


Рис. 4.21

Для виведення звіту про результати розв'язання задачі виберіть у діалоговому вікні **Результати пошуку розв'язання** необхідний тип звіту: **Результати, Стійкість і Межі**.

Для виведення результатів звіту **Excel** створює нову робочу сторінку, якій автоматично присвоюється ім'я **Звіт за результатами**.

Звіт про результати розв'язання показаний на рис. 4.22. У першій частині звіту вміщуються відомості про цільову функцію. У стовпці **Вихідно** перебуває значення цільової функції до початку обчислень, у рядку **Результат** - значення, отримане в результаті розв'язання задачі. Друга частина таблиці містить значення шуканих змінних, отриманих в результаті розв'язання задачі. Зі звіту видно, що максимальне значення $x_1 = 0$, $x_2 = 0$.

Microsoft Excel 9.0 Отчет по результатам

Рабочий лист: [ПримерМ2.xls]Лист1

Отчет создан: 18.05.2006 17:08:34

Целевая ячейка (Максимум)

Ячейка	Имя	Исходно	Результат
\$C\$4	Целевая функция	25	25

Изменяемые ячейки

Ячейка	Имя	Исходно	Результат
\$A\$3	x1	0	0
\$B\$3	x2	0	0

Ограничения

Ячейка	Имя	Значение	формула	Статус	Разница
\$A\$7	Ограничения	0	\$A\$7<=\$B\$7	не связан.	9
\$A\$8	Ограничения	0	\$A\$8>=\$B\$8	связанное	0
\$A\$9	Ограничения	0	\$A\$9>=\$B\$9	связанное	0

Отчет по результатам 1 / Лист1 / Лист2 / Ли

Рис. 4.22

5. ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ НА БАЗІ СИСТЕМИ AUTOCAD

5.1. Вікно графічного редактора

Серед систем автоматизованого проектування визнаним лідером є AutoCAD.

AutoCAD являє собою мікрокомп'ютерну систему автоматизації креслярських робіт. Швидкість і легкість, з якою можуть бути виконані підготовка й модифікація креслення з використанням обчислювальної системи, забезпечує істотну економію часу в порівнянні з «ручним» кресленням.

Запустити AutoCAD в операційній системі Windows можливо декількома способами:

- на робочому столі Window знайти значок AutoCAD і активізувати його подвійним натисканням „миші”;
- натиснути кнопку **Пуск**, відкрити меню **Програми** й вибрати пункт AutoCAD.

Можливі два варіанти початку роботи:

- у вікні Select template (Вибір шаблону);
- у вікні Startup (При завантаженні).

У головному меню виберіть Інструменти - Параметри, а потім у діалоговому вікні Опції, яке відкрилося, на вкладці Система виберіть у списку Автозавантаження необхідний варіант запуску. Якщо ви хочете відразу починати роботу і не використати шаблони, виберіть Do not show a startup dialog.

Після того, як AutoCAD запущений на виконання, з'явиться основне вікно додатка, показане на рис. 5.1. Воно має ту ж структуру, що й більшість додатків Windows.

Зверху вниз розташовується заголовок вікна, рядок меню й стандартна панель інструментів. Центральна область вікна називається графічною або робочою областю. У лівій частині робочого поля розташовується значок системи координат.

Рис. 5.1

У нижній частині вікна знаходиться область командних рядків, у якій за замовчуванням міститься три рядки. Сюди вводяться команди AutoCAD.

У розпорядження користувача надається безліч панелей. З лівого краю вікна розташовані панелі **Draw** (Креслення) і **Modify** (Зміна). Панелі за допомогою „миші” можна перетягувати й розташовувати з іншого краю вікна. Панель **Draw** містить інструменти для креслення ліній, дуг, окружностей і т.п. На панелі **Modify** знаходяться кнопки, за допомогою яких виконуються певні дії з елементами креслення: видалити, перемістити, повернути й т.д.

Команди AutoCAD можна вводити декількома способами:

- у головному меню вибрати Малювання і потім необхідну команду;
- на панелі інструментів натиснути кнопку;

- вибрати команду в екранному меню;
- ввести команду із клавіатури.

Для виведення екранного меню необхідно вибрати в головному меню **Інструменти – Параметри**. У діалоговому вікні **Опції** перейти на вкладку **Екран** й установити прапорець **Параметри** екранного меню. Після цього праворуч (за замовчуванням) з'явиться екранне меню (рис. 5.2). Якщо натиснути на будь-яку кнопку, з'явиться меню нижнього рівня. Для повернення до кореневого меню потрібно вибрати у верхній частині екранного меню елемент **AutoCAD** і виконати клацання.

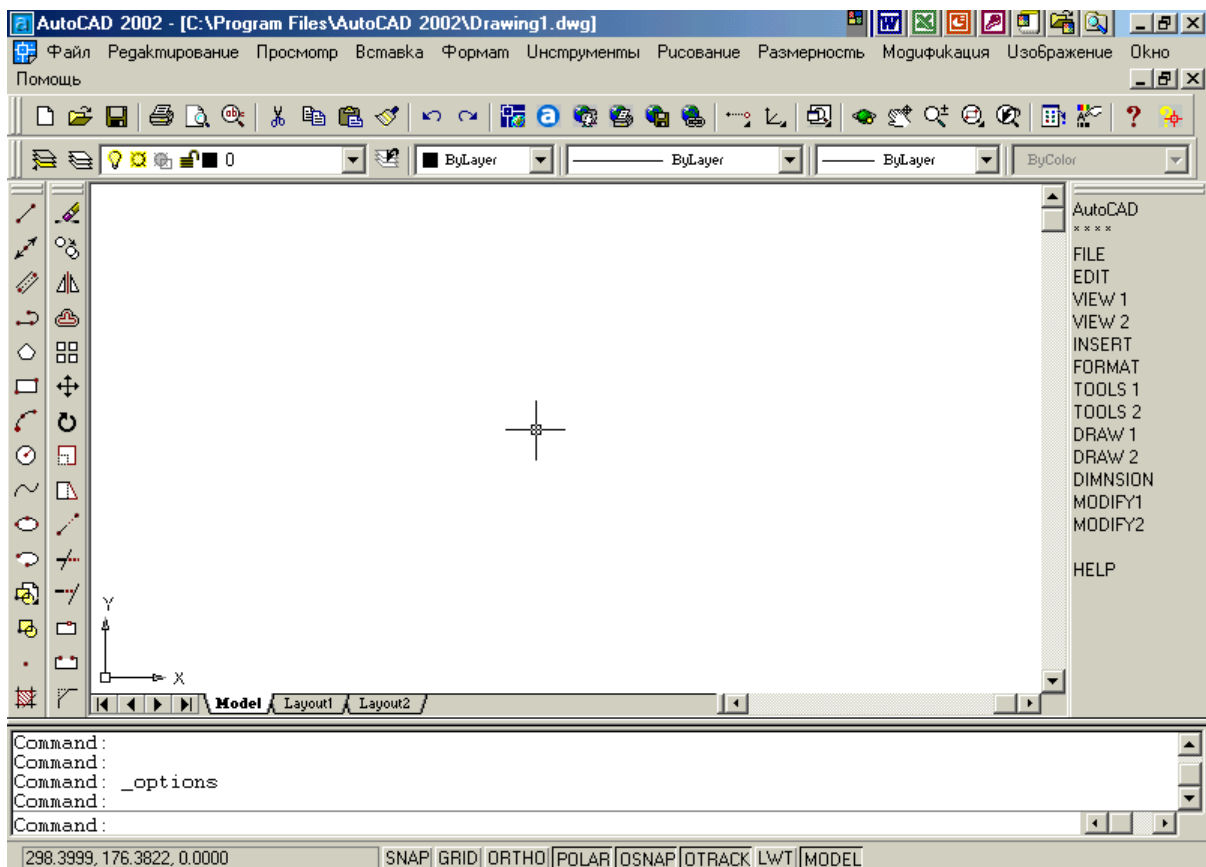


Рис. 5.2

Вибір способу введення команд визначається перевагами користувача.

Для введення команд AutoCAD можна використати контекстне меню.

5.2. Системи координат

У системах автоматизованого проектування виробів машинобудування одне із центральних місць належить геометричній моделі конструкції, яка розробляється. Комп'ютерні методи при розв'язанні геометричних задач ґрунтуються на застосуванні математичного апарата аналітичної геометрії, чисельних методів обробки, булевої алгебри й т.д.

Геометричні задачі комп'ютерного моделювання можна умовно розділити на дві групи: а) формування й подання геометричних моделей; б) синтез моделей на різних етапах проектування.

При конструюванні обмін інформацією здійснюється на основі ескізних і робочих креслень із використанням нормативно-довідкової і технічної документації. На цьому етапі геометричні моделі мають допоміжне значення. Істотне значення геометричні моделі мають на заключному етапі робіт для відображення результатів розроблення у вигляді креслень, графіків і схем.

З появою систем автоматизованого проектування традиційний підхід до етапів проектування зазнав змін. Ескізне проектування, розрахунки, видача результатів розроблень тепер виконуються в єдиній геометричній формі. У цих умовах креслення починає відігравати допоміжну роль, а методи його створення засновані на методах комп'ютерної графіки.

Для зручності формування геометричних об'єктів використовуються такі види систем координат:

- а) ортогональна декартова (двовимірна) - при формуванні креслень і інших графічних документів;
- б) афінна - для побудови спеціальних зображень (наприклад, для створення аксонометричних проєкцій);
- в) полярна - для зображення поверхонь;
- г) циліндрична - для формування циліндричних тіл;
- д) сферична.

При геометричному моделюванні просторових об'єктів використовується тривимірна система, а при формуванні плоских - двовимірна система координат.

В AutoCAD використовується два види систем координат: світова система координат і одна або більше користувальницьких (локальних) систем координат (рис. 5.3). Можливо створювати всі об'єкти у світовій системі координат. Так як об'єкти можуть займати в просторі довільне положення, то приходиться постійно перераховувати координати точок, що вводять. Точки об'єкта зручніше подавати в локальній системі координат, пов'язаній з деякою базовою точкою об'єкта. Ця система координат може перебувати в будь-якій точці простору й повернена щодо світової системи координат під будь-яким кутом.

Для визначення нової координатної площини необхідно викликати команду UCS:

```
Command (Команда): UCS
Current ucs name: *TOP* (Ім'я поточної
користувальницької системи координат: *Верхня*).
Enter an option
[New/Move/orthoGraphic/Prev/Restore/Save/Del/Apply/?/
World] <World> (Введіть параметр
[Нова/Перемістити/Ортогональна/Попередня/Відновити/
Зберегти/Видалити/
Застосувати/?/Абсолютна] < Абсолютна >):
```

Параметри командного рядка мають такі значення:

- New** – створює нову користувальницьку систему координат;
- Move** – переміщає користувальницьку систему координат уздовж осі Z;
- orthoGraphic** – вибирає стандартну ортогональну систему координат;
- Prev** – відновлює попередню користувальницьку систему координат;
- Restore** – відновлює іменовану користувальницьку систему координат;
- Save** – зберігає поточну користувальницьку систему координат під зазначеним ім'ям;
- Del** – видаляє ім'я збереженої користувальником системи координат;

Apply – застосовує налаштування користувальницької системи координат щодо обраного видового екрана або всіх активних видових екранів;

? - відображає список збережених користувальницьких систем координат;

World – вирівнює користувальницьку систему координат щодо абсолютної системи координат.

Якщо користувач прийняв рішення щодо створення нової системи координат, у командному рядку варто набрати N. Після натискання клавіші з'явиться запрошення задати параметри:

Specify origin of new UCS or

[Zaxis/3point/Object/Face/View/X/Y/Z] <0, 0, 0>: (Виберіть центр системи координат або [ОсьZ/3точки/Об'єкт/Грань/Вид/ X/Y/Z] <0, 0, 0>: (Введіть параметри).

Параметр **New** може мати одне з таких значень:

Specify origin of new UCS – переміщає початок користувальницької системи координат у нову точку;

Zaxis – переносить початок нової користувальницької системи координат у зазначену користувачем тривимірну точку й направляє вісь Z відповідно до іншої зазначеної користувачем точки;

3point – задає положення нової користувальницької системи координат у просторі (перша зазначена користувачем тривимірна точка – початок, друга визначає позитивний напрямок осі X, третя - позитивний напрямок осі Y);

Object – сполучає користувальницьку систему координат з обраним об'єктом;

Face - сполучає користувальницьку систему координат із гранню тривимірного твердого тіла;

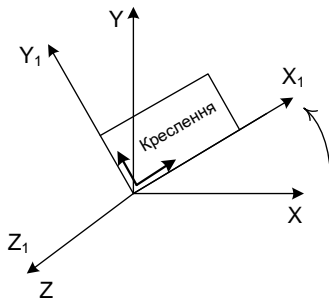
View - сполучає користувальницьку систему координат з поточним видом;

X – повертає систему координат щодо осі X;

Y - повертає систему координат щодо осі Y;

Z - повертає систему координат щодо осі Z.

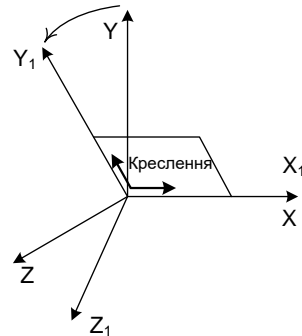
Поворот локальної системи координат
навколо осі Z



Command: UCS
Current ucs name: *Тор* (Ім'я поточної системи координат: *Верхня*)
Enter an option [New/Move/orthoGraphic/Prev/Restore/Save/Del/Apply/?/World] <World>: N
Specify origin of new UCS or [Zaxis/3Point/Object/Face/View/X/Y/Z] <0,0,0>: Z

а)

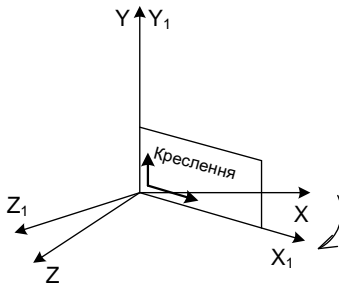
Поворот локальної системи координат
навколо осі X



Command: UCS
Current ucs name: *Тор* (Ім'я поточної користувачької системи координат: *Верхня*)
Enter an option [New/Move/orthoGraphic/Prev/Restore/Save/Del/Apply/?/World] <World>: N
Specify origin of new UCS or [Zaxis/3Point/Object/Face/View/X/Y/Z] <0,0,0>: X

б)

Поворот локальної системи координат
навколо осі Y



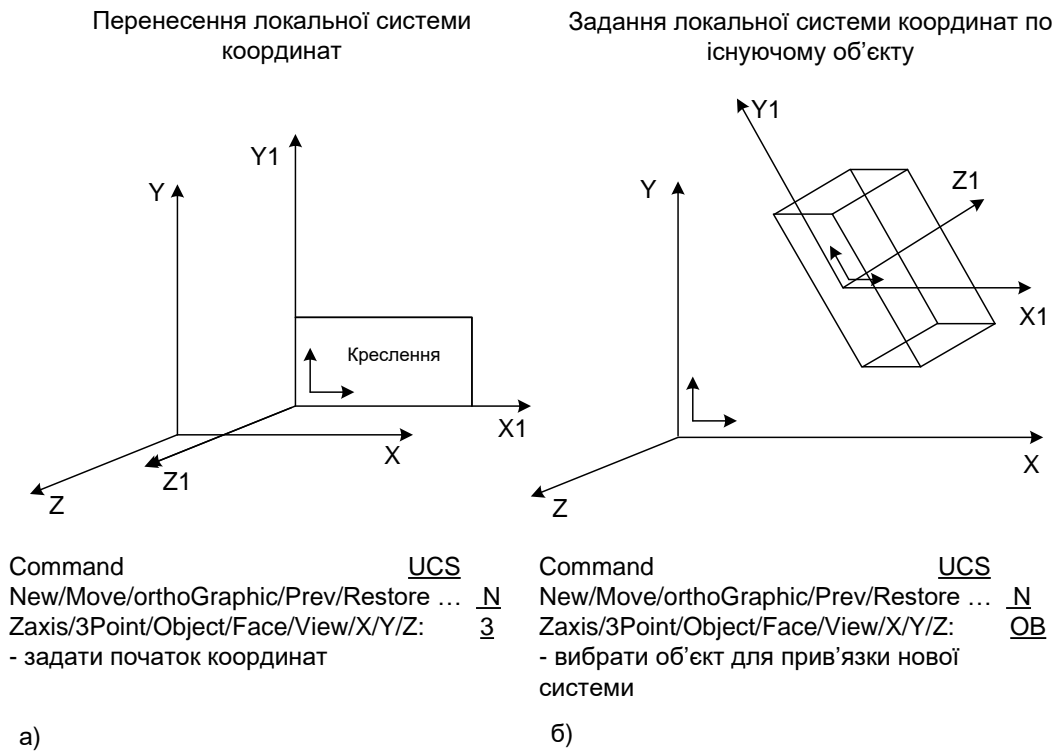
Command: UCS
Current ucs name: *Тор* (Ім'я поточної системи координат: *Верхня*)
Enter an option [New/Move/orthoGraphic/Prev/Restore/Save/Del/Apply/?/World] <World>: N
Specify origin of new UCS or [Zaxis/3Point/Object/Face/View/X/Y/Z] <0,0,0>: Y

в)

Рис. 5.3

При розробленні складних креслень можливо використовувати велику кількість локальних систем координат (рис. 5.4,в). Кожній системі координат можна присвоїти ім'я. Система координат, що використовується в цей момент, називається поточною системою координат. Для переходу до поточної системи координат необхідно виконати команду `UCS R <Ім'я>`. У кожній системі координат можна створювати двовимірні й тривимірні об'єкти.

Передбачається, що в поточній системі координат площина креслення збігається із площиною XOY (рис. 5.4,в). Тому зміна координатної системи приводить до перевизначення площини креслення.



Моделювання в декількох локальних системах координат

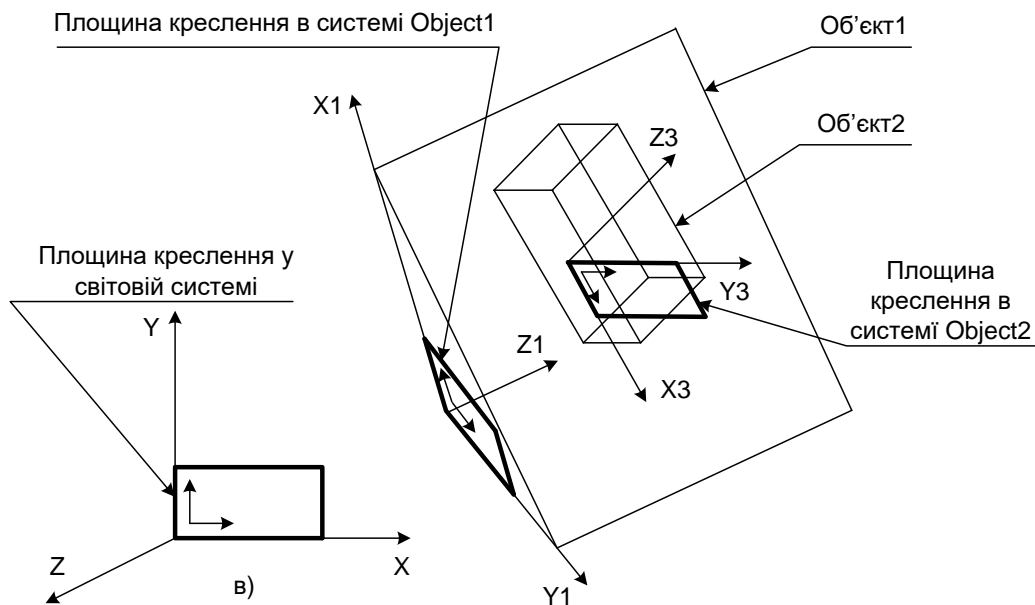


Рис. 5.4

5.3. Креслення об'єктів

Процес розроблення креслення включає креслення ліній, окружностей, дуг й інших елементів, які називаються об'єктами або примітивами. Для вказівки системі того, який примітив варто зобразити, вводять команди. Команди можуть бути набрані із клавіатури, обрані з екранного меню або введені шляхом натискання кнопки. Далі, відповідаючи на запити, що з'явилися в командному рядку, вводять певні параметри.

AutoCAD використовує п'ять способів задання точок у командах креслення і редагування:

X, Y - абсолютні координати в поточній користувальницькій системі координат;

@X,Y – зсув щодо останньої зазначеної точки;

@відстань<кут з віссю X – зсув у полярних координатах щодо останньої зазначеної точки;

відстань, кут – уведення значення відстані щодо останньої точки, задання напрямку за допомогою „миші” й наступне натискання клавіші **<Enter>**;

щиглик „мишею” – вибір точки за допомогою „миші”.

Основним елементом кожного креслення є лінія. Для її креслення використовується команда **Line**. Розглянемо креслення ліній на простих прикладах.

Нехай потрібно побудувати лінію, що починається в точці з координатами (10,20) і закінчується в точці (40,40).

Введемо в командному рядку команду **Line**. Після натискання на клавішу **Enter** у командному рядку з'явиться діалог:

```
Command: line  
Specify first point:
```

Введемо в командний рядок значення 10,20 і натиснемо на клавішу **Enter**. З'явиться таке повідомлення:

```
Command: line  
Specify first point: 10,20  
Specify next point or [Undo]:
```

Уведемо в командний рядок значення 40,40 і натиснемо на клавішу Enter. У командному рядку знову з'явиться повідомлення

Specify next point or [Undo]:,

тому що система готова продовжувати креслення. Натиснемо на клавішу Enter і завершимо побудову відрізка.

Остаточо діалог буде виглядати так:

Command: line

Specify first point: 10,20

Specify next point or [Undo]: 40,40

Specify next point or [Undo]:

Перейдемо тепер до більш складного завдання. Побудуємо прямокутник, координати вершин якого (10,50), (10,100), (50,100), (50,50). Будемо використовувати метод абсолютного задання координат.

Command: line

Specify first point: 10,50

Specify next point or [Undo]: 10,100

Specify next point or [Undo]: 50,100

Specify next point or [Close/Undo]: 50,50

Specify next point or [Close/Undo]: c

Операцію c виконують для того, щоб замкнути ламану.

Накреслимо п'ятикутник, використовуючи метод відносних прямокутних координат. Тут координати X й Y задаються щодо останньої точки. Запис @20,30 варто розуміти так: щодо останньої точки переміститися в напрямку осі X на 20 одиниць й у напрямку осі Y на 30 одиниць.

Command: line

Specify first point: 65,0

Specify next point or [Undo]: @35,10

Specify next point or [Undo]: @35,-10

Specify next point or [Close/Undo]: 0,-30

Specify next point or [Close/Undo]: c

Тепер намалюємо трикутник, використовуючи метод відносних полярних координат. У цьому випадку вказують точку, що розташовується на заданій відстані під заданим кутом.

Command: line

Specify first point: 60,50

Specify next point or [Undo]: @55<65

Specify next point or [Undo]: @55<-70

Specify next point or [Close/Undo]: c

Для завершення побудуємо окружність радіусом 40 із центром у точці (140,70). Викличемо команду **Circle** з панелі інструментів або за допомогою командного рядка.

Command: circle

Specify center point for circle or [3P/2P/Ttr/ (tan tan radius)]: 140,70

Specify radius of circle or [Diameter]: D

Specify diameter of circle: 40

На рис. 5.5 показані результати побудови об'єктів.

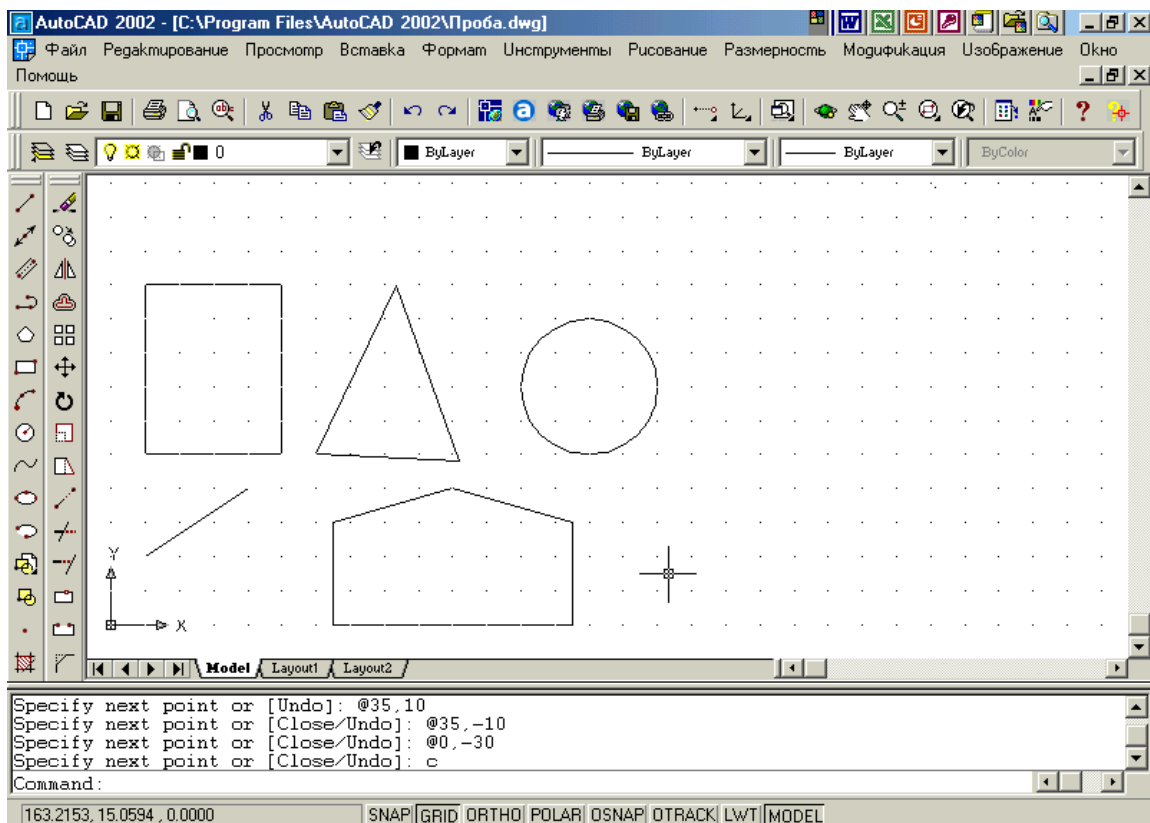


Рис. 5.5

5.4. Лінійні перетворення об'єктів

Перетворення називають лінійним, якщо воно має такі основні властивості:

а) відрізок прямої лінії перетвориться в новий відрізок прямої, площа - у нову площину;

б) якщо точка P , що належить відрізку, ділить його в заданому відношенні, то і після перетворення це відношення зберігається;

в) відношення площ і периметрів при лінійних перетвореннях зберігається;

г) три точки, що не лежать на одній прямій, перетворюються в три числа, що також не лежать на одній прямій, тобто існує єдине перетворення площини.

У системі AutoCAD використовуються такі типи перетворень об'єктів (рис. 5.6 - 5.9):

- а) паралельний перенос (**Move**);
- б) копіювання (**Copy**);
- в) розмноження у вигляді прямокутних масивів об'єктів (**Rectangular Array**);
- г) полярне розмноження (**Polar Array**);
- д) масштабування (**Scale**);
- е) поворот (**Rotate**);
- ж) дзеркальне відображення (**Mirror**);
- к) масштабування при вставленні в креслення заздалегідь складеного блоку (**Insert**);
- л) витягування (**Stretch**).

Перенос і копіювання після задання команди вимагають вказівки базової точки, щодо якої можна точно вказати вектор переносу. Для повороту й масштабування вказівка вибору положення базової точки дуже важлива, тому що впливає на нове положення об'єкта (рис. 5.6).

Лінійне перетворення об'єктів можна виконувати як на площині (2D), так й у тривимірному просторі (3D).

Приклади лінійних перетворень об'єктів наведені на рис. 5.6 - 5.9.

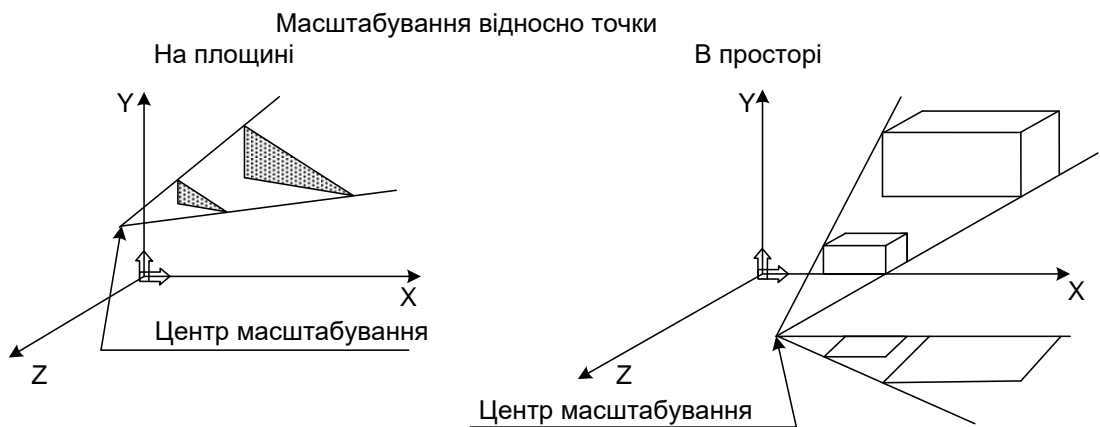
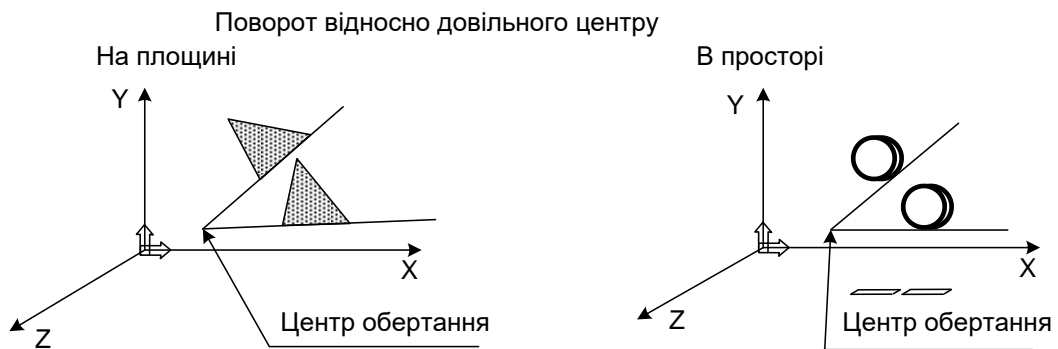
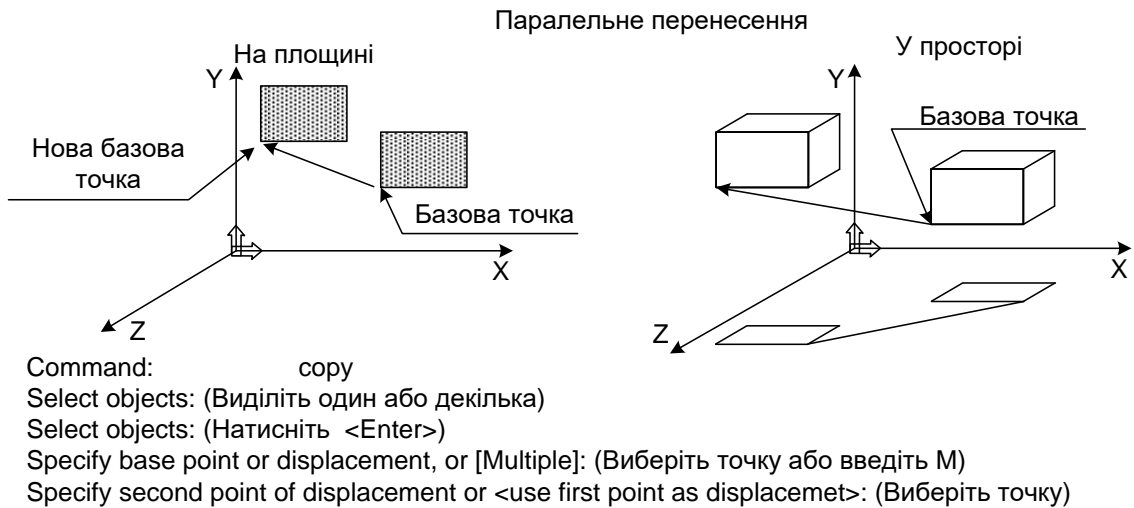
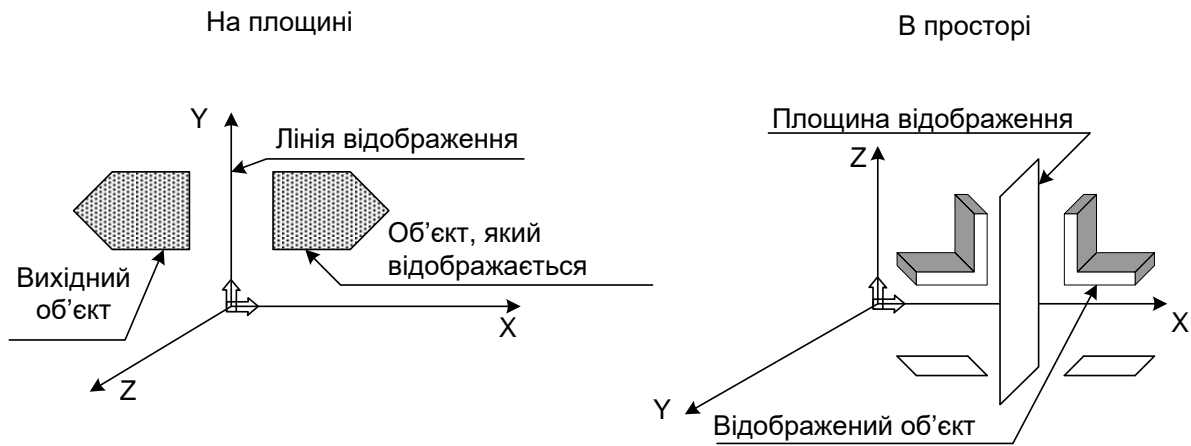


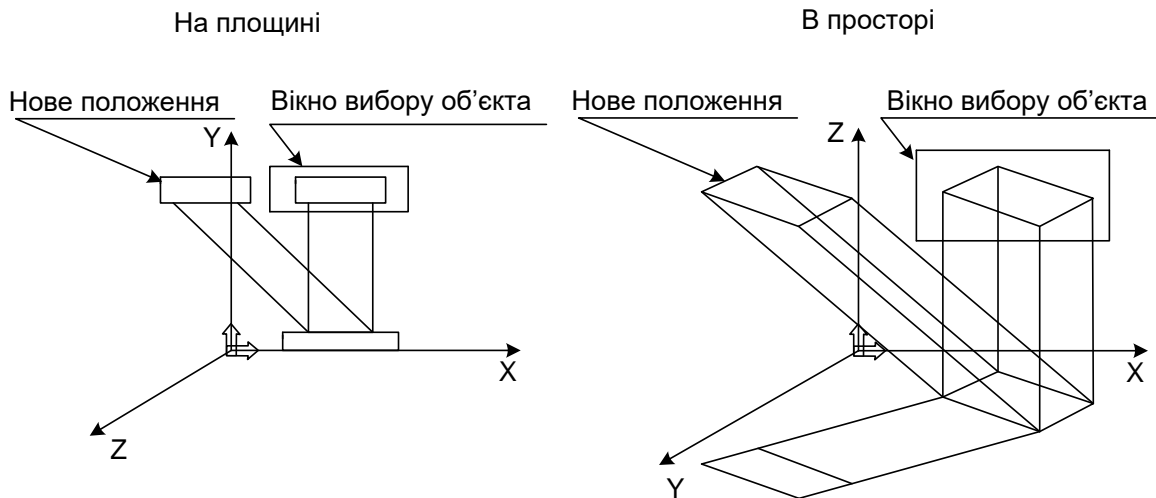
Рис. 5.6

Дзеркальне відображення



Command: mirror
Select objects: (Виділіть один або декілька об'єктів)
Select objects: (Натисніть <Enter> для завершення)
Specify first point of mirror line: (Виберіть точку)
Delete source objects? [Yes/No] <N>: (Введіть Y або N)

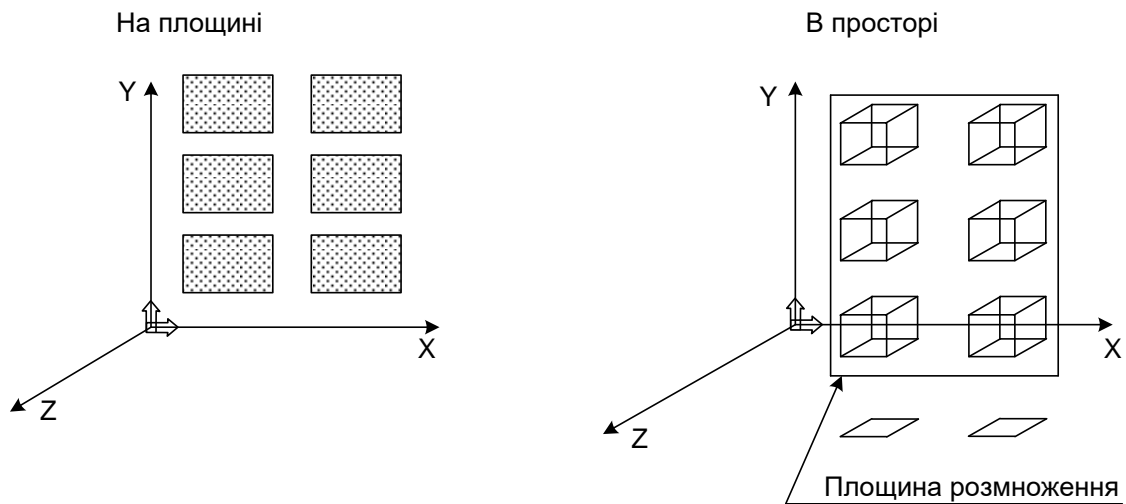
Протягування об'єкта



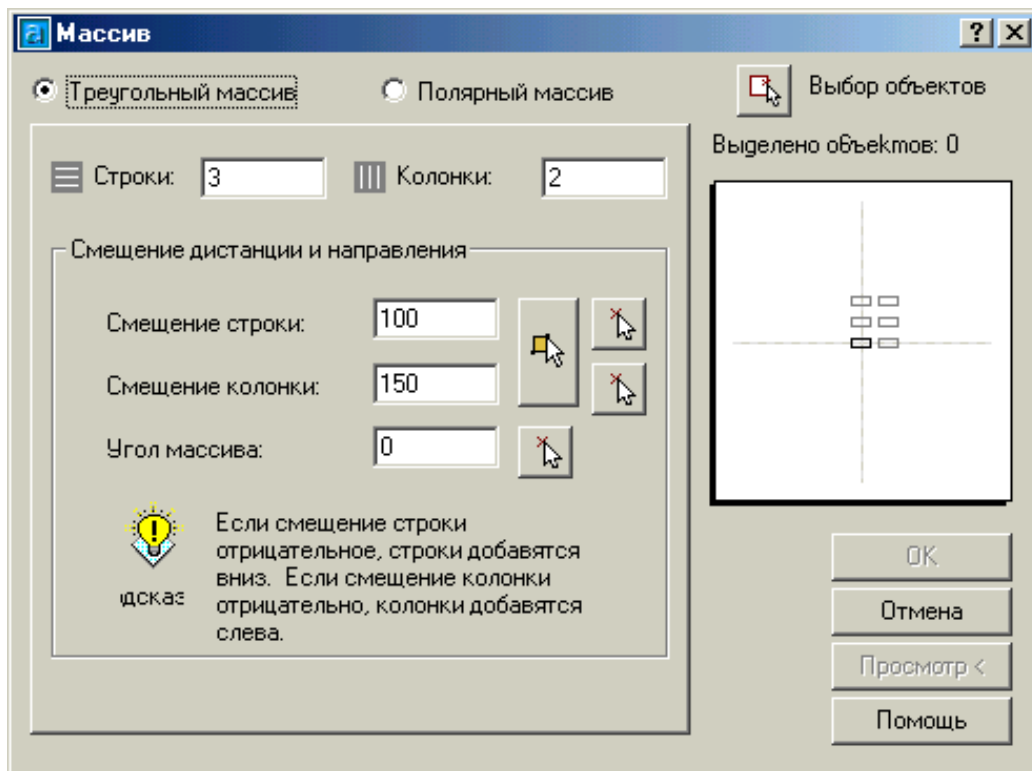
Command: stretch
Select objects to stretch by crossing-window or crossing-polygon...: (Виберіть об'єкти, які розтягуються за допомогою січної рамки або січного багатокутника...)
Select objects: (Введіть C або CP)
Specify first corner: (Виберіть точку)
Specify opposite corner: (Виберіть точку)
Select objects: (Натисніть <Enter> для завершення)
Specify base point or displacement: (Виберіть вихідну точку)
Specify second point of displacement or <use first point as displacement>: (Виберіть точку)

Рис. 5.7

Прямокутне розмноження (трикутний масив)



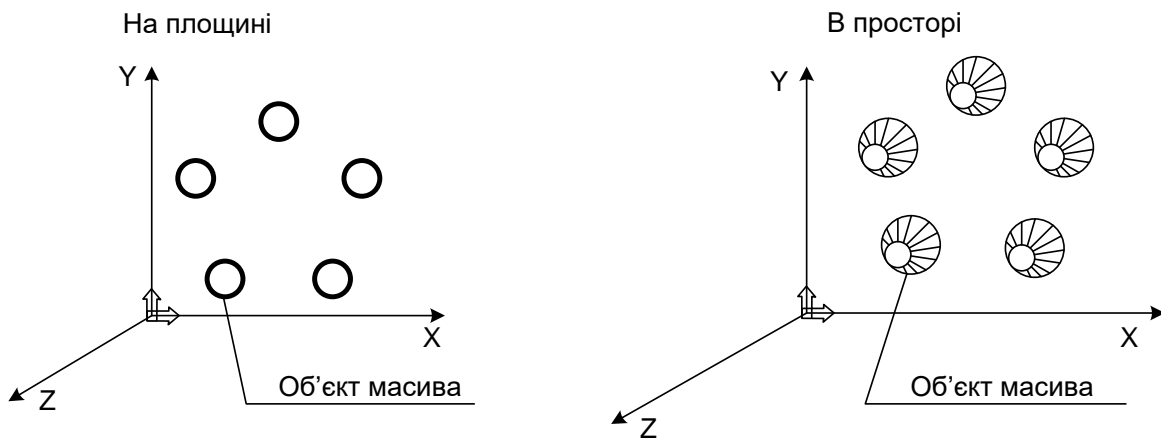
Command: array (відображення діалогового вікна)



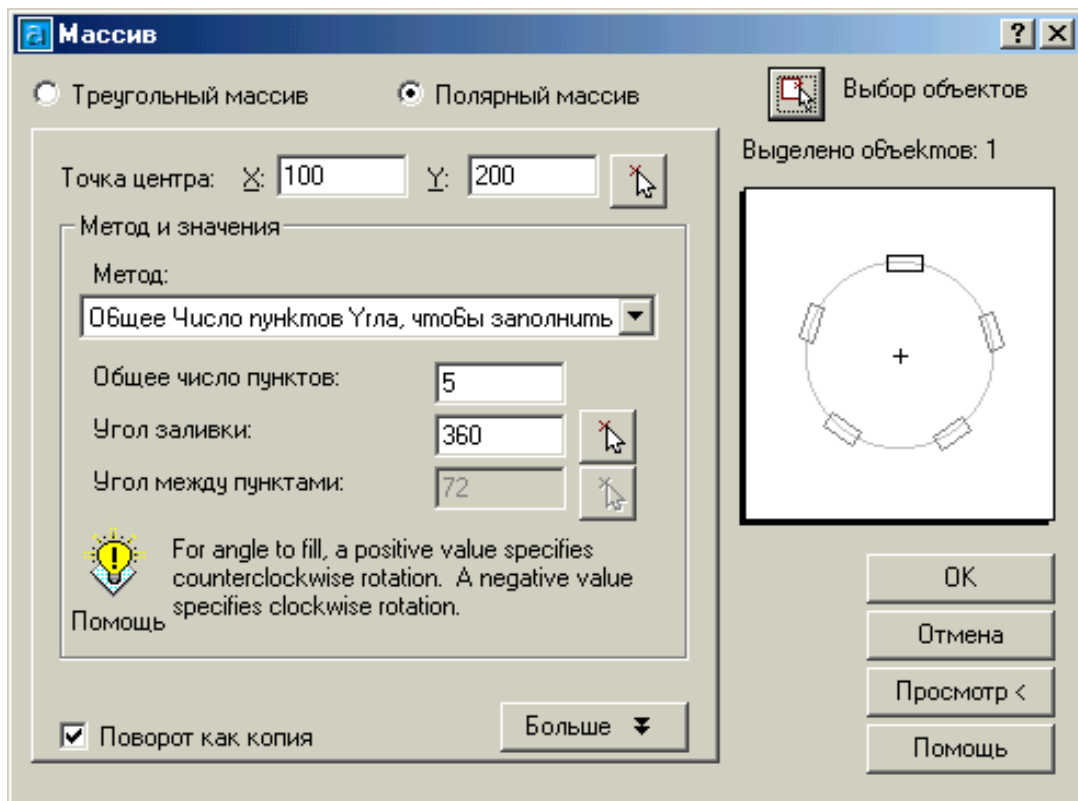
Command: array
 Select objects: (Виділіть один або декілька об'єктів)
 Select objects: (Натисніть <Enter>)
 Enter the type of array [Rectangular/Polar] <R>: (Виберіть R)
 Enter the number of rows (---) <1>: 3
 Enter the number of columns (| |) <1>: 2
 Enter the distance between rows or specify unit cell (---): 100
 Specify the distance between columns (| |): 150

Рис. 5.8

Полярне розмноження



Command: array (відображення діалогового вікна)



Command: array
 Select objects: (Виділіть один або декілька об'єктів)
 Select objects: (Натисніть <Enter>)
 Enter the type of array [Rectangular/Polar] <R>: (Виберіть P)
 Specify center point of array: 100, 200
 Enter the number of items in the array: 5
 Specify the angle to fill (+=ccw, -=cw) <360>: (Введіть значення або натисніть <Enter>)
 Rotate arrayed objects? [Yes/No] <Yes>: Y

Рис. 5.9

5.5. Методи формування плоских геометричних об'єктів

При описі й конструюванні плоских геометричних моделей застосовують два підходи:

1) метод точного аналітичного опису об'єктів за допомогою таких елементів, як пряма, окружність, криві другого порядку і їхньої композиції;

2) наближені методи з використанням інтерполяції й апроксимації. Моделі, одержувані наближеними методами, засновані на спеціальних методах, наведених у теорії сплайнів.

Плоскі моделі будуються за допомогою прямих, окружностей або дуг. У цьому випадку використовуються такі примітиви, як: **Point, Line, Pline, Arc, Circle, Poligon, Hatch**.

При формуванні складних плоских моделей використовують операції перетворення й геометричні обчислення. Необхідно мати на увазі, що візуальне рисування за допомогою курсора, яке використовують багато графічних редакторів, не забезпечує точність конструкції елементів моделі.

У геометричному моделюванні широко застосовуються наближені плоскі моделі. Різні складні криві будуються за сукупністю відомих точок, знайдених чисельними методами.

Для побудови наближеної кривої необхідно задати характеристичну ламану, яка в першому наближенні передає форму кривої, і виконати задану апроксимацію ламаної.

Для перетворення ламаної в криву використовують два методи: апроксимацію й інтерполяцію.

При використанні апроксимації отримана крива обов'язково повинна проходити через проміжні точки ламаної, але повинна завжди проходити через граничні точки, а функція визначається за граничними умовами і методом обчислення.

Інтерполяція є частковим випадком апроксимації. При цьому необхідний збіг вихідної характеристичної ламаної і інтерполяційної кривої в заданих точках.

У системі AutoCAD створювати й редагувати наближені криві можливо за допомогою команд **Spline, SplinElit, Pline** й **Pedit**.

При побудові таких кривих спочатку створюється характеристична ламана, проведена через задані точки (рис. 5.10,а). Для цього можна використати примітив **Pline**, що дозволяє формувати сукупність відрізків і дуг (рис. 5.10,б). Наприклад, можна створити примітив, що повністю складається з дуг й окружностей (рис. 5.10,в).

Математична обробка ламаної виконується командою **Pedit**. Команда має такі параметри для редагування двовимірної полілінії:

Open/Close/Join/Width/Edit vertex/Fit/Spline/Decurve/Ltype
gen//Undo

Опція F, яку можливо дати після вибору ламаної, дозволяє виконати інтерполяцію ламаної (рис. 5.11,в). При циклічній інтерполяції кожен сегмент ламаної буде замінений на дугу окружності. У режимі редагування вершин (опція E) можливо перевизначити напрямок дотичної до кривої в будь-якій вершині ламаної (рис. 5.11,б) або змінити положення вершин (рис. 5.10,в).

Команда **Pedit** у режимі редагування вершин (опція E) дає можливість локально перевизначити сплайн. При переміщенні вершин ламаної можна додати зміни кривої в межах між суміжними вершинами (рис. 5.11,е).

Для рисування нерівномірного B-сплайна можна використати команду **Spline**:

Command: spline

Specify first point or [Object]: (Виберіть точку або введіть O)

Specify next point: (Виберіть точку)

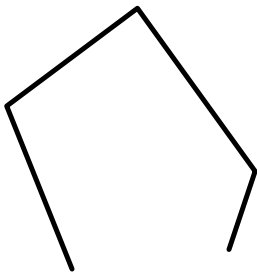
Specify next point or [Close/Fit tolerance] <start tangent>: (Виберіть точку або введіть параметр)

Specify next point or [Close/Fit tolerance]: (Натисніть <Enter> для визначення точки дотичної)

Specify start tangent: (Виберіть точку)

Specify end tangent: (Виберіть точку)

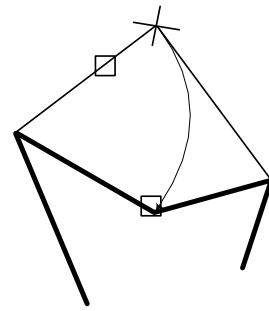
Примітив «Ламана»



Command pline
Specify start point: (Виберіть початкову точку)
Specify next point or [Arc/Halfwidth/Length/Undo/Width]: (Виберіть точку або введіть параметр)
Specify next point or [Arc/Close/Halfwidth/Length/Undo/Width]: (Виберіть точку або введіть параметр)

а)

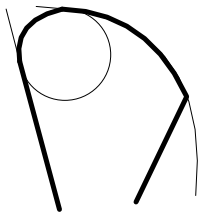
Переміщення вершини «Ламаної»



Command pedit
Select poliline : - вказати маркером
Open/Close/Join/Width/Edit vertex/Fix/Spline/Decurve/Ltype gen/Undo: E
Next/Prev/Move ... M
- вказати маркером нову вершину

г)

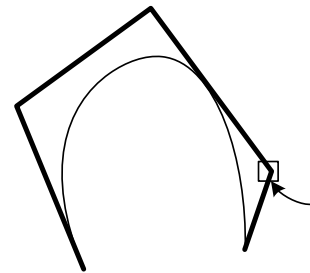
Примітив «Ламана», складено з сегментів ліній і дуг окружностей



Command pline
Перехід із режиму ліній в дуги A
Перехід із режиму дуг в лінії L

б)

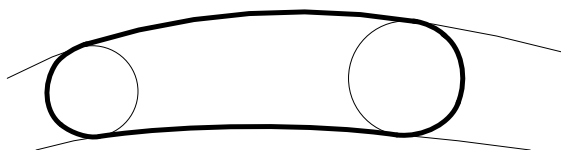
Сплайн-апроксимація «Ламаної»



Command pedit
Select poliline - вказати маркером
Open/Close/Join ... S - сплайн

д)

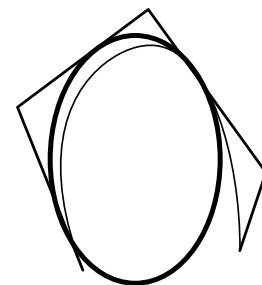
Примітив «Ламана з дуг окружностей»



Command pline
Start point:
- перейти в режим дуг і виконати побудову

в)

Замикання сплайн - апроксимації

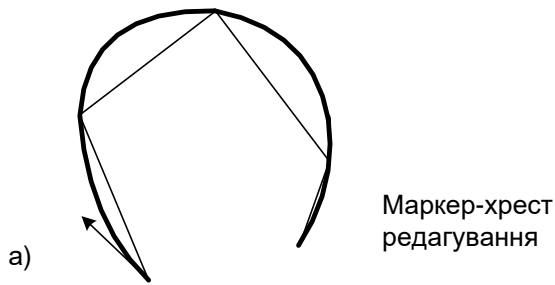


Command pedit
Select poliline - вказати маркером
Open/Close/Join ... C - замкнути

ж)

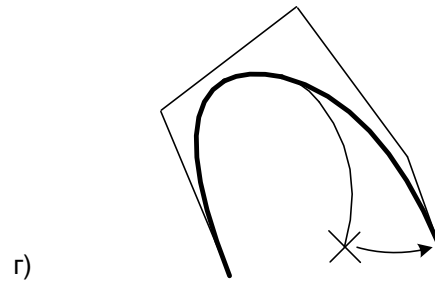
Рис. 5.10

Циклічна інтерполяція примітива полілінії дугами окружності



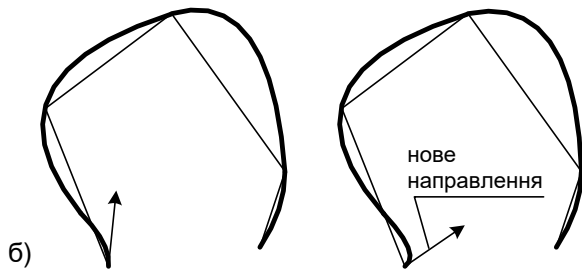
Command pedit
 Select poliline - вказати курсором полілінію
 Open/Close/Join/Width/Edit vertex ... E

Модифікація вершин сплайн-апроксимації

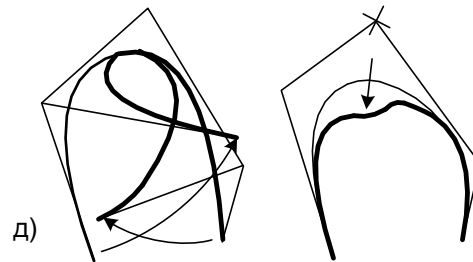


Command pedi
 Select poliline : - вказати курсором полілінію
 Open/Close/Join ... E - редактор - маркером-хрестом найти потрібну вершину, дати команду на перенесення
 Next/Prev/Move ... M
 - вказати курсором нову вершину

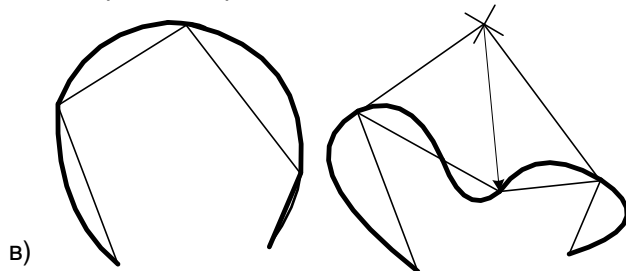
Модифікація вихідного напрямку дуги



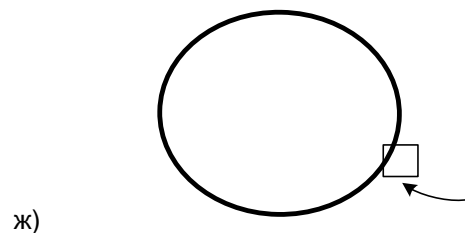
Command pedit
 Select poliline
 Open/Close/Join/Width/Edit vertex ... E
 Next/Previous/Break/Insert ... D
 - вказати курсором новий напрям



Модифікація вершини полілінії за положенням



Command pedit
 Select poliline - вказати курсором
 Open/Close/Join/Width ... E - редагувати
 Next/Previous/Break/Insert/Move ... M - рухати
 - вказати курсором нову вершину



Command area
 Specify first corner ... O - вказати курсором
 Select objects: - вказати курсором
 Area = 370280.95, Perimeter = 789.03

Рис. 5.11

Редагування сплайнів виконують командою **SplinEdit**. Основні опції команди:

Fit data/Close/Move vertex/Refine ...

Вибір параметрів командного рядка забезпечує замикання відкритого сплайна, переміщення керуючої вершини, розмикання замкнутого сплайна, додавання керуючої точки, зміна напрямку сплайна і т.д.

5.6. Об'єктна прив'язка

При побудові геометричних моделей виникають завдання, пов'язані з геометричними розрахунками. Вирішуються два види завдань: позиційні й метричні.

Позиційні завдання полягають у визначенні характерних точок об'єктів і точок їхнього перетинання або торкання. Метричні завдання пов'язані з визначенням координат точок, відстаней, метричних характеристик об'єктів (периметрів, площ, моментів інерції й т.п.).

Координати точок не завжди зручно вводити із клавіатури. Часто точки визначаються на основі геометричних побудов. Для цього використовують спеціальний режим - об'єктну прив'язку. У цьому випадку на екрані з'являється спеціальний маркер і замість точного задання координат характерної точки досить указати на примітив або об'єкт із точкою, що цікавить Вас.

Використовується два режими об'єктної прив'язки: поточна (постійна) і одинична (тимчасова).

Одинична об'єктна прив'язка здійснюється вказівкою типу об'єктної прив'язки при запиті координат точки й поширюється тільки на цей запит.

Поточна об'єктна прив'язка здійснюється командою **Osnap** із вказівкою типу об'єктної прив'язки. Вона задається при необхідності кількарізного використання характерних точок одного типу. Для задання параметрів поточної об'єктної прив'язки необхідно ввести команду **Osnap**, вибрати команду **Drafting Settings** (Параметри креслення), відкрити вкладку **Object Snap** (Об'єктна прив'язка) і відзначити три або чотири режими прив'язки одночасно (рис. 5.12).

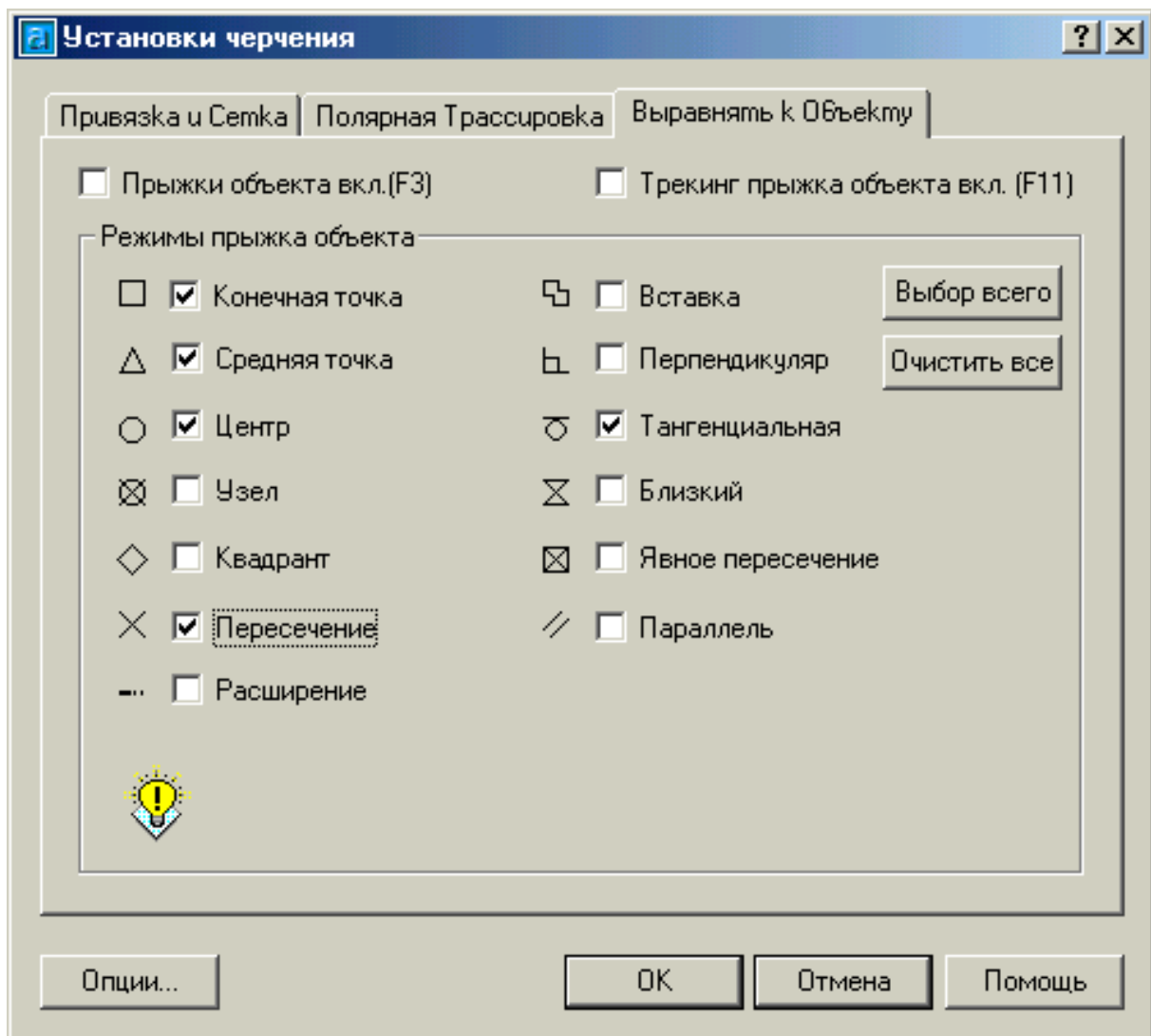


Рис. 5.12

Використовуються такі типи об'єктної прив'язки:

ENDpoint – введіть **end** для об'єктної прив'язки до кінцевої точки лінії, полілінії й дуги;

MIDpoint – введіть **mid** для об'єктної прив'язки до середньої точки ліній і дуг;

CENter – введіть **cen** для об'єктної прив'язки до центральної точки дуги або кола;

NODe – введіть **nod** для об'єктної прив'язки до зазначеного об'єкта;

QUAdrAnt – введіть **qua** для об'єктної прив'язки до точки квадранта кола або дуги;

INTersection – введіть **int** для об'єктної прив'язки до точки перетинання двох об'єктів, до самоперетинаючого об'єкта або об'єктів, які будуть перетинатися при продовженні;

PERpendicular – введіть **per** для автоматичної побудови перпендикуляра, проведеного до об'єкта;

TANgent – введіть **tan** для автоматичної побудови дотичної до кола або дуги;

NEArest – введіть **nea** для об'єктної прив'язки до найближчої точки зазначеного графічного примітива;

APParent – введіть **app** для об'єктної прив'язки в точці передбачуваного візуального перетинання двох об'єктів на екрані, які реально (фізично) не перетинаються;

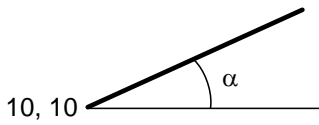
PARallel – введіть **par** для об'єктної прив'язки до паралельно зміщеного об'єкта;

EXTension – введіть **ext** для об'єктної прив'язки до продовжених контурів об'єктів.

У командний рядок досить уводити перші три букви як аббревіатуру параметра.

На рис. 5.13 - 5.16 показані приклади об'єктної прив'язки.

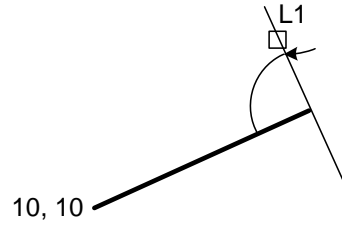
Лінія під кутом до осі X



Command line
Specify first point: 10,10
Specify next point or [Undo]: @ 80<30

а)

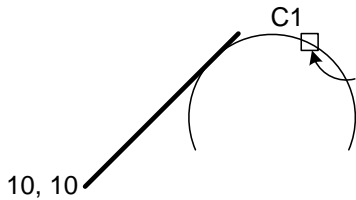
Перпендикуляр до прямої



Command line
Specify first point: 10,10
Specify next point or [Undo]: per
- вказати курсором на лінію L1

г)

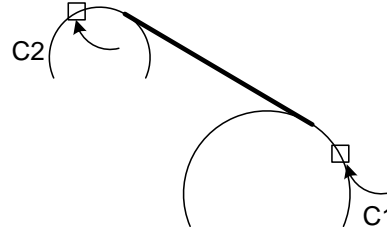
Лінія, дотична до дуги



Command line
Specify first point: 10,10
Specify next point or [Undo]: tan
- натиснути <Enter> і вказати курсором
окружність C1

б)

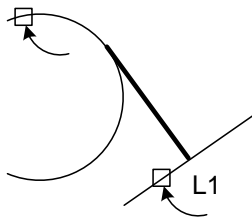
Лінія, дотична до двох дуг



Command line
Specify first point: tan
- натиснути <Enter> і вказати курсором
окружність C1
Specify next point or [Undo]: tan
- натиснути <Enter> і вказати курсором
окружність C2

д)

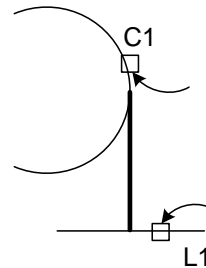
Перпендикуляр із прямої з дотичною дуги



Command line
Specify first point: tan
- натиснути <Enter> і вказати курсором
окружність
Specify next point or [Undo]: per
- натиснути <Enter> і вказати курсором
лінію L1

в)

Пряма із квадрантної точки дуги

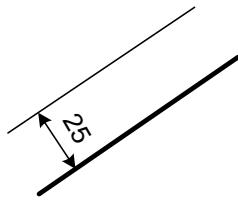


Command line
Specify first point: qua
- натиснути <Enter> і вказати курсором
окружність C1
Specify next point or [Undo]: per
- натиснути <Enter> і вказати курсором
лінію L1

ж)

Рис. 5.13

Лінія, паралельна заданій лінії



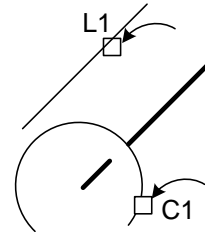
Command offset

Specify offset distance or [Through]: 25

Specify object to offset or <exit>: - вказати курсором вихідний об'єкт

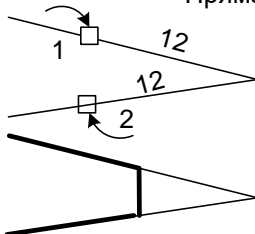
- a) Specify point on side to offset: - вказати курсором зверху

Лінія через центр окружності, паралельна заданій лінії



г)

Пряма фаска



Command chamfer

Select first line or [Polyline/Distance ... D

Specify first chamfer distance: 12

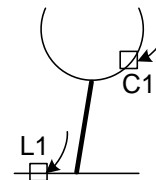
Specify second chamfer distance: 12

Select first line: - вказати курсором

Select second line: - вказати курсором

б)

Пряма через середні точки об'єкта



Command line

Specify first point: MID

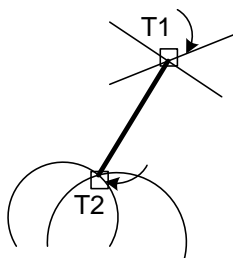
- натиснути <Enter> і вказати курсором на дугу C1

Specify next point or [Undo]: MID

- натиснути <Enter> і вказати курсором на лінію L1

д)

Пряма через точки перетину



Command line

Specify first point: int

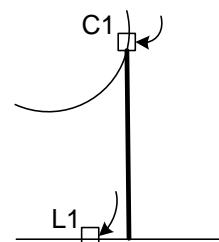
- натиснути <Enter> і вказати курсором на перетинання прямих T1

Specify next point or [Undo]: int

- натиснути <Enter> і вказати курсором на перетин дуг T2

в)

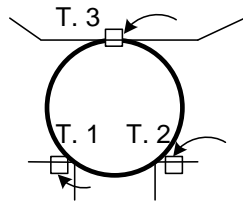
Перпендикуляр до прямої до перетину з дугою



ж)

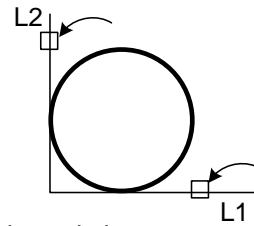
Рис. 5.14

Окружність через три точки



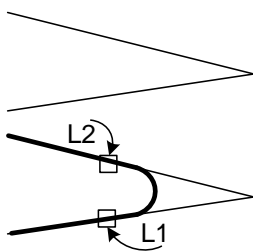
- Command circle
 3P/2P/Ttr (tan tan radius) 3P
 Specify first point on circle: END
 - вказати курсором Т. 1
 Specify second point on circle: END
 - вказати курсором Т. 2
 Specify third point on circle: MID
 - вказати курсором Т. 3

Окружність, дотична до двох прямих

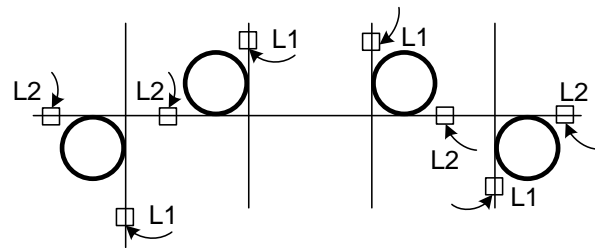


- Command circle
 3P/2P/Ttr (tan tan radius) Ttr
 Specify point on object for first tangent ... : -
 вказати курсором на лінію L1
 Specify point on object for second tangent ... :
 - вказати курсором на лінію L2
 Specify radius of circle <0.5>: 28
 - ввести радіус

Радіальне округлення ліній



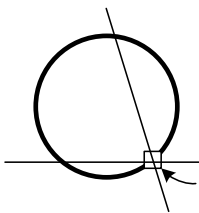
- Command fillet
 Select first object or [Polyline/Radius ... R
 Specify fillet radius: 15
 Select first object: - вказати курсором
 б) Select second object: - вказати курсором



Часткові випадки визначення місця вказівки

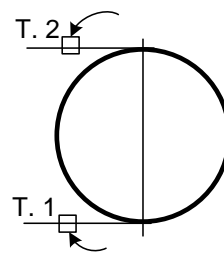
д)

Окружність через точку перетину



- Command circle
 Specify center point for circle ... : 50,75
 Specify radius of circle ... : int
 - вказати курсором на точку перетину
 двох ліній
 в)

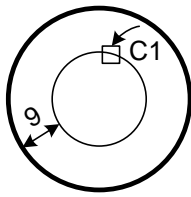
Окружність через дві точки



- Command circle
 Specify center point ... : 2P
 Specify first end point ... END
 - вказати курсором на точку Т.1
 Specify second end point ... END
 ж) - вкажіть курсором на точку Т.2

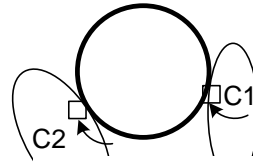
Рис. 5.15

Окружність через три точки



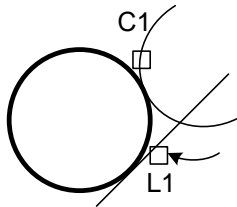
Command offset
 Specify offset distance: 9
 Select object to offset:
 - вказати курсором окружність C1
 Specify point on side to offset:
 - вказати курсором в сторону виконання нової окружності

Окружність, дотична до двох еліпсів

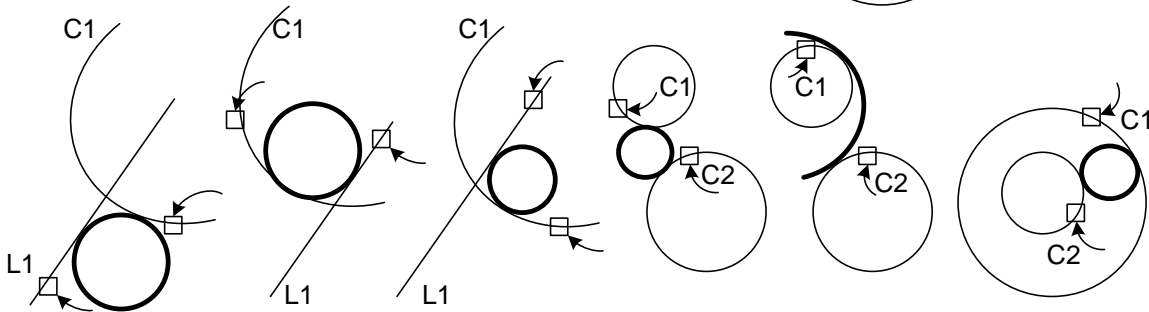
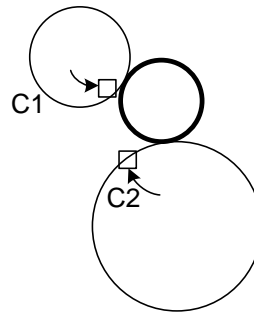


Command circle
 3P/2P/Ttr (tan tan radius) Ttr
 Specify point on object for first tangent ... : -
 вказати курсором на еліпс C1
 Specify point on object for second tangent ... :
 - вказати курсором на еліпс C2

Окружність, дотична до прямої і дуги



Окружність, дотична до двох дуг



Command circle
 3P/2P/Ttr (tan tan radius) Ttr
 Specify point on object for first tangent ... : -
 вказати курсором на лінію L1
 Specify point on object for second tangent ... :
 - вказати курсором на дугу C1
 Specify radius of circle: 25

Command circle
 3P/2P/Ttr (tan tan radius) Ttr
 Specify point on object for first tangent ... : -
 вказати курсором на окружність C1
 Specify point on object for second tangent ... :
 - вказати курсором на окружність C2

Рис. 5.16

5.7. Метричні задачі

Розв'язання метричних задач дає можливість визначити:

- 1) координати окремих точок;
- 2) параметри окремих примітивів;
- 3) відстані між двома точками;
- 4) периметри.

Наведемо приклади розв'язання метричних завдань.

1. Визначити координати точки відрізка, зображеного на рис. 5.5.

Для одержання довідкової інформації використовується команда ID. Вона показує в командному рядку тривимірні координати обраної точки.

Введемо в командний рядок:

```
Command: id  
Specify point: end (Натисніть клавішу <Enter>, вкажіть задану точку )  
of X=40.0000   Y=40.0000   Z=0.0000
```

Результат з'являється в командному рядку.

2. Визначення координат точки перетину двох ліній.

```
Command: id  
Specify point: app (Натисніть клавішу <Enter>, укажіть точку перетину.)  
of X=168.6672   Y=88.3356   Z=0.0000
```

Результат можливо обробляти (копіювати, змінювати параметри й ін.) у текстовому вікні AutoCAD (рис. 5.16).

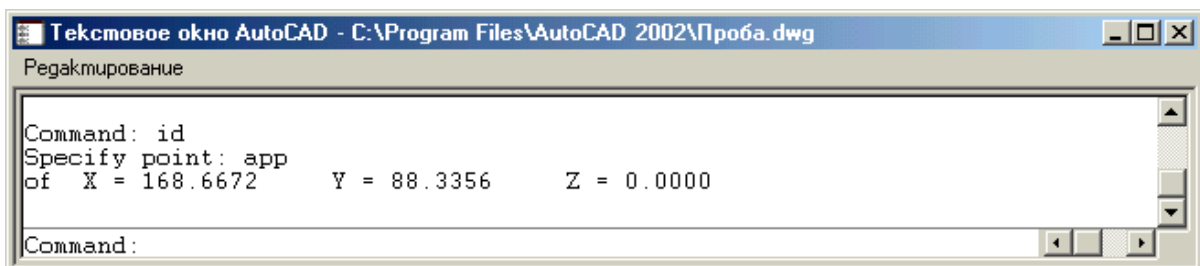


Рис. 5.17

3. Повний список параметрів обраного об'єкта видається командою:

Command: list

Select objects:

LINE Layer: "0" (тип об'єкта і ім'я шару)
Space: Model space
Handle = 3E0
from point, X= 10.0000 Y= 20.0000 Z= 0.0000 (початкова точка)
to point, X= 40.0000 Y= 40.0000 Z= 0.0000 (кінцева точка)
Length = 36.0555, Angle in XY Plane = 34 (довжина й кут)
Delta X = 30.0000, Delta Y = 20.0000, Delta Z = 0.0000
(збільшення)

4. Визначення відстані і кутів у тривимірному просторі між двома точками виконується командою Dist:

Command: dist

Specify first point:

Specify second point:

Distance=76.2499, Angle in XY Plane=23, Angle from XY Plane=0

Delta X=70.0367, Delta Y=30.1479, Delta Z=0.0000

5. Обчислення площі й периметра замкнутих плоских фігур виконується за командою Area.

Command: area

Specify first corner point or [Object/Add/Subtract]: O

Specify object:

Area = 1559.3356, Perimeter = 162.6223

5.8. Проставляння розмірів

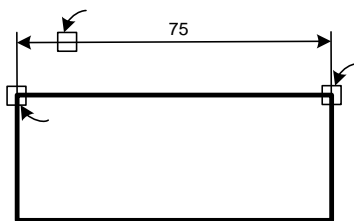
В AutoCAD процес нанесення розмірів автоматизований. AutoCAD автоматично вимірює об'єкт і проставляє на кресленні значення розмірів.

Проставлянням розмірів керує команда **DIM**, що дозволяє перейти в режим керування розмірами.

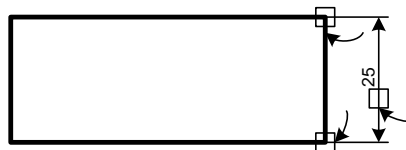
Існує декілька способів виклику команд нанесення розмірів. Команди створення розмірів перебувають у меню **Dimension** (Розмір). Крім того, за допомогою команди **Toolbars** (Панелі інструментів) можна створити панель **Dimension** (Розмір) і використати її для проставляння розмірів (рис. 5.18). Нарешті, команди нанесення розмірів можна безпосередньо вводити в командному рядку.

Рис. 5.18

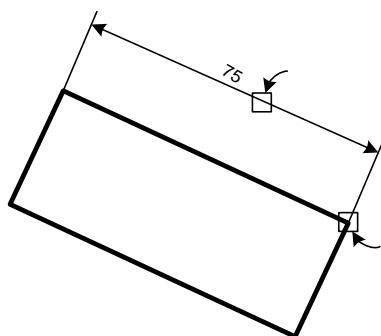
Приклади проставляння розмірів показані на рис. 5.19 - 5.20.



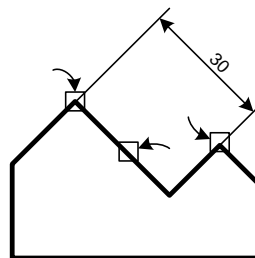
Command: dim H
HOR - горизонтальне розташування
розмірної лінії
Вказати курсором на першу точку розміру
Вказати курсором на другу точку розміру
Вказати розташування розмірної лінії
Enter dimension text: - ввести значення
розміру



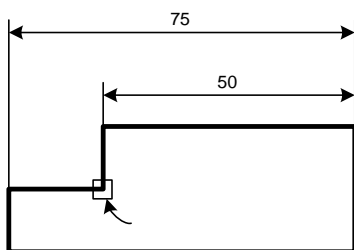
Command: dim
VER - вертикальне розташування розмірної
лінії
Вказати курсором на першу точку розміру
Вказати курсором на другу точку розміру
Вказати розташування розмірної лінії
Enter dimension text: - ввести значення
розміру



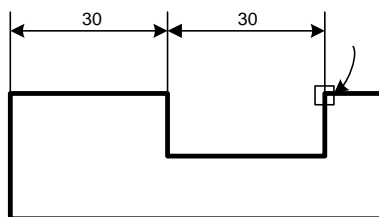
Command: dal - розмірна лінія проходить
через початкову і кінцеву точки
Вказати курсором на першу точку розміру
Вказати курсором на другу точку розміру
Вказати розташування розмірної лінії
Enter dimension text: - ввести значення
розміру



Command: dli - розмірна лінія проходить
під заданим кутом
Вказати курсором на першу точку розміру
Вказати курсором на другу точку розміру
Mtext/Text/Angle/Horizontal/Vertical/Rotate
- визначити положення лінії розміру

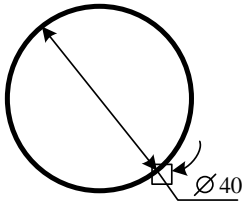


Command: dba - проставлення
послідуючого розміру від бази, заданої
попереднім розміром. Команда дається у
випадку проставлення паралельного
розмірного ланцюга. Кожен послідуєчий
розмір потребує задання другої розмірної
лінії
Specify second extension ... - друга точка

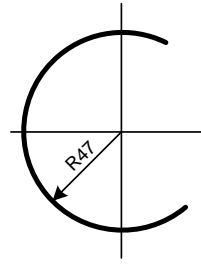


Command: dco - починає новий розмір з
другої виносної лінії попереднього розміру
Specify second extension ... - друга точка
розміру

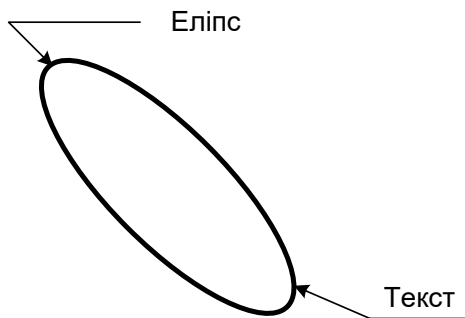
Рис. 5.19



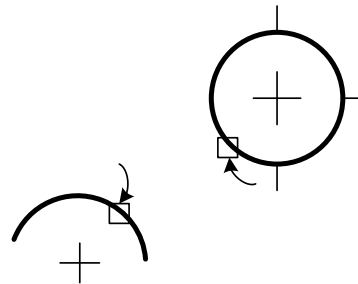
Command: dimdia
 Select arc or circle - вказати маркером
 Знак Ø встановлюється автоматично, але при зміні числа з клавіатури вводять текст у вигляді %%d 40.
 Після завершення команди буде сформовано надпис Ø40



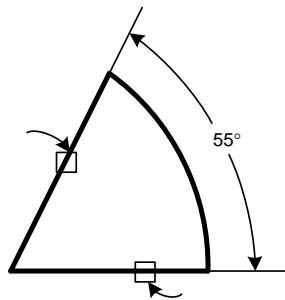
Command: dimrad
 Select arc or circle - вказати маркером
 Specify dimension line location - визначити розташування розміру



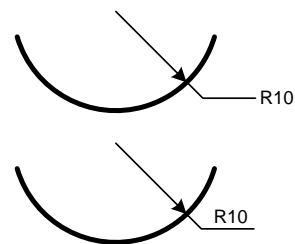
Command: leader
 - виконання надписів з виносними полочками
 Specify leader start point - вибрати вихідну точку виноски
 Specify next point or [Annotation/Format/Undo] - виберіть слідувачу точку або натисніть <Enter> для введення тексту



Command: dimcenter
 - виконання осьових ліній для окружностей або дуг
 Select arc or circle - вибрати дугу або коло.
 Розмір засічок осьових ліній встановлюється системною змінною DIM: DimCen



Command: dinangular
 - простановка кутового розміру
 Select arc, circle, line or <specify vertex>
 - виберіть об'єкт
 Specify second angle endpoint ... - друга точка



Command: explode
 - розчленовує блок, полілінію, зв'язані розміри, штриховку та ін. на складові елементи
 Select objects ... - вибрати один або декілька об'єктів
 Select objects - натисніть <Enter> для завершення виділення

Рис. 5.20

5.9. Приклад розроблення плоских моделей

Розроблення креслення починається з перевірки правильності установок й основних режимів системи. Основними установками є:

- 1) шари (**Layer**)
- 2) одиниці виміру (**Units**);
- 3) текстові стилі (**Style**);
- 4) формати (**Limits**);
- 5) розмір твердого кроку курсора (**Snap**);
- 6) розмір кроку екранної сітки (**Grid**).

1. Креслення містить багато різних об'єктів. Тому доцільно для кожного об'єкта креслення задавати шар, на якому він розташований. Шари подібні до кальок, накладених одна на одну. Кожен шар несе інформацію про певні елементи креслення. Крім того, у процесі розроблення креслення використовуються допоміжні побудови, які на остаточному кресленні не потрібні.

Створимо шари: побудов (Post); основної моделі (Osn); осьових ліній (Osi); розмірних ліній (Razm) (рис. 5.21).

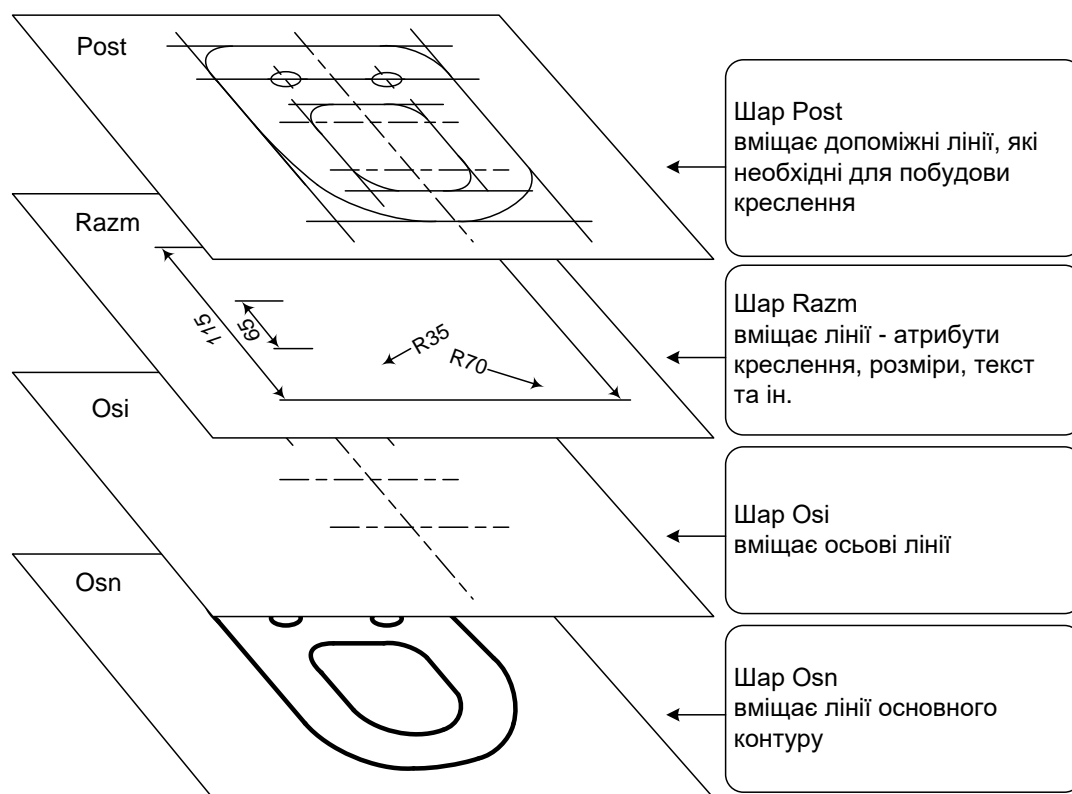


Рис. 5.21

Якщо натиснути на кнопку **Layer** (Шар) на панелі шарів або вибрати команду з меню **Format**, відкриється діалогове вікно **Layer Properties Manager** (Менеджер властивостей шару) (рис. 5.22).

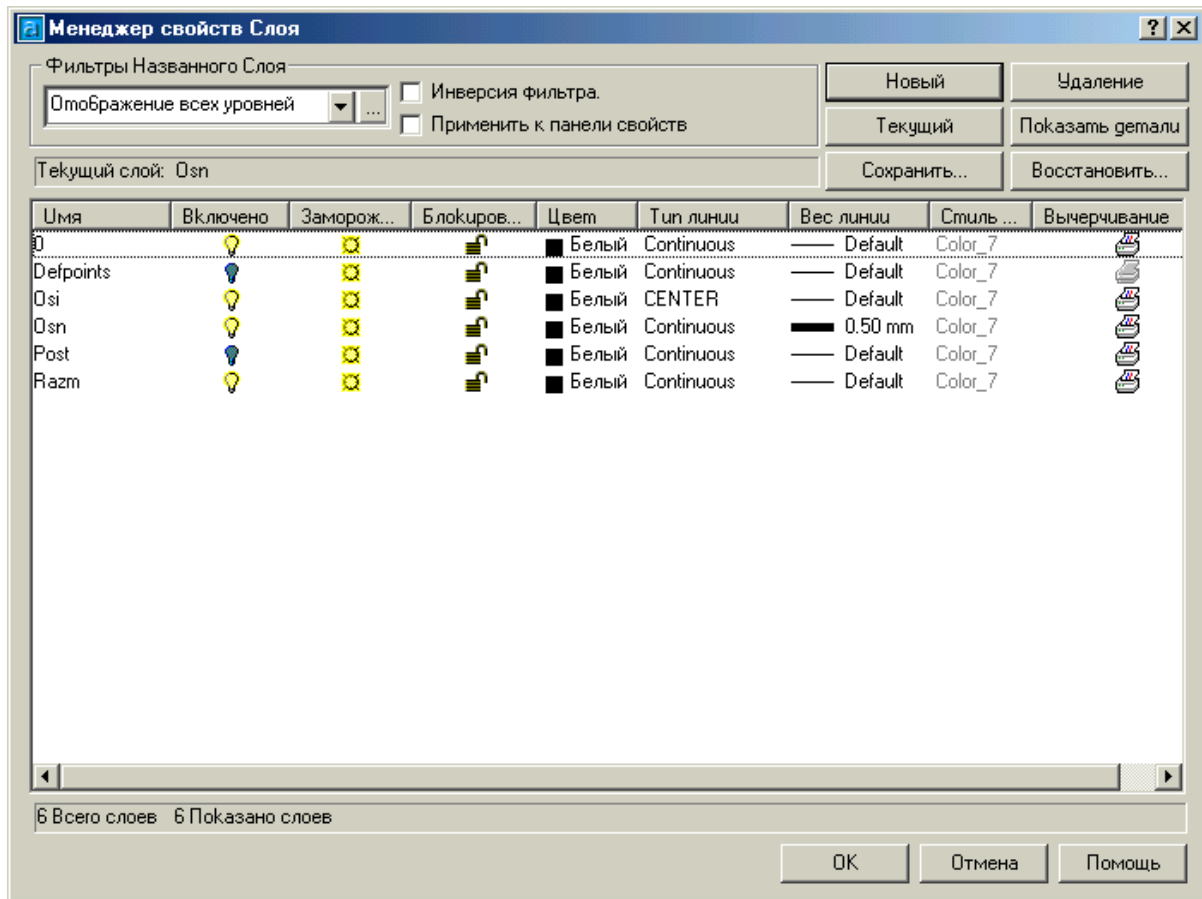


Рис. 5.22

Менеджер властивостей дозволяє створити новий, видалити непотрібний шар, задати стан шару (поточний, активний, вимикання, «заморожений»).

2. Для вибору одиниць виміру введіть у командному рядку команду `units` і натисніть на клавішу `<Enter>`. Відкриється діалогове вікно (рис. 5.23), у якому необхідно встановити одиниці виміру.

3. Розміри математичного простору для створення креслення визначаються командою `Limits`:

Command: limits

Specify lower left corner or [ON/OFF] <0.0000, 0.0000>: 0,0 (визначити лівий нижній кут межі креслення)

Specify upper right corner <12.0000, 9.0000>: 210, 298 (визначити правий верхній кут межі креслення)

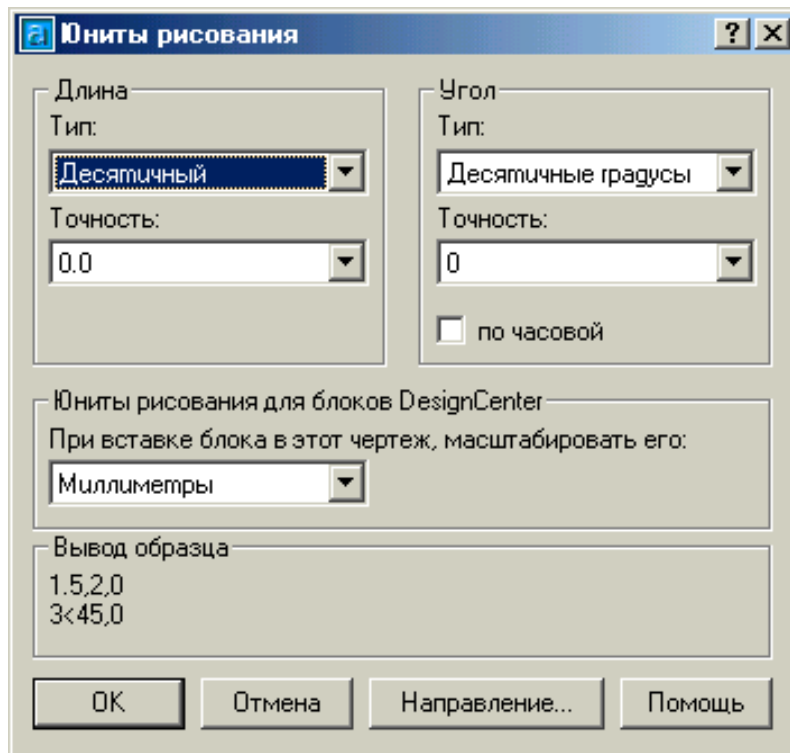


Рис. 5.23

4. Для розміщення в кресленні текстових рядків використовується команда **Text**:

Command: text

Current text style: "Standard" Text height: 0.2000

Specify start point of text or [Justify/Style]: (Вибрати точку або ввести параметр)

Specify height <0.2000> (Ввести розмір тексту)

Specify rotation angle of text <0>: (Ввести значення)

Enter text: (Ввести текст і натиснути <Enter>)

Enter text: (Натиснути <Enter> для завершення команди)

Створення нового стилю або перегляд списку сформованих стилів виконується командою **Style**. Можна також вибрати команду **Text Style** (Стиль тексту) у меню **Format**, і на екрані з'явиться вікно, що дозволяє створювати й редагувати стилі тексту (рис. 5.24).

5. При формуванні моделі за допомогою курсора можна забезпечити точне задання точки за умовною сіткою. Це забезпечується командою **Snap**. Вона встановлює режим координатної прив'язки курсора, при якому мінімальний зсув курсора можливий тільки на величину, визначену кроком прив'язки. Величину кроку прив'язки можна регулювати:

Command: snap

Specify snap spacing or [ON/OFF/Aspect/Rotate/Style/Type] <0.5> (вказіть значення кроку прив'язки або параметр)

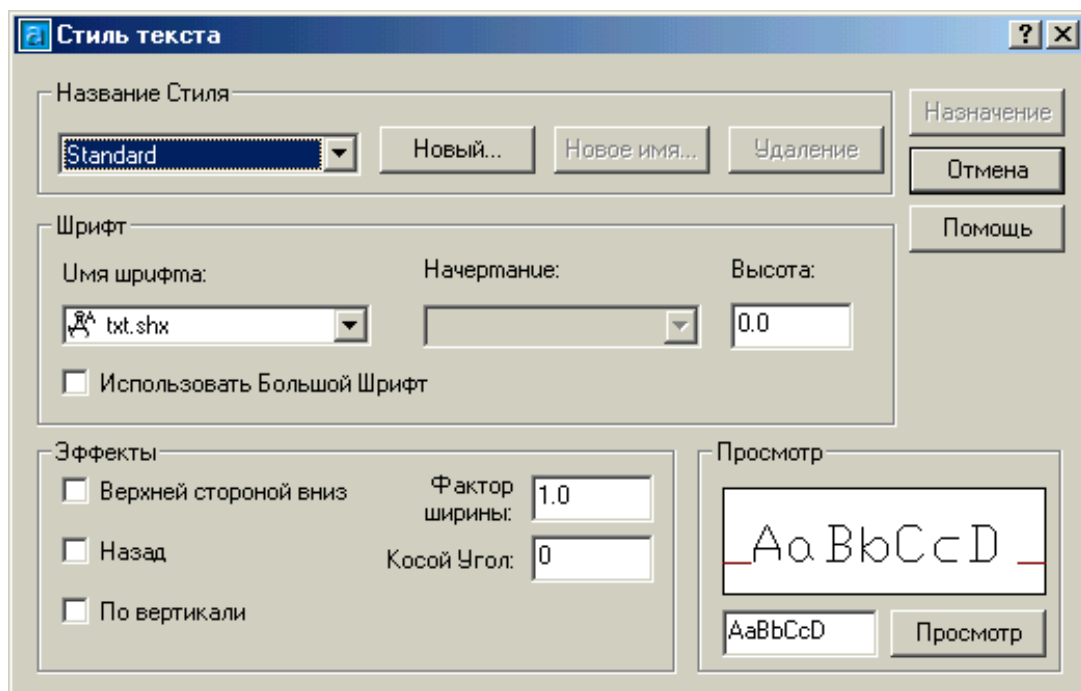


Рис. 5.24

6. При формуванні моделі можна створити на екрані зображення ортогональної точкової сітки за заданими межами креслення. Ця сітка допомагає орієнтуватися в математичному просторі. Вона не є частиною креслення і не виводиться на плотер або принтер. За командою можна регулювати цю сітку.

Command: grid

Specify grid spacing (X) or [ON/OFF/Snap/Aspect] <0.5000>: (введіть крок сітки або параметр)

Основні установки виконані й можна приступати до розроблення креслення:

1) почнемо із креслення рамки й основного напису креслення.

Установимо поточний шар **Osn**. У рядку стану натиснемо на кнопку **Ortho**. У цьому випадку при кресленні й редагуванні зсув можна задати тільки в горизонтальному або вертикальному напрямку. Для креслення рамки й основного напису досить використати команду **Line**. Введення тексту виконаємо за допомогою команди **Text**;

2) застосуємо команду **Ucs** і виберемо початок локальної системи координат у точці (110, 140);

3) встановимо поточний шар для виконання осей **Osi**. Проведемо осьові лінії;

4) встановимо поточний шар для побудов **Post**. Відключимо режим **Ortho**. Виконаємо допоміжні побудови, використовуючи команди **Line**, **Circle**, **Arc**. Рекомендується починати побудову із прямих ліній. Можна використати симетричне відображення вертикальних ліній. При кресленні дуг краще використати об'єкту прив'язку;

5) встановимо поточний шар для обведеного контуру **Osn**. Для зручного захоплення характеристичних точок необхідно включити режим поточної об'єктної прив'язки кінцевих (**End**), квадратних (**Qua**) точок і точок перетинання (**Int**). У цьому випадку можна вказувати на точки приблизно;

6) обведемо контур, використовуючи команду **Pline**. Товщину ліній обведення задамо рівною 0.5 мм;

7) встановимо поточний шар для проставляння розмірів й «заморозимо» шар, у якому виконувалися побудови;

8) проставимо розміри, використовуючи приклади на рис. 5.18 - 5.20.

Результати роботи показані на рис. 5.25.

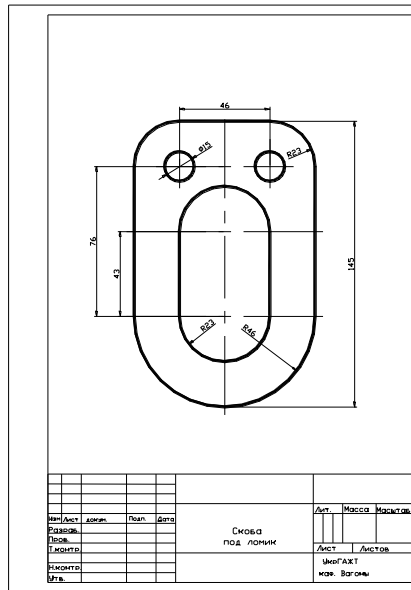


Рис. 5.25

6. РОЗРАХУНОК НА МІЦНІСТЬ В MSC/NASTRAN

6.1. Основні етапи підготовки

Проектування конструкції незмінно пов'язане з аналізом її міцності. Сучасним методом розрахунку конструкцій є метод кінцевих елементів (МКЕ). Він дозволяє розраховувати різні конструкції (балки, пластини, оболонки й т.п.) єдиним способом. Відомі різні пакети прикладних програм кінцево-елементного аналізу: ANSYS й DesignSpace фірми ANSYS Corporation, Cosmos фірми Structural Research & Analysis Corporation, DesignWorks фірми CADSI й ін.

Одним із широко відомих розробників програм кінцево-елементного аналізу є компанія MacNeal-Schwendler Corporation - MSC. Цією компанією розроблена система NASTRAN (NAsa STRuctural ANalysis - аналіз конструкцій національного комітету США з аеронавтики й космічних досліджень). Надалі буде використований один із програмних продуктів цієї фірми - MSC.visualNasran for Windows [18].

Програма кінцево-елементного розрахунку конструкції включає такі основні етапи:

- розроблення геометричної моделі конструкції;
- задання характеристик матеріалів елементів конструкції;
- вибір типів кінцевих елементів;
- розбивання конструкції на кінцеві елементи;
- задання граничних умов;
- розрахунок конструкції – звичайно-елементний аналіз;
- аналіз результатів розрахунку;
- виведення результатів.

Геометрична модель конструкції може бути імпортована з іншої програми (наприклад, AutoCAD).

6.2. Інтерфейс програми

Після завантаження на екрані з'являється вікно програми, яке прийнято називати інтерфейсом користувача (рис. 6.1). Інтерфейс виступає в ролі посередника між людиною й комп'ютером. Він надає користувачеві все необхідне для роботи: діалогові вікна, палітри інструментів і т.д.

В операційну систему закладений наскрізний принцип організації всіх додатків, що працюють під її керуванням - віконний інтерфейс. Всі загальні елементи вікон виконані в єдиному стандарті.

Інтерфейс користувача включає заголовок програми, головне меню, панель інструментів, панель команд, рядок підказки й статусний рядок. У центрі головного вікна розташовується графічне вікно, у якому відображається розроблювана модель і результати розрахунку.

Головне командне меню розташовується під смугою заголовка і включає такі групи команд: **File** (Файл), **Tools** (Інструменти), **Geometry** (Геометрія), **Model** (Модель), **Mesh** (Сітка), **Modify** (Змінити), **List** (Список), **Delete** (Видалити), **Group** (Групи), **View** (Вид), **Help** (Довідка).

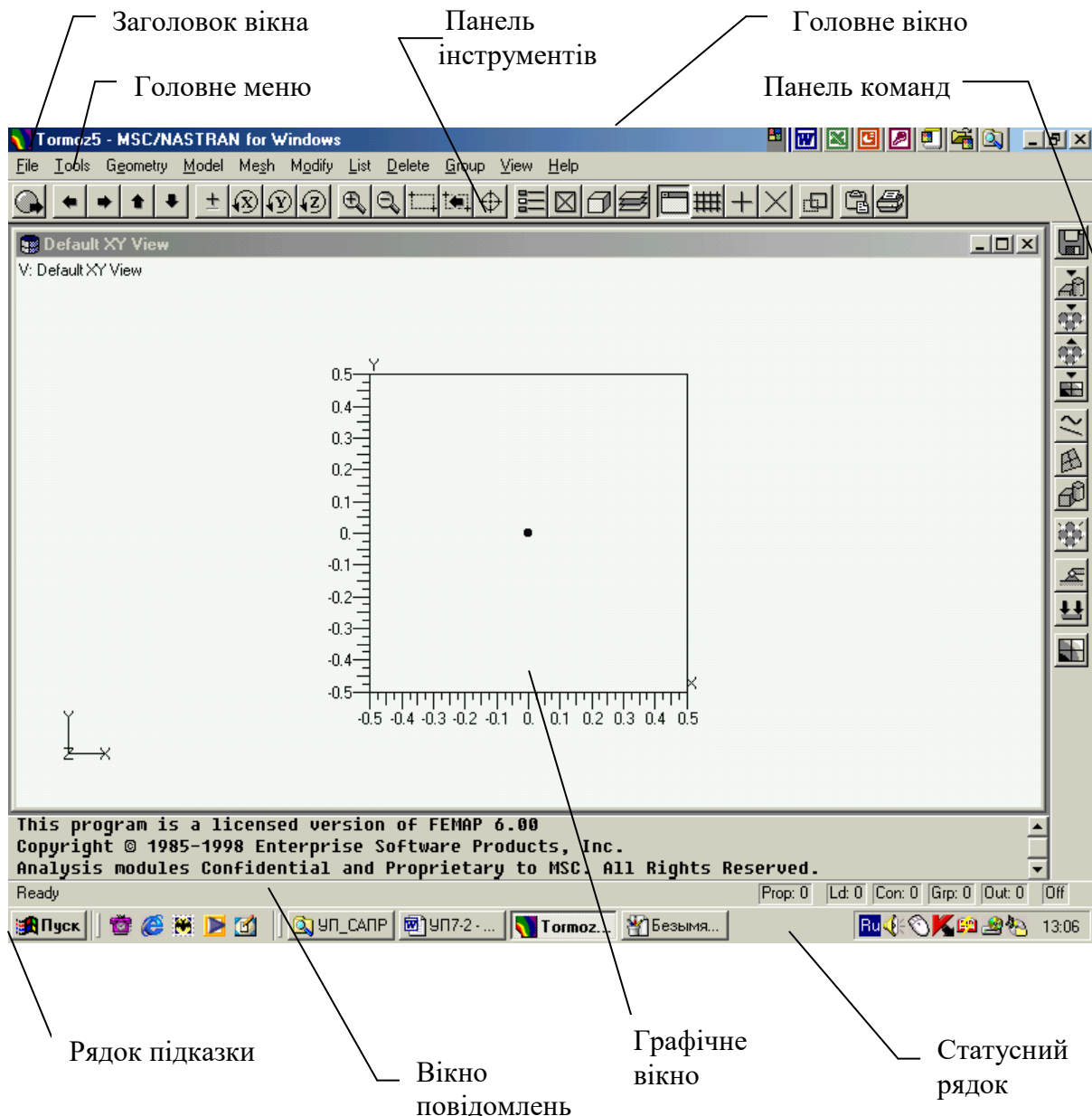


Рис. 6.1

На панелі інструментів **View** містяться кнопки, за допомогою яких відображається модель, яка розробляється. Призначення кнопок показано на рис. 6.2.

Панель команд забезпечує доступ до основних команд меню (рис. 6.1). При натисканні на будь-яку кнопку панелі команд вона модифікується. Наприклад, після натискання на кнопку **PostProcess** (Результати розрахунку) панель набуває вигляду, показаного праворуч на рис. 6.3.

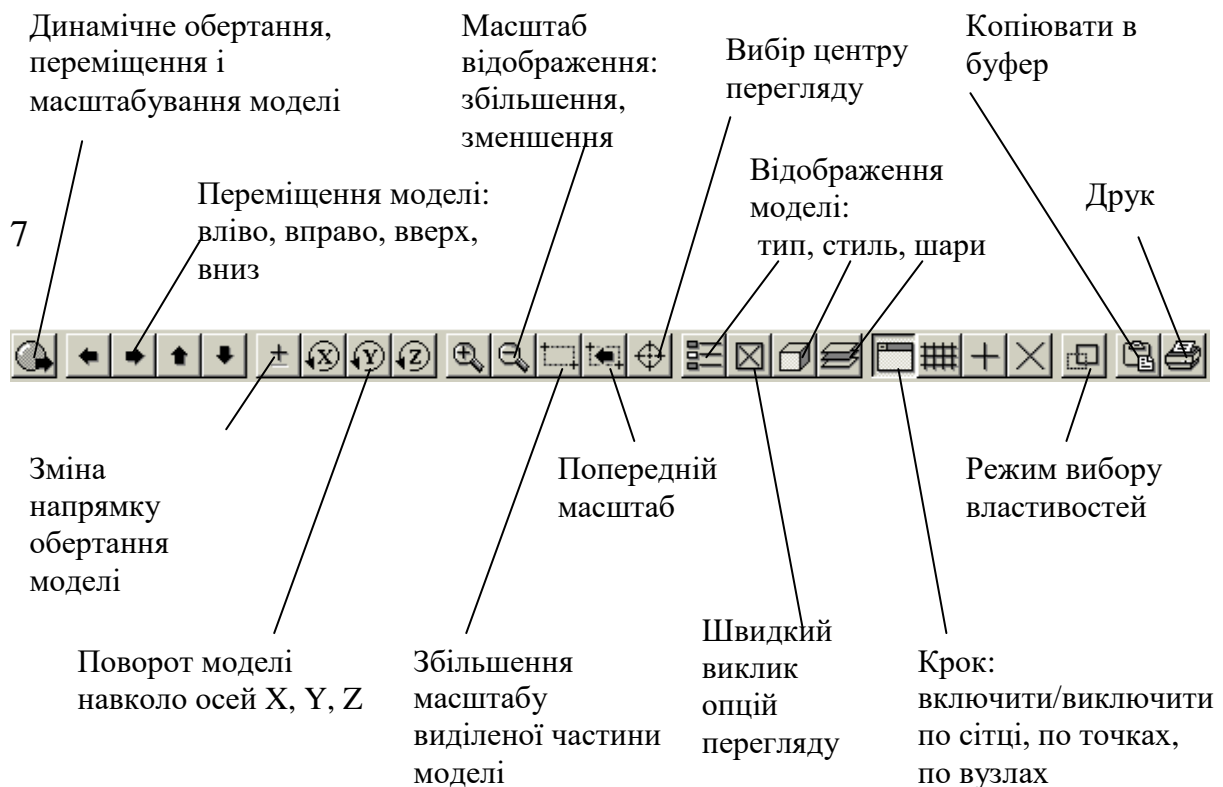


Рис. 6.2

Модель, яка розробляється, і результати розрахунку відображаються в графічному вікні. У центрі вікна є точка початку координат.

У вікні повідомлень відображається інформація про властивості моделі, результати виконання команд і т.п.

Розміри графічного вікна й вікна повідомлень можна змінювати.

Подвійне натискання „мишею” у вікні повідомлень розгортає вікно до повних розмірів і дає можливість переглянути всі повідомлення про раніше виконані дії. Повторним подвійним натисканням вікно згортається до попередніх розмірів.

У статусному рядку вказуються поточні значення ідентифікаторів властивостей кінцевих елементів (**Prop**), навантажень (**Ld**), граничних умов (**Con**), груп (**Grp**) і результатів (**Out**).

Рис. 6.3.

Можна використати інші панелі інструментів. Для цього необхідно скористатися пунктом меню **Tools – Toolbox** і вибрати необхідну панель (рис. 6.4).

Рис. 6.4

6.3. Створення точок, прямих ліній, дуг, окружностей і сплайнів

При розробленні геометричних моделей використовуються такі геометричні об'єкти: точки, прямі лінії, криві (дуги, окружності, сплайни), поверхні, а також обсяги й тіла.

Після запуску MSC/N4W для створення нової моделі в діалоговому вікні, що відкрилося, варто натиснути кнопку **New Model** (Нова модель). У головному меню виберемо пункт **Geometry – Point** (Геометрія – Точка). На екрані з'явиться вікно, у якому можна ввести характеристики точки (рис. 6.5). Таке вікно типово для MSC/N4W.

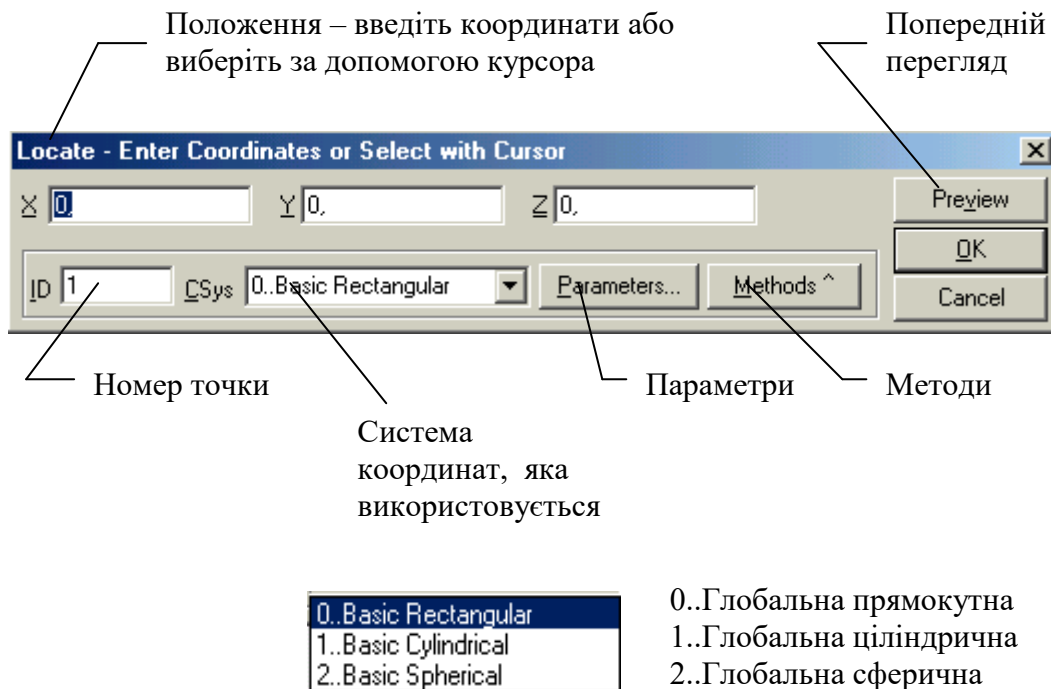


Рис. 6.5

В цьому вікні можна задати:

- **X, Y, Z** – координати точки;
- **ID** – номер (ідентифікатор) точки;
- **Csys** – система координат, яка використовується; за замовчуванням прийнята прямокутна система координат;
- **Parameters** (Параметри) – кнопка, після натискання на яку з'являється діалогове вікно **Geometry Parameters** (Геометричні параметри);

- **Methods** – (Методи). Натискання цієї кнопки викликає меню, у якому можна задати положення точки. За замовчуванням встановлений спосіб **Locate** (Положення), що задає точку трьома координатами.

Введемо координати точки: X: -0.35, Y: 0.2, Z: 0 і натиснемо кнопку ОК. Точка з'явиться в графічному вікні і діалогове вікно буде готове до введення координат наступної точки.

Введемо координати точки 2: X: 0.05, Y: 0.5, Z: 0 і натиснемо кнопку ОК.

Таким же чином введемо координати точок 3, 4, 5. Натиснемо кнопку Cancel і припинимо уведення точок.

Тепер перейдемо до побудови прямих ліній.

Кожен геометричний об'єкт (точка, пряма, окружність і т.п.) можна побудувати багатьма способами.

Виберемо пункт меню **Geometry – Curve - Line** (Геометрія – Пряма). У меню, що випадає, можна побачити різні методи побудови прямих ліній (рис. 6.6). У нашому прикладі точки вже побудовані. Тому має сенс побудувати прямі лінії по точках.

Пряма лінія

Project Points...	За проєкціями точок (на робочу площину)
Horizontal...	Горизонтальна
Vertical...	Вертикальна
Perpendicular...	Перпендикуляр
Parallel...	Паралель
Midline...	Середня лінія
At Angle...	За кутом
Angle to Curve...	Під кутом до лінії
Point and Tangent...	За умовами торкання і кінцевою точкою
Tangent...	Дотична до двох кривих
Rectangle...	Прямокутник
Continuous...	Замкнута (багатокутник)
Points...	За наявними точками
Coordinates...	За координатами
Offset...	За зміщенням від заданої кривої
Vectored...	Вздовж вектора

Рис 6.6

Виберемо пункт меню **Geometry – Curve – Line - Points** (Геометрія – Пряма – По точках). При цьому з'явиться діалогове вікно **Create Line from Points** (Створити пряму по точках), у якому варто ввести номери (ідентифікатори) початкової й кінцевої точок і натиснути ОК.

Подібним чином можна побудувати інші лінії. Для завершення побудови ліній необхідно натиснути кнопку Cancel.

Деякі властивості ліній, які відображені, можна задати за допомогою пункту меню **View – Options** (Опції виду). Вибір цього пункту викликає поява вікна **View Options** (рис. 6.7). Його можна викликати також натисканням клавіші F6.

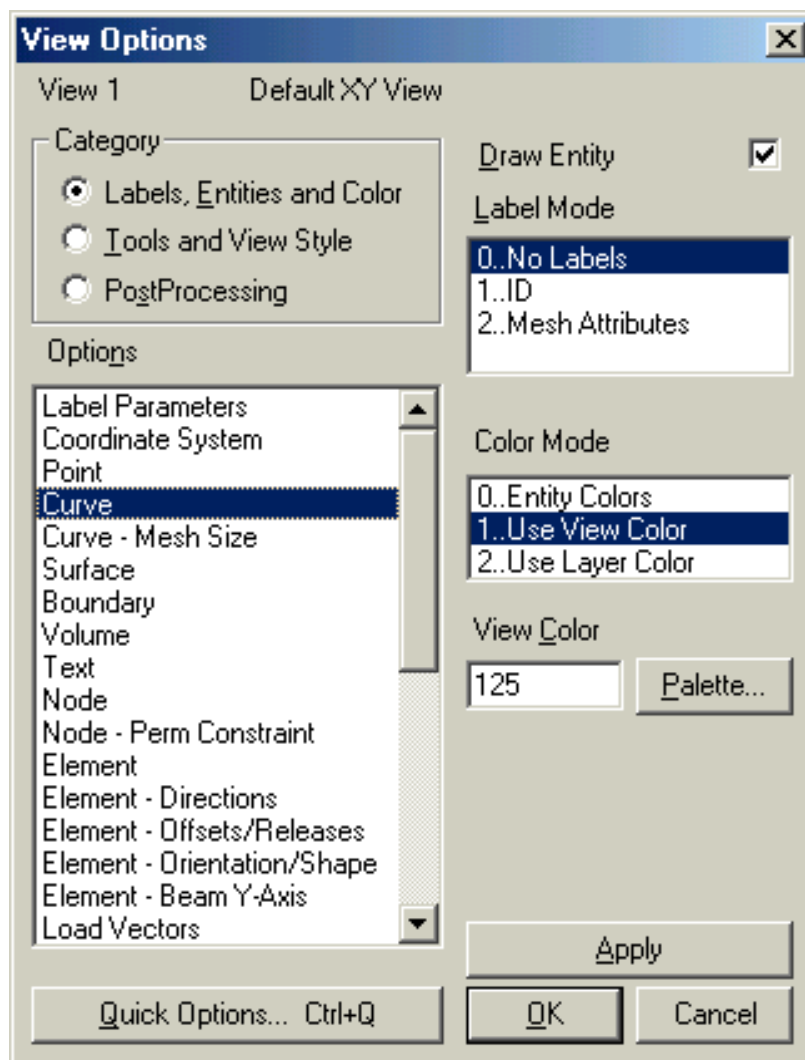


Рис. 6.7

У лівій верхній частині вікна показані три категорії (**Category**) параметрів:

- **Labels, Entities and Color** (Позначення, об'єкти й кольори);
- **Tools and View Style** (Інструменти й стиль виду);
- **PostProcessing** (Результати розрахунку).

Категорія **Labels, Entities and Color** встановлює параметри, які відносяться до виду системи координат, точок, ліній, поверхонь й ін.

Категорія **Tools and View Style** має у своєму розпорядженні набір інструментів і встановлює стиль відображення. Наприклад, параметр **Origin** визначає початок координат (відобразити/сховати, кольори); параметр **View Legend** встановлює відображення легенди (вид, навантаження, граничні умови); параметр **Symbols** встановлює розмір і кольори символів вузлів, точок й ін.

Категорія встановлює параметри, які відносяться до стилю відображення результатів розрахунку. Наприклад, параметр **Deformed Model** визначає стиль відображення деформованого стану моделі (кольори сітки елементів); параметр **Post Titles** встановлює найменування вихідних параметрів.

У правій частині вікна **View Options** містяться елементи керування опціями відображення: **Draw Entity** (Відобразити об'єкт), **Color Mode** (Колірний режим), **View Color** (Кольори виду) і кнопка **Palette** (Палітра).

Щоб відобразити на екрані лінії чорних кольорів, виберемо категорію **Labels, Entities and Color** і у розділі **Options** - параметр **Curve** (Крива); у списку **Label Mode** – 0..No Labels (не відображати). За допомогою кнопки **Palette** можна вибрати кольори лінії і натиснути на кнопку **Apply** (Застосувати). Після цього лінії на екрані набудуть чорних кольорів.

Для побудови окружностей використовується пункт меню **Geometry – Curve - Circle** (Окружність). У підменю цього пункту можна побачити способи побудови окружностей (рис. 6.8).

Окружність

R <u>a</u> d <u>i</u> u <u>s</u> ...	Радіус (задається двома точками)
D <u>i</u> a <u>m</u> e <u>t</u> e <u>r</u> ...	Діаметр (задається двома точками)
C <u>e</u> n <u>t</u> e <u>r</u> ...	Центр (по центру і радіусу)
T <u>w</u> o P <u>o</u> i <u>n</u> t <u>s</u> ...	По двох точках (і радіусу)
P <u>o</u> i <u>n</u> t-T <u>a</u> n <u>g</u> e <u>n</u> t...	По двох точках і дотику (з заданою кривою)
T <u>a</u> n <u>g</u> e <u>n</u> t t <u>o</u> C <u>u</u> r <u>v</u> e <u>s</u> ...	По дотику з двома кривими
C <u>o</u> n <u>c</u> e <u>n</u> t <u>r</u> i <u>c</u> ...	Концентрична (до даної окружності)
P <u>o</u> i <u>n</u> t <u>s</u> o <u>n</u> A <u>r</u> c...	По точках (трьох) на дузі
C <u>e</u> n <u>t</u> e <u>r</u> a <u>n</u> d P <u>o</u> i <u>n</u> t <u>s</u> ...	

Рис. 6.8

Побудуємо окружність радіуса 0.15 із центром на початку координат.

Виберемо метод побудови окружності по центру і радіусу. Викличемо команду **Geometry – Curve – Circle- Center** (Геометрія – Окружність - Центр). У діалоговому вікні, що з'явилося, введемо координати центра окружності й натиснемо ОК. При цьому з'явиться вікно, у якому потрібно ввести значення радіуса. Після цього натиснемо ОК і Cancel для завершення побудови окружності. Результати побудови точок, прямих й окружності показані на рис. 6.9.

В MSC/N4V є кілька способів побудови дуг (рис. 6.10). Після виклику команди меню **Geometry – Curve – Arc** можна вибрати один зі способів побудови дуги.

Якщо потрібно побудувати криву складнішої форми, то необхідно звернутися до команди **Spline** (Сплайн).

Сплайн створюють у тих випадках, коли потрібно з'єднати ряд точок не ламаної, а гладкої кривої. У цьому випадку часто використовується інтерполяція кубічними сплайнами, тобто відрізками кубічних парабол. Більш складним типом інтерполяції є так називана інтерполяція B - сплайнами. Сплайни можуть бути лінійними, квадратичними або кубічними.

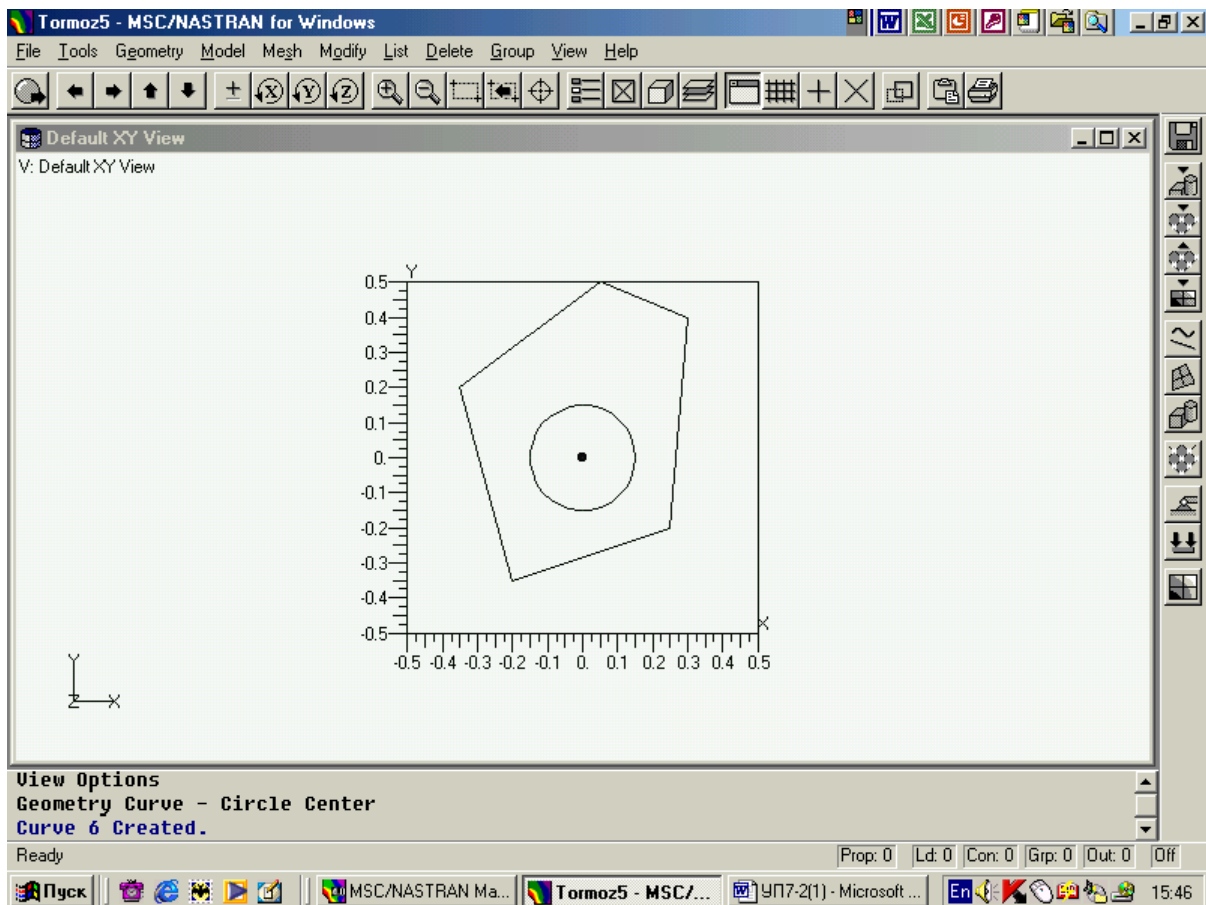


Рис. 6.9

Дуга

- Center-Start-End...
 - Radius-Start-End...
 - Angle-Start-End...
 - Angle-Center-Start...
 - Chord-Center-Start...
-
- Points...
 - Center and Points...
 - Start-End-Direction...

- Центр – початок - кінець
- Радіус – початок – кінець
- Кут – початок – кінець
- Кут – центр – початок**
- Хорда – центр - початок

- По точках
- По центру і точках
- Початок– кінець – напрямок (дотичне)

Рис. 6.10

Використовується декілька способів побудови сплайнів (рис. 6.11).

Створимо сплайн по точках, що лежать у робочій площині, або по їхніх проекціях, якщо точки перебувають поза нею.

Крива - сплайн

Project Control Points...	Проекції контрольних точок
Project Points...	Проекції точок
Ellipse...	Еліпс
Parabola...	Парабола
Hyperbola...	Гіпербола
Control Points...	Контрольні точки
Points...	Точки
Equation...	Рівняння
Tangents...	Дотичні
Blend...	Сполучення (двох кривих)
Midspline...	Середня крива
Offset...	Зміщення
Multiple Curves...	Замкнута крива

Рис. 6.11

Виберемо пункт меню **Geometry – Curve Spline – Project Control Points**. У діалоговому меню, що відкривається, введемо координати точок: X: -0.6, Y: 0.3, Z: 0; X: -0.35, Y: -0.1, Z: 0; X: 0.1, Y: 0.45, Z: 0; X: 0.3, Y: 0.25, Z: 0. Результати введення координат відображаються на екрані.

Натиснемо ОК. На екрані з'являється сплайн (рис. 6.12). Для завершення роботи натиснемо клавішу Cancel.

За допомогою сплайнів можна будувати інші криві: еліпс, параболу, гіперболу.

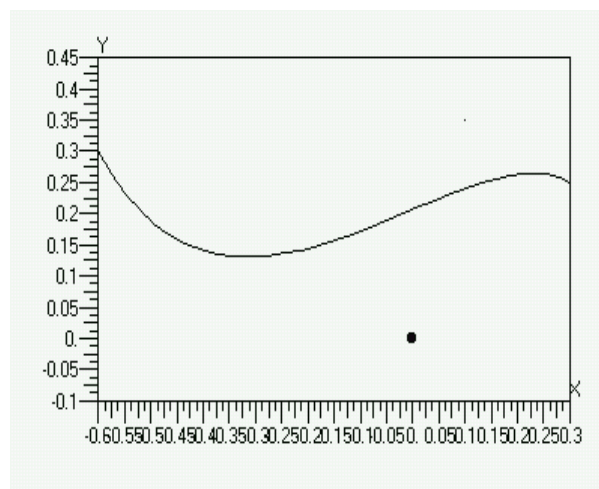


Рис. 6.12

6.4. Створення поверхонь

В MSC/N4V можна створювати поверхні трьох видів: **Surface** (Поверхня), **Boundary Surface** (Гранична поверхня) і **Midsurface** (Серединна поверхня). Нижче розглядаються способи створення звичайних поверхонь, які можна вибрати в меню **Geometry – Surface** (рис. 6.13).

Поверхня	
<u>C</u> orners...	За кутами
E <u>d</u> ge <u>C</u> urves...	За кромками
<u>A</u> ligned <u>C</u> urves...	За рядом кромок
<u>R</u> uled...	Лінійчата
<u>E</u> xtrude...	Видавити
<u>R</u> evolve...	Обертати
<u>S</u> weep...	Витягнути по
<u>P</u> lane...	Площина
<u>C</u> ylinder...	Циліндр
<u>S</u> phere...	Сфера
<u>O</u> ffset...	Зміщення

Рис. 6.13

Розглянемо приклади створення поверхонь.

Для створення поверхні за чотирма точками виберемо пункт меню **Geometry – Surface – Corners** (Геометрія – Поверхня – За кутами) і у вікні, що відкрилося, введемо координати точок: X: -0.3, Y: 0.15, Z: 0; X: 0.2, Y: 0.35, Z: 0; X: 0.3, Y: -0.1, Z: 0; X: -0.2, Y: -0.15, Z: 0. Для завершення побудови натиснемо кнопку **Cancel**. Результат наведений на рис. 6.14.

Побудуємо поверхню по чотирьох лініях - точках, що утворять замкнутий контур.

Викличемо команду **Geometry – Line – Project Points** й введемо координати точок початку (X: -0.3, Y: 0.15, Z: 0) і кінця (X: 0.2, Y: 0.35, Z: 0) відрізка прямої. Викличемо команду **Geometry – Curve – Arc** й у підменю виберемо **Radius – Start – End**. Введемо в діалоговому вікні координати точок початку (X: 0.2, Y: 0.35, Z: 0) і кінця (X: 0.3, Y: -0.1, Z: 0) дуги, а потім її радіус (0.5). Побудуємо другу пряму з координатами початку (X:

0.3, $Y: -0.1, Z: 0$) і кінця ($X: -0.2, Y: -0.15, Z: 0$) відрізка. Потім побудуємо другу дугу з координатами точок початку ($X: -0.2, Y: -0.15, Z: 0$) і кінця ($X: -0.3, Y: 0.15, Z: 0$) дуги, а потім її радіус (0.6). Натиснемо кнопки OK і Cancel.

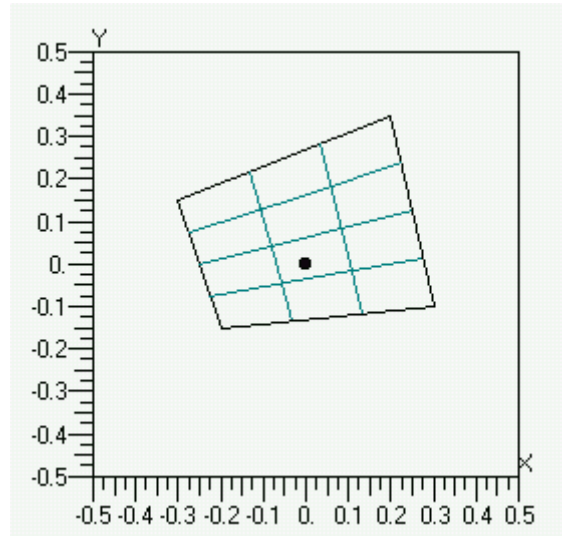


Рис. 6.14

Викличемо команду **Geometry – Surface – Edge Curves**. У діалоговому вікні послідовно відзначимо „мишею” точки і натиснемо кнопки OK і Cancel.

Результат наведений на рис. 6.15.

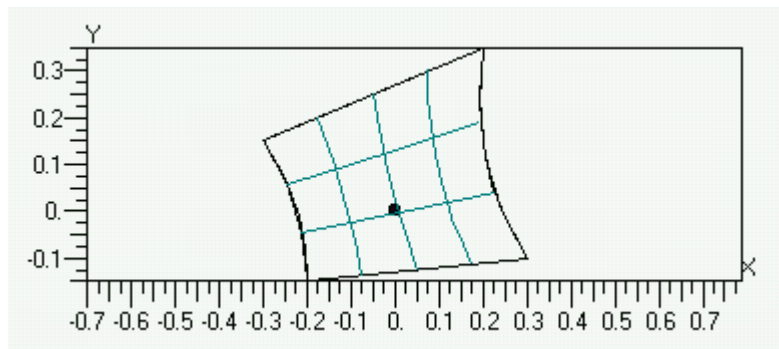


Рис. 6.15

Розглянемо метод утворення поверхні видавлюванням.

Спочатку створимо контур. Виберемо пункт меню **Geometry – Line – Coordinates** й введемо координати початку ($X: -0.35, Y: 0, Z: 0$) і кінця ($X: -0.2, Y: 0, Z: 0$) відрізка прямої. Викличемо команду **Geometry – Curve – Arc** й у підменю

виберемо **Radius – Start – End**. Введемо в діалоговому вікні координати точок початку (X: 0.2, Y: 0, Z: 0) і кінця (X: -0.2, Y: 0, Z: 0) дуги, а потім її радіус (0.25). Побудуємо другу пряму з координатами початку (X: 0.2, Y: 0, Z: 0) і кінця (X: 0.35, Y: 0, Z: 0) відрізка.

Викличемо команду **Geometry – Surface – Extrude**. У діалоговому меню, що з'явилося, відзначимо „мишею” вихідний контур і натиснемо ОК. Укажемо вектор напрямку видавлювання. Результат показаний на рис. 6.16.

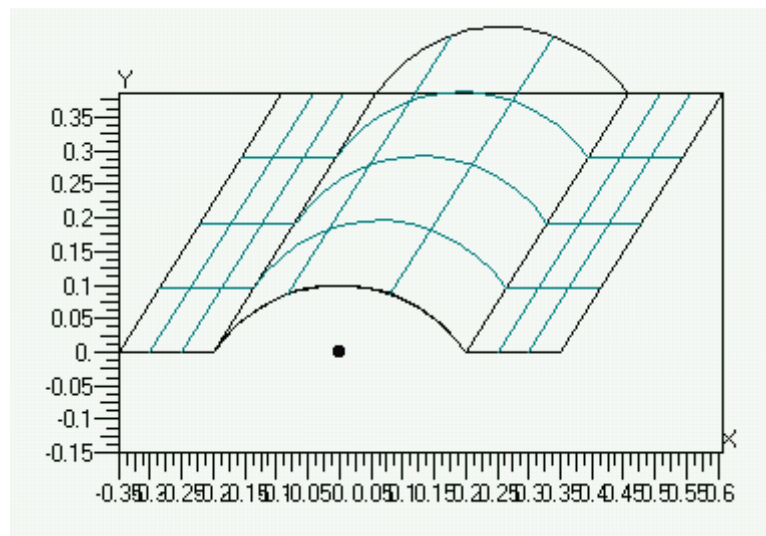


Рис. 6.16

Використовуючи команди **Plane** (Площина), **Cylinder** (Циліндр), **Sphere** (Сфера), можна створити площину, циліндричну (конічну) і сферичну поверхні (рис. 6.17).

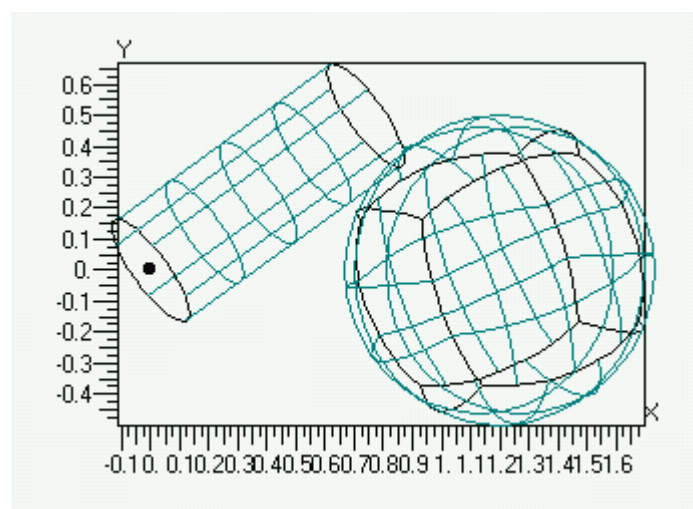


Рис. 6.17

6.5. Розрахунок важеля гальмової передачі вагона

Виконаємо розрахунок на міцність важеля гальмової передачі вагона. Важіль виконаний у вигляді пластини товщиною 0.014 м. До важеля прикладені горизонтальні сили: $P_1 = 27.35$ кН, $P_2 = 52.3$ кН, $P_3 = 25$ кН. Важіль виготовлений зі сталі Ст. 3. Розміри важеля й схема додатка навантажень показані на рис. 6.18.

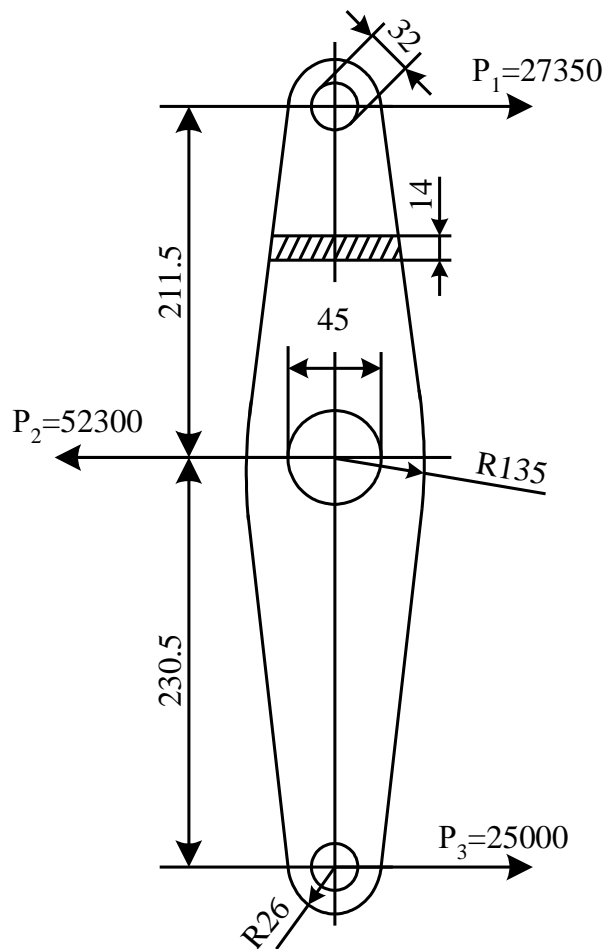


Рис. 6.18

Запустимо MSC/N4V і у діалоговому вікні виберемо пункт **New Model** і виконаємо налаштування. Виберемо в головному меню пункт **Tools – Workplane** (Інструменти – Робоча площина) і в діалоговому вікні **Workplane Management** (Керування робочою площиною) натиснемо кнопку **Snap Options** (Параметри

кроку). У діалоговому вікні **Snap To** (Крок к), що з'явилося, у розділі **Grid and Rules Spacing** (Розмір сітки й шкали) виберемо опцію **Nonuniform** (Різні) і у вікнах **X Grid** й **Y Grid** введемо значення 0.05 (м) для кроків по осях X й Y. Включимо опцію **Snap Grid** (Крок по сітці). Інші параметри залишимо без зміни. Натиснемо кнопку ОК і збережемо установки.







Побудову моделі почнемо із створення геометрії важеля.

Виберемо пункт меню **Geometry – Curve – Circle – Center** (Геометрія – Округність - Центр) і введемо координати центра отвору діаметром 0.045 **X: 0, Y: 0**. У діалоговому вікні введемо значення радіуса 0.0225, ОК.

Знову введемо координати центра округності **X: 0, Y: 0** і задамо значення радіуса 0.0675, ОК.

Побудуємо дві округності у верхній частині важеля. Для цього введемо координати центра **X: 0, Y: 0.2115**, ОК і значення радіуса 0.016 меншої округності, ОК. Знову введемо координати центра **X: 0, Y: 0.2115**, ОК і значення радіуса 0.026 більшої округності, ОК.

Тепер побудуємо дві округності в нижній частині важеля. Уведемо координати центра **X: 0, Y: -0.2305**, ОК і значення радіуса 0.016 меншої округності, ОК. Знову введемо координати центра **X: 0, Y: -0.2305**, ОК і значення радіуса 0.026 більшої округності, ОК, Cancel.

Збільшимо масштаб зображення за допомогою кнопок  й . Якщо буде потреба, кнопками , ,  і  можна переміщати зображення моделі.

Побудуємо тепер дотичні до зовнішніх округностей. Введемо команду **Geometry – Curve – Line – Tangent** (Геометрія – Пряма – Дотична). У діалоговому вікні виберемо за допомогою „миші” спочатку більшу округність, потім маленьку й натиснемо кнопку ОК. У діалоговому вікні, що з'явилося, **Locate - Enter Location Near Tangent on First Curve** (Положення - Задайте положення поблизу дотичної до першої округності) натисніть „мишею” поблизу точки торкання прямої з більшою округністю й натисніть ОК. Таким же чином побудуйте інші дотичні. Модель буде мати вигляд, показаний на рис. 6.19.

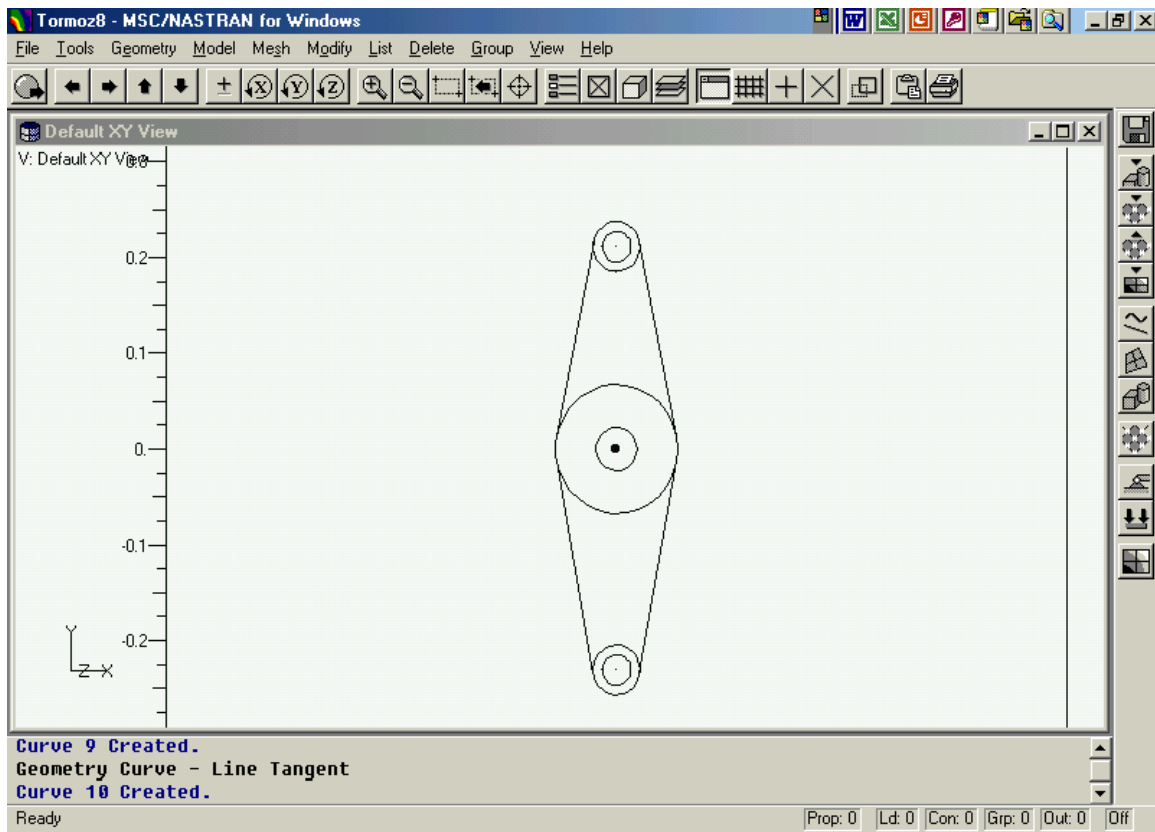



Рис.6.19

Видалимо більшу окружність у верхній частині моделі. Для цього за допомогою кнопки  (**Zoom**) збільшимо верхню частину зображення важеля. Виберемо в головному меню **Delete – Geometry – Curve**. З'явиться вікно, показане на рис. 6.20.

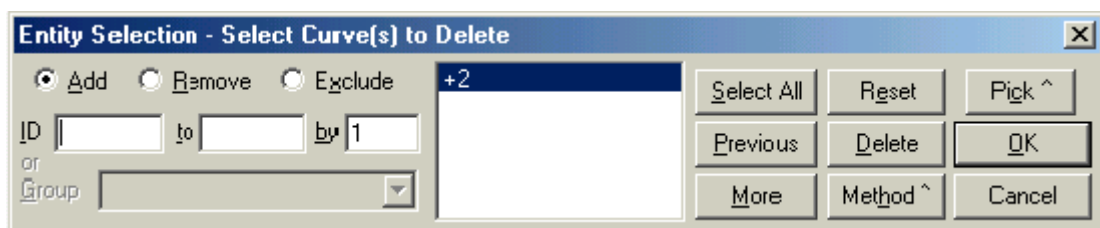


Рис. 6.20

Натиснемо на кнопку  (**Snap to Point**) для прив'язки до точок.

Укажіть „мишею” на окружність, яку потрібно видалити, і натисніть кнопку **OK**. Верхня частина моделі набуде вигляду, показаного на рис. 6.21,а.

Тепер побудуємо дугу окружності. Виберемо пункт меню **Geometry – Curve – Arc** і потім метод побудови дуги **Center – Start – End**. У діалоговому вікні, що з'явилося, укажемо мишею на центр дуги (рис 6.22) і натиснемо кнопку ОК. З'явиться подібне діалогове вікно, у якому вкажемо „мишею” на початок дуги (права точка), ОК, кінець дуги (ліва точка), ОК і натиснемо кнопку ОК, Cancel. Дуга побудована (рис. 6.21, б).

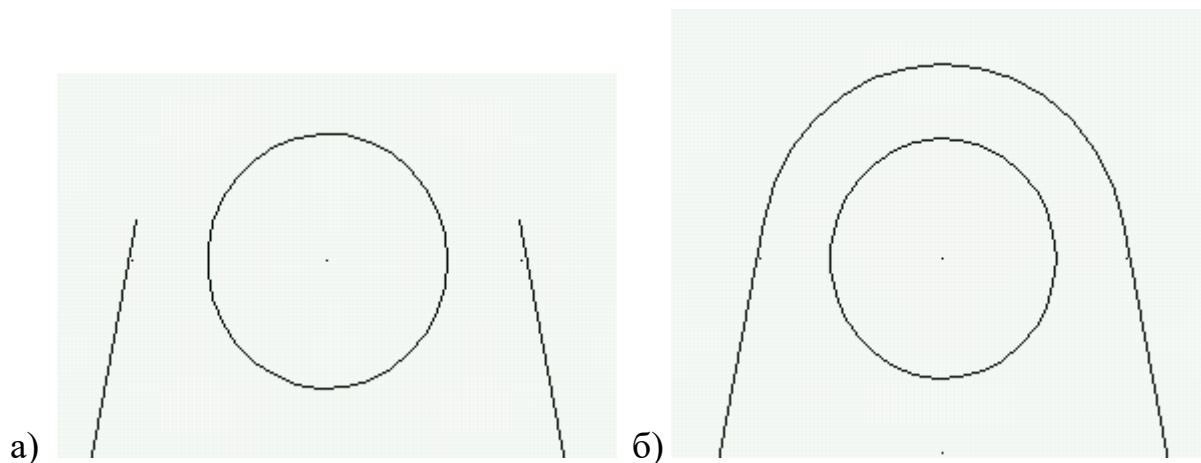


Рис. 6.21

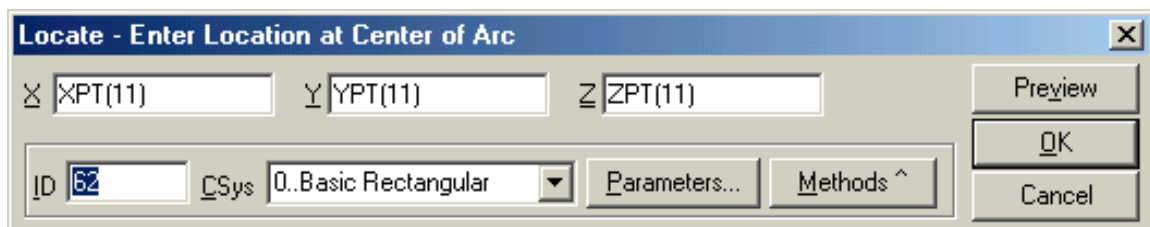


Рис. 6.22

Виконаємо такі ж дії (видалення окружностей і побудова дуг) для середньої й нижньої частини важеля. У результаті одержимо зображення важеля, наведене на рис. 6.23.

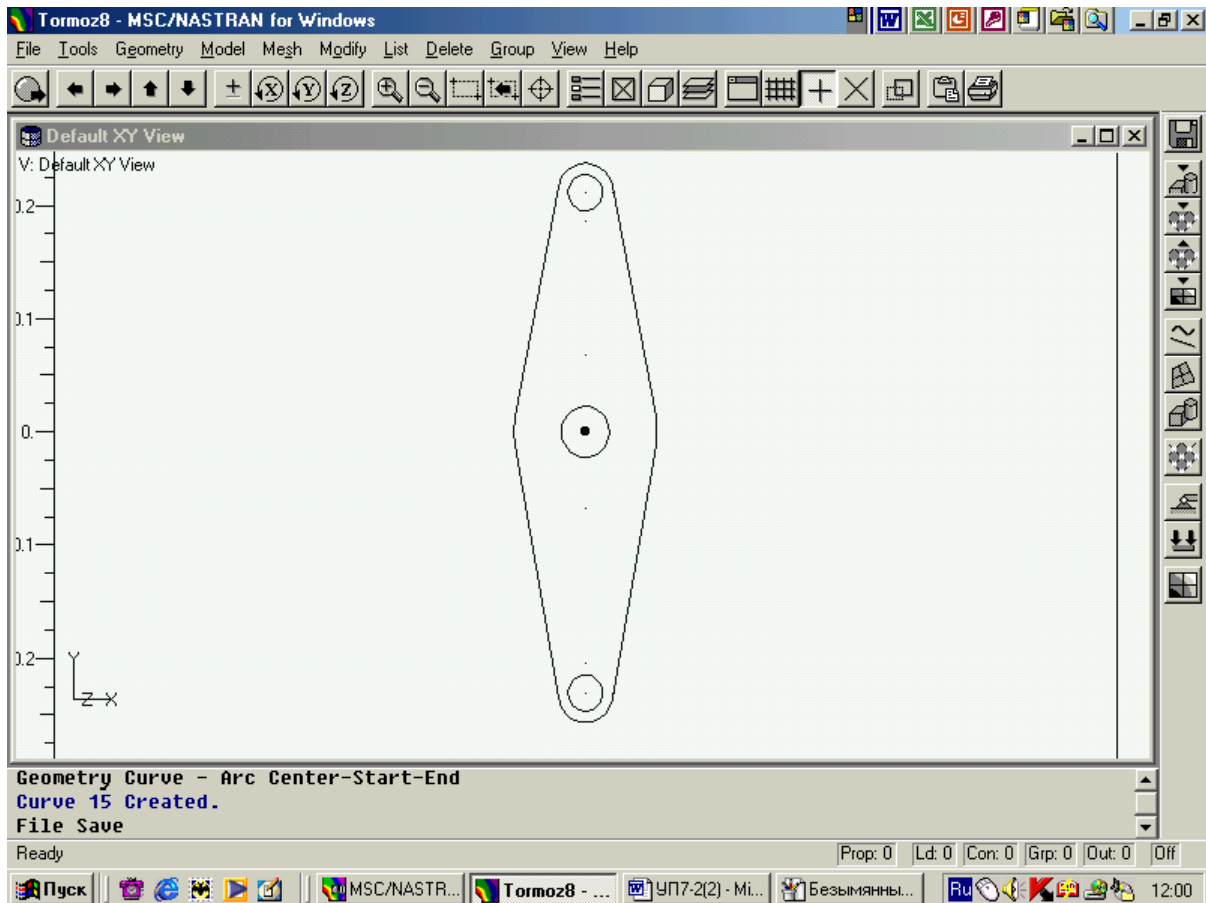


Рис. 6.23

У процесі побудови геометричної моделі можуть з'явитися точки, які надалі не будуть використовуватися і їх варто видалити. Це так звані точки, які не видаляють (NonDeleteable).

У головному меню виберемо пункт **Delete – Geometry – Points**. З'явиться діалогове вікно, показане на рис. 6.24.

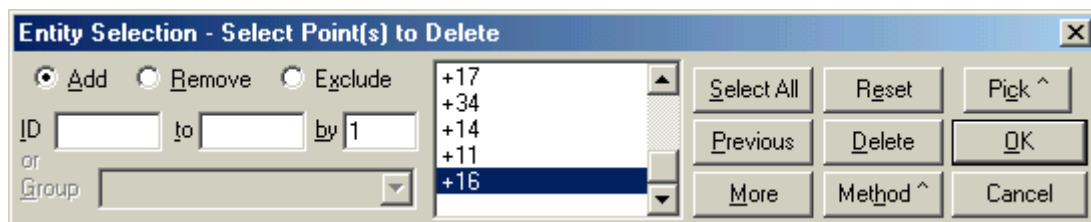


Рис. 6.24

Відзначимо „мишею” точки, які підлягають видаленню, і натиснемо кнопку ОК. Підтвердимо повідомлення про видалення точок.

У вікні повідомлень можна одержати інформацію про число вилучених точок. Приклад інформації показаний на рис. 6.25.

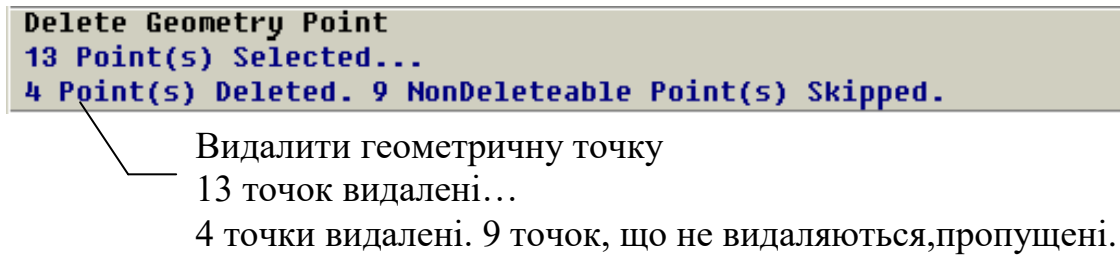


Рис. 6.25

Задамо характеристики матеріалу. Виберемо в головному меню пункт **Model – Material**. З'явиться діалогове вікно, показане на рис. 6.26.

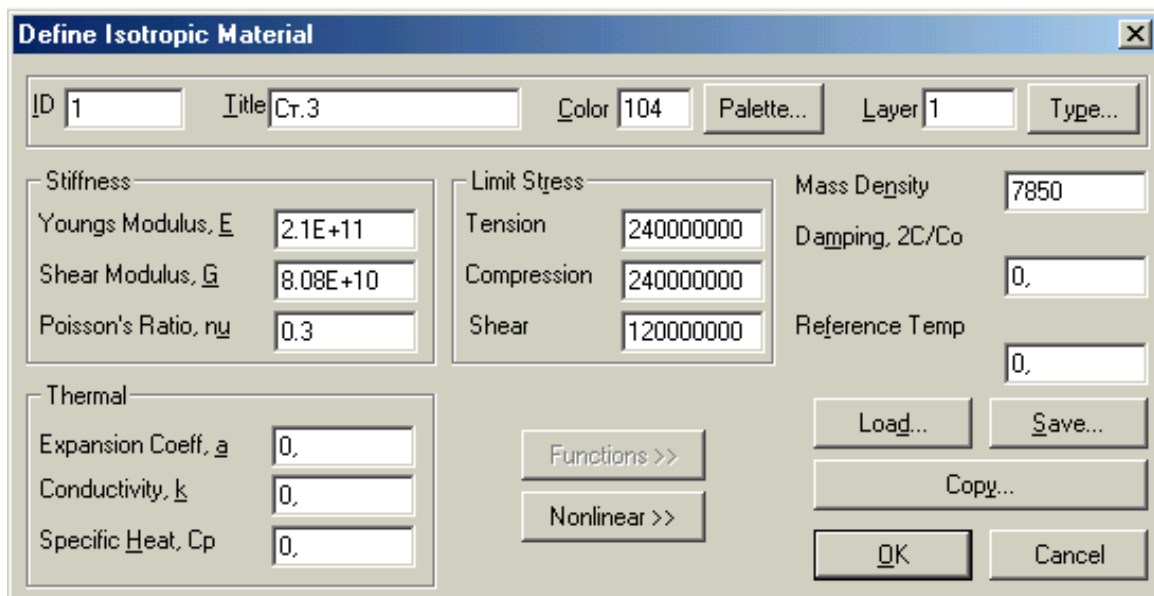


Рис. 6.26

У вікні введемо назву матеріалу (**Title**), модуль пружності (**Young, E**), модуль зрушення (**Sher Moduls, G**), коефіцієнт Пуассона (**Poisson's Ratio, nu**), а також граничні значення напруг (**Limit Stress**): межа текучості при розтяганні (**Tension**), стиску (**Compression**) і зрушенні (**Shear**). У цьому вікні також вводиться щільність матеріалу (**Mass Density**). Як одиниці вимірів приймаються кілограм (кг), ньютон (Н), метр (м). При цьому напруги будуть отримані в паскалях (Па).

Введемо характеристики матеріалу, як показано на рис. 6.26, і натиснемо ОК. Якщо після введення характеристик натиснути на кнопку **Save** (Зберегти), то вони будуть збережені в бібліотеці матеріалів. За допомогою кнопки **Load** (Завантажити) можна переглянути або завантажити матеріали, які є в бібліотеці. Після натискання на кнопку **Load**, відкриється вікно, показане на рис. 6.27. У цьому вікні можна вибрати необхідний матеріал, зробити пошук або підключити іншу бібліотеку.

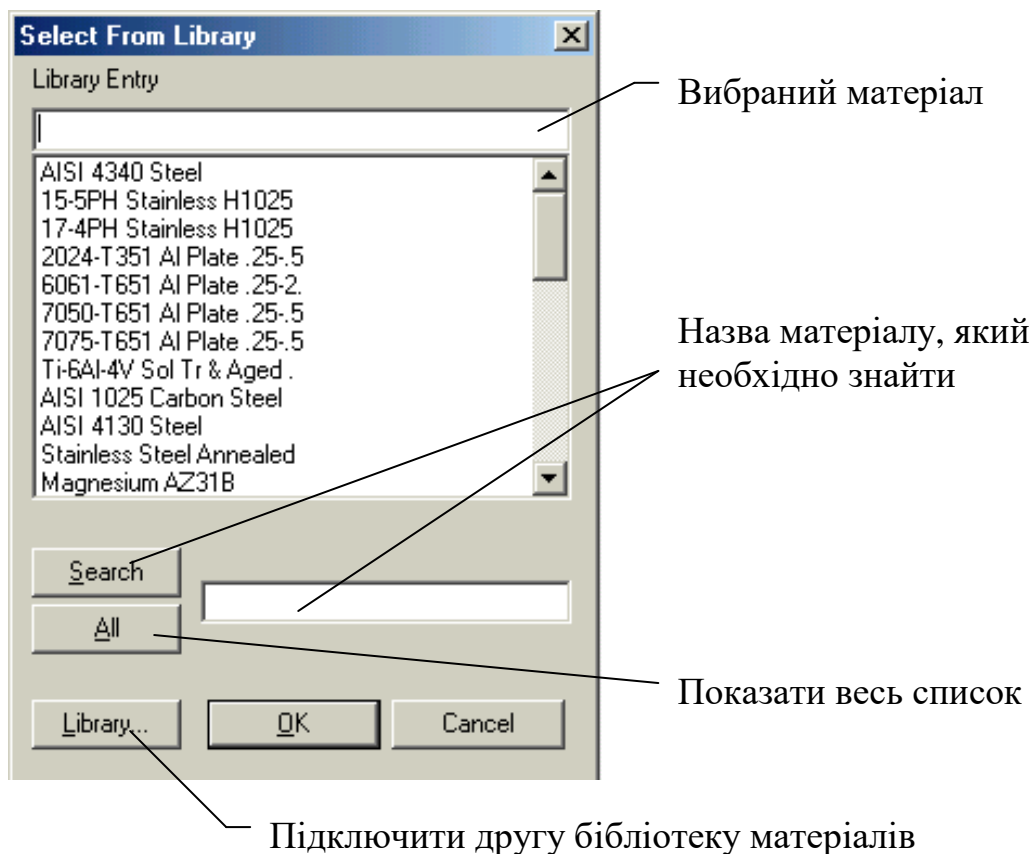


Рис. 6.27

Натиснемо на кнопку **Cancel**, потім на кнопку **OK**. Знову з'явиться вікно для введення характеристик матеріалу, показане на рис. 6.27. Тепер у вікні **ID** буде зазначений номер матеріалу **ID=2**. Це вікно необхідно заповнювати в тій випадку, коли в конструкції використовується два різних матеріали.

Наступний етап розрахунку складається з вибору типу й параметрів кінцевих елементів.

Для задання властивостей моделі виберемо пункт меню **Model - Property** (Модель – Властивості). З'явиться діалогове вікно **Define Property** (Задати властивості). У заголовку вікна (рис. 6.28) можна побачити тип елемента **Plate** (Пластина). Цей тип кінцевих елементів встановлений за замовчуванням.

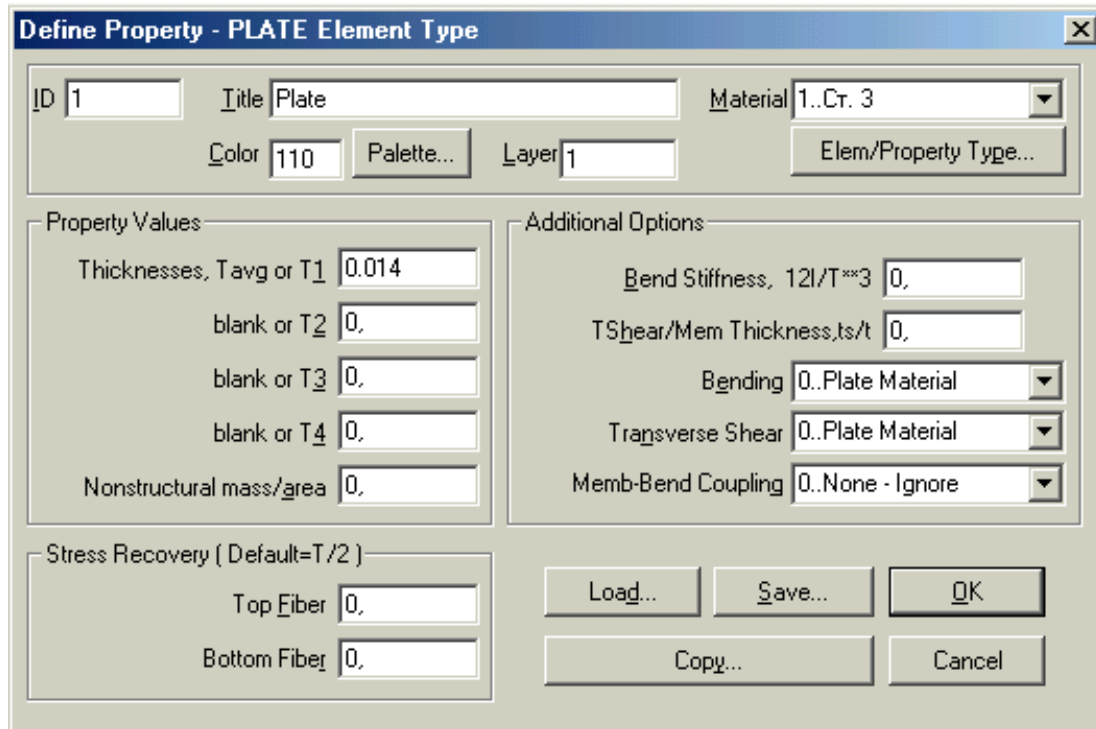


Рис. 6.28

Виберемо назву типу елемента (**Title**), встановимо кольори (**Color**). У списку, що відкривається, виберемо матеріал (**Material**), у розділі **Property Values** (Значення параметрів), **Thicknesses** (Товщина) введемо значення 0.014 (14 мм). Значення інших параметрів приймемо за замовчуванням. Для завершення натиснемо клавішу ОК. Знову з'явиться вікно, в якому можна ввести параметри елементів іншого типу. Натиснемо клавішу Cancel.

Визначимо область, що буде розбиватися на кінцеві елементи. Виберемо пункт меню **Geometry – Boundary Surface** (Геометрія – Гранична поверхня).

У діалоговому вікні вкажемо „мишею” лінії, що обмежують область розбивання, і натиснемо кнопку ОК. Можна натиснути на

кнопку **Select All** (Вибрати всі), потім на кнопку ОК і на кнопку Cancel.

Граничні області будуть сформовані, про що можливо судити по товщині ліній, що обмежують контури моделі (рис. 6.29).

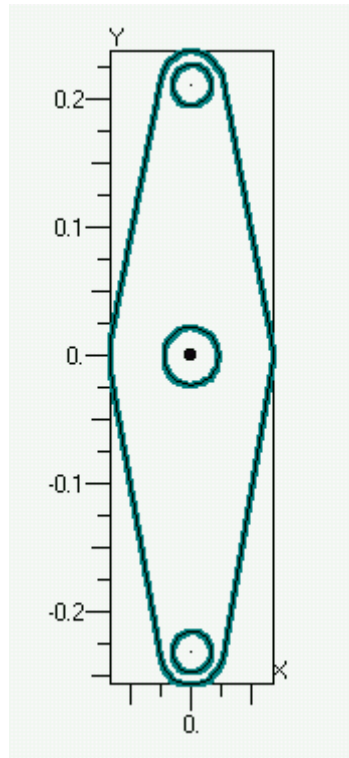


Рис. 6.29

Задамо розміри звичайно – елементної сітки. Виберемо в головному меню пункт **Mesh – Mesh Control – Define Size** (Сітка – Керування сіткою – Задати розмір). У вікні **Default Mesh Size** введемо розмір кінцевого елемента 0.01 (10 мм) і натиснемо ОК (рис. 6.30).

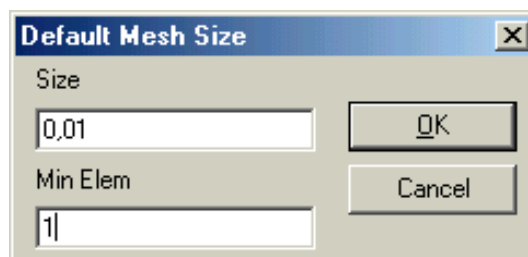


Рис. 6.30

Виконаємо розбивання моделі на кінцеві елементи. У головному меню виберемо пункт **Mesh – Geometry - Surface** (Сітка – Геометрія – Поверхня) і в діалоговому вікні, що з'явилось, **Entity Selection** (Вибір об'єкта) натиснемо на **кнопку Select All, OK**.

З'явиться діалогове вікно **Automesh Surface** (Авторозбивання поверхонь) (рис. 6.31), у якому виберемо зі списку Property (Властивості) елемент із ім'ям Plate і натиснемо на кнопку **OK**.

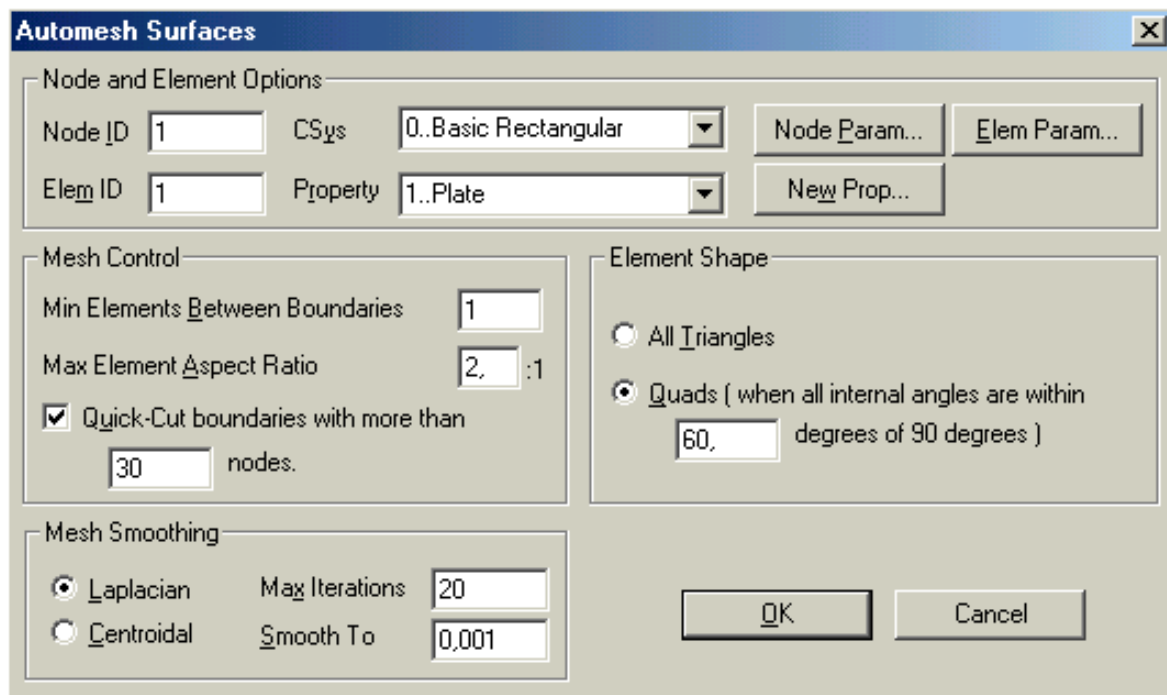


Рис. 6.31

На граничній поверхні з'явиться звичайно-елементна сітка (рис. 6.32). У рядку підказки з'явиться повідомлення про число вузлів і число елементів: **Nodes: 601, Elements: 539**.

Наступним етапом підготовки розрахункової моделі є задання граничних умов.

У головному меню виберемо пункт **Model – Constraint – Set** (Модель – Граничні умови – Варіант).

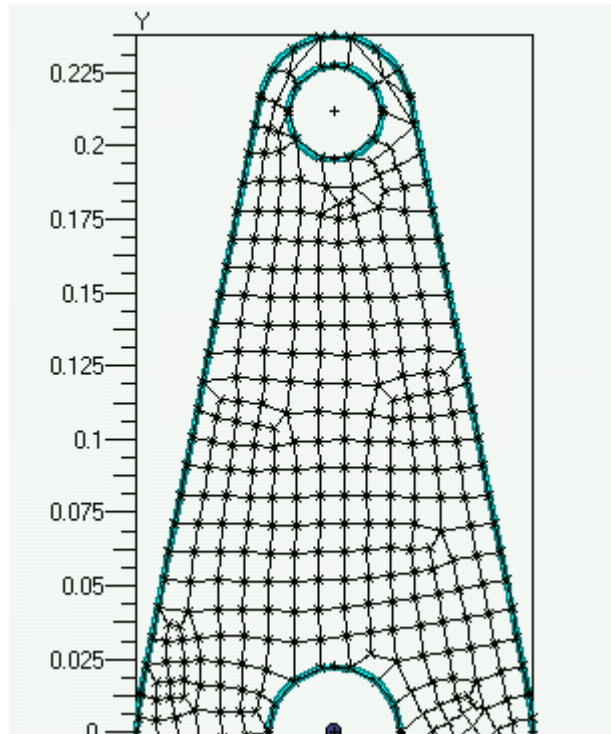


Рис. 6.32

З'явиться вікно (рис. 6.33), у якому в поле **Title** введемо назву варіанта граничних умов – «Шарнір середній і нижній» і натиснемо кнопку ОК.

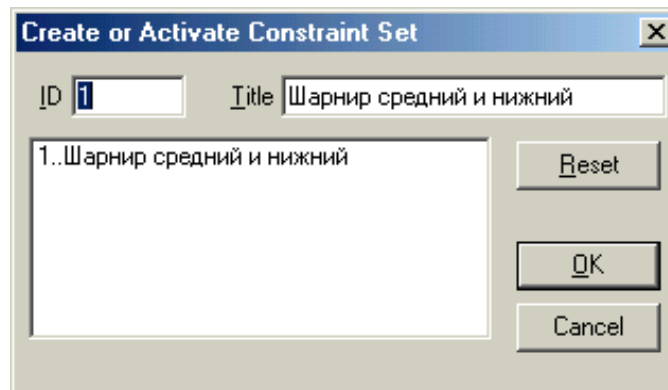


Рис. 6.33

Можна створити кілька варіантів граничних умов.

Виберемо пункт меню **Model – Constraint – On Curve** (Модель – Граничні умови – На кривій). З'явиться вікно, у якому відзначимо „мишею” центральний і нижній отвори й натиснемо

кнопку ОК. У діалоговому вікні, що з'явилось, **Create Constraint on Geometry** (Створити граничні умови на геометрії) у розділі **DOF (Degrees of Freedom)** - Ступені свободи виберемо другий варіант граничних умов - **Pinned - No Translation** (Шарнір - Заборонені переміщення) і натиснемо на кнопку ОК. Вікно повинне мати вигляд, показаний на рис. 6.34.

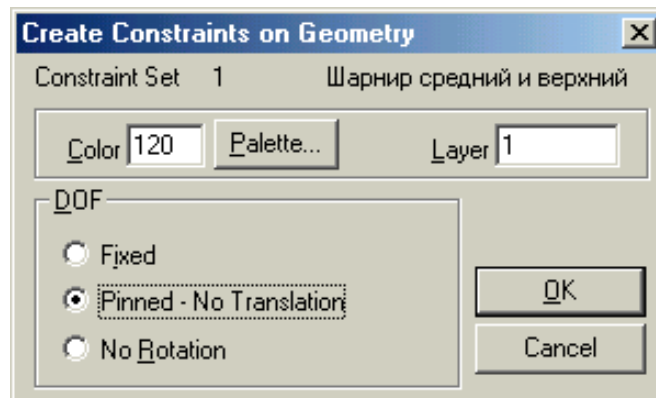


Рис. 6.34

Задамо навантаження, що прикладають до моделі.

Виберемо пункт меню **Model – Load - Set** (Модель – Навантаження – Варіант). З'явиться вікно (рис. 6.35), яке подібно вікну, показаному на рис. 6.33. У поле **Title** введемо назву варіанта навантаження – $F_x = 5470$ Н і натиснемо кнопку ОК.

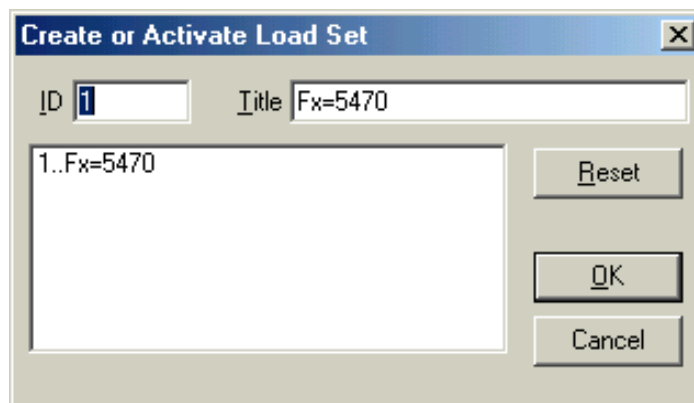



Рис. 6.35

За допомогою кнопки  збільшимо масштаб зображення важеля поблизу верхнього отвору (рис. 6.36).

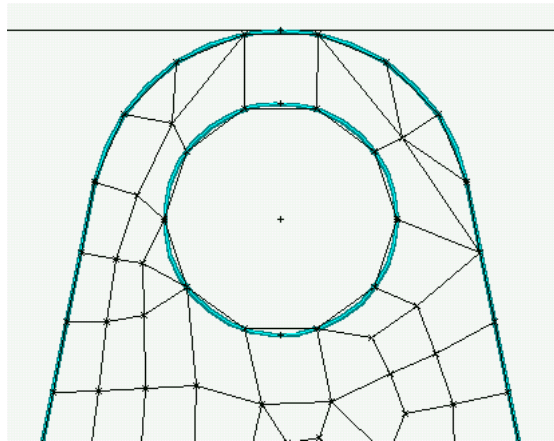


Рис. 6.36

Виберемо пункт меню **Model – Load - Nodal** (Модель – Навантаження – Вузлова), виділимо „мишею” п’ять вузлів на окружності й натиснемо кнопку ОК (рис. 6.37).

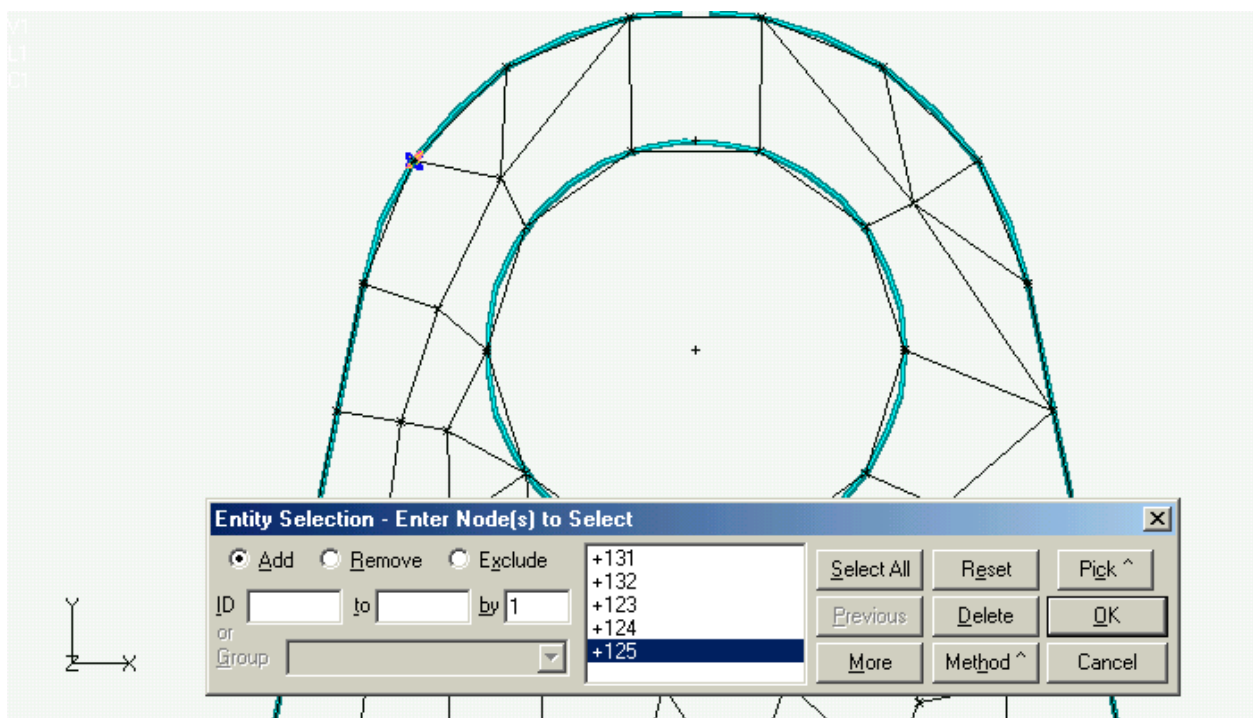


Рис. 6.37

З’явиться вікно **Create Loads on Nodes** (Створити навантаження у вузлах), у якому виберемо **Force** (Сила), **Components** (Компоненти) і введемо значення навантаження у вузлі $F_x = 5470$ (рис. 6.38). Натиснемо кнопку ОК, Cancel.

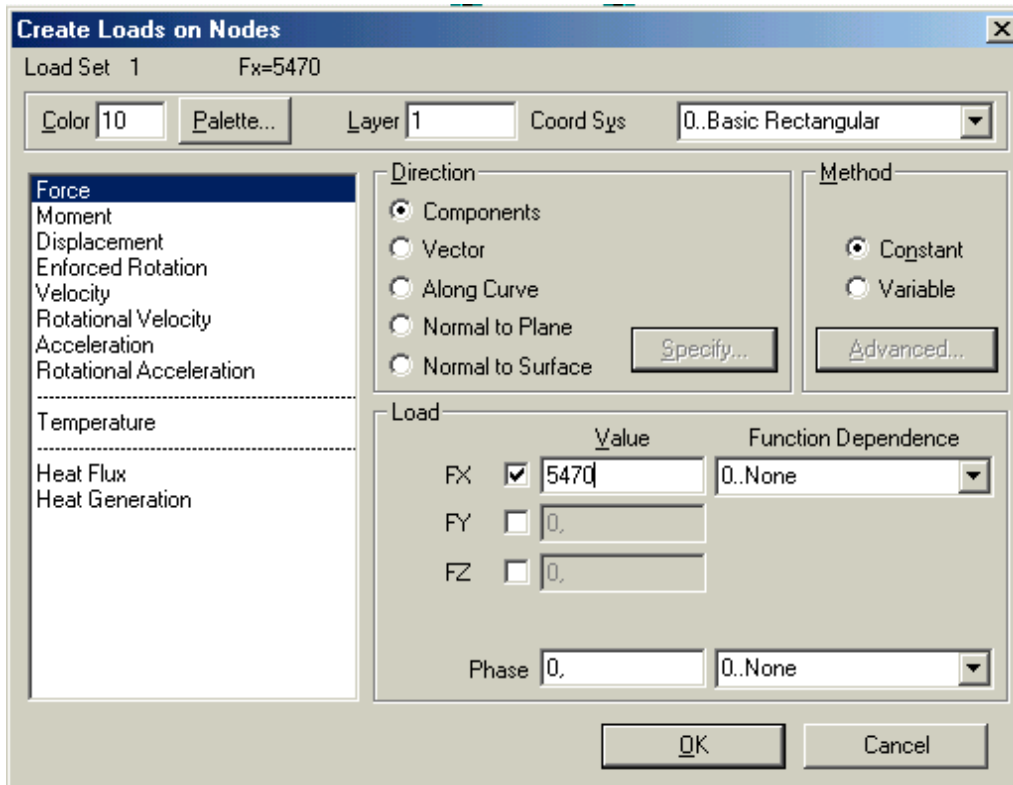


Рис. 6.38

Навантаження у вигляді стрілок і числових значень будуть відображатися на моделі, як показано на рис. 6.39.

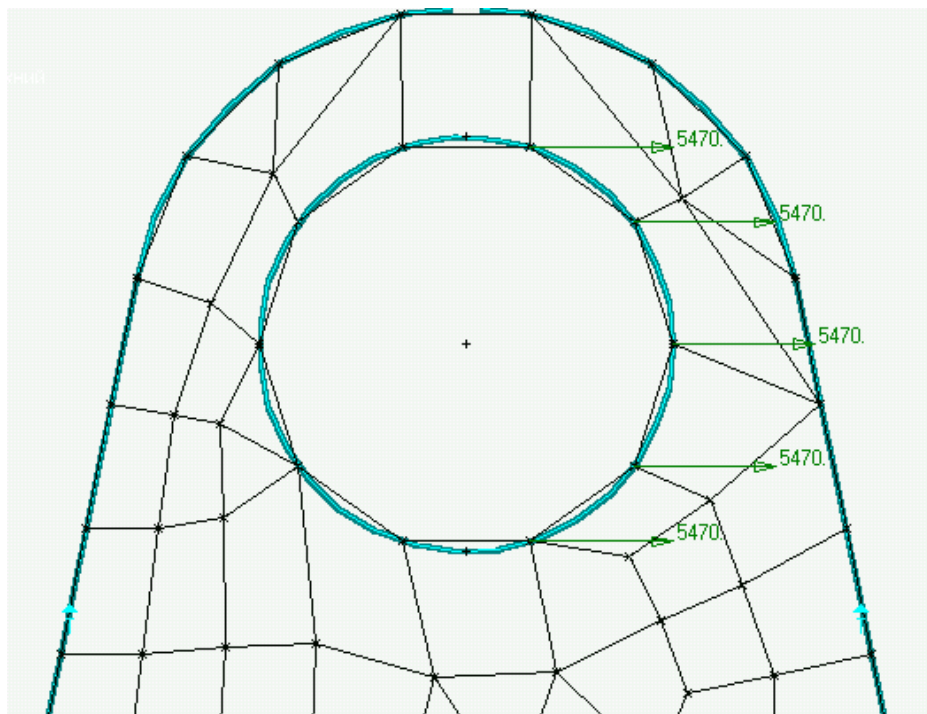


Рис. 6.39

У лівому верхньому куті графічного вікна можна показати позначення поточних настроювань моделі: вид - V (View - Вид), L (Load - Навантаження), C (Constraint - Граничні умови).

Виберемо в головному меню пункт **View – Options** і у вікні, яке з'явилося, категорію **Tools and View Style** (Інструменти і стиль виду). Встановимо параметр **View Legend** (Легенда), у розділі **Legend Style** (Стиль легенди) - пункт **1..Titles** (Назва) і натиснемо кнопку **Apply** (Застосувати). Крім того, видалимо позначення типу граничних умов – букву **F**, а також значення сили – **5470**. Для цього виберемо категорію **Labels, Entities and Colors**. У цій категорії виберемо параметр **Constraint** і пункт **0..No Labels** (Без мітки) розділу **Label Mode** (Режим мітки). Далі виберемо параметр **Load - Force**, встановимо **Label Mode** у положення **0..No Labels**, натиснемо **Apply**, ОК.

Підготовча робота виконана й можна приступати до розрахунку.

Будемо виконувати два види розрахунку: статичний розрахунок і розрахунок важеля на стійкість.

Виберемо пункт меню **File - Analyze** (Файл – Розрахунок, аналіз). З'явиться діалогове меню **NASTRAN Analysis Control** (Керування параметрами розрахунку) – рисунок 6.40.

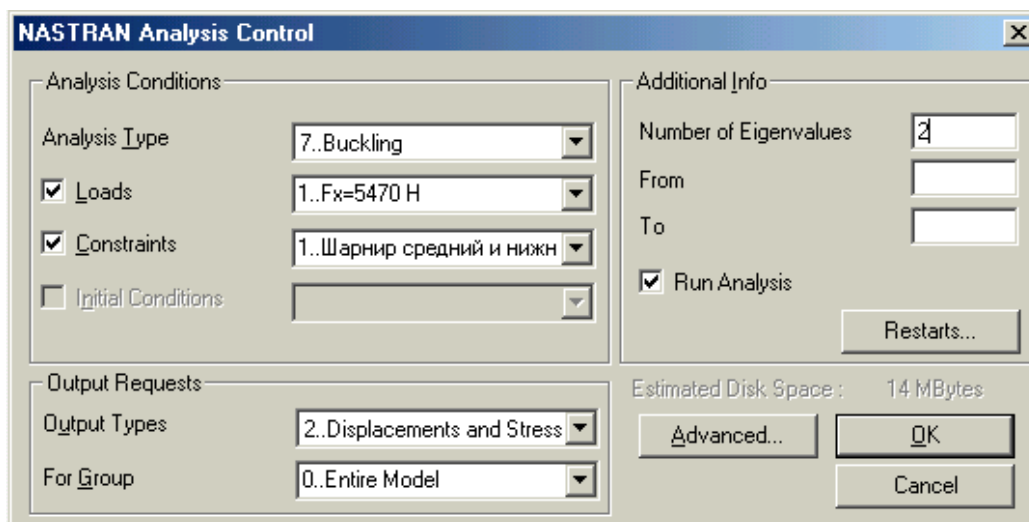


Рис. 6.40

У розділі **Analysis Conditions** (Умови розрахунку) вказуються види розрахунку. За замовчуванням зазначений вид розрахунку **Analysis Type: 1..Static** - статичний.

У списку, що розкривається, виберемо пункт 7..Buckling (Стійкість). Оскільки задано один варіант навантаження і граничних умов, то у вікні буде зазначений варіант навантаження - Loads: 1..Fx=5470 Н і варіант граничних умов - Constraints: 1..Шарнір середній і нижній.

У розділі **Output Requests** (Необхідні вихідні результати) типи вихідних параметрів (**Output Types**): 0..Standard - стандартний, 1..Displacements Only - тільки переміщення, 2..Displacements and Stresses - переміщення й напруги, All - всі параметри. Виберемо тип **Output Type**: 2..Displacements and Stresses (Переміщення і напруги).

У пункті **For Group** (Для групи) виберемо встановлений за замовчуванням параметр 0..Entire Model (Вся модель).

При розрахунку на стійкість стає активним розділ вікна **Additional Info** (Додаткова інформація). Тут можна вказати число критичних значень навантаження - власних значень параметра β (**Number of Eigenvalues**). За замовчуванням параметр **Number of Eigenvalues** дорівнює 1. У поля From (Від) і To (До) можна ввести межі, у яких буде перебувати значення параметра β . Практичне значення має найменше значення параметра β , що дає можливість визначити найменшу величину навантаження, при якій відбудеться втрата стійкості.

В діалоговому вікні вказується необхідний обсяг дискового простору – **Estimated Disk Space**. У нашому випадку для виконання розрахунку потрібно 14 Mbytes.

Після натискання кнопки ОК буде виконана трансляція об'єктів моделі. Програма видасть запит про збереження (рис. 6.41), на який відповімо ствердно – натиснемо кнопку **Так**.

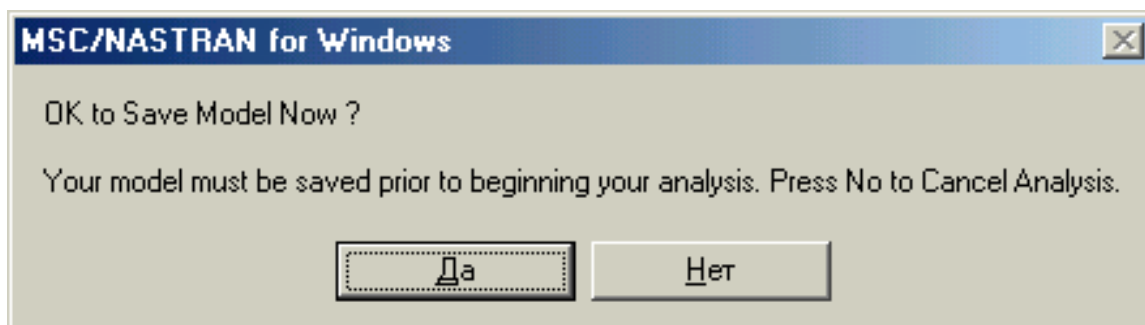


Рис. 6.41

Програма виконає розрахунок і виведе повідомлення у вікні (Огляд повідомлень), показане на рис. 6.42.

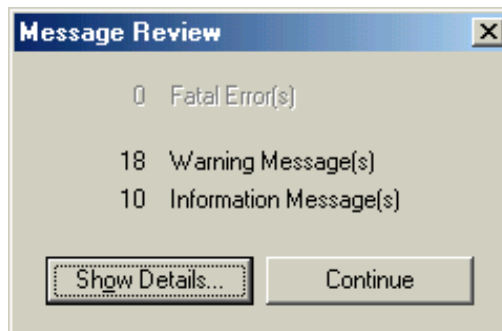




Рис. 6.42

З повідомлення випливає, що фатальних помилок не виявлено (**0 Fatal Error(s)**). Докладну інформацію про попереджувальні (**Warning Message(s)**) і інформаційні (**Information Message(s)**) повідомлення можна отримати, натиснувши кнопку **Show Details**.

Натиснемо кнопку **Continue** (Продовжити). Результати розрахунку будуть збережені у файлі моделі.

Виконаємо підготовчі операції по установленню режимів і параметрів перегляду результатів.

Натиснемо кнопку **PostProcess** (Результати)  на панелі команд (див. рис. 6.2). Панель набуде вигляду, показаного на рис. 6.3. Потім натиснемо на кнопку **Post Data** (Вихідні дані)  на панелі команд. З'явиться діалогове вікно **Select PostProcessing Data** (Вибір вихідних даних), показане на рис. 6.43.

Діалогове вікно містить п'ять розділів.

У розділі **Data Selection** (Вибір даних) можна вибрати категорію (**Category**) і тип (**Type**) вихідних даних. У списку, що розкривається, **Category** (Категорія) виберемо 0..Any Output (Будь-яку) і в списку **Type** (Тип) 0..Value or Magnitude - (Значення або амплітуда).

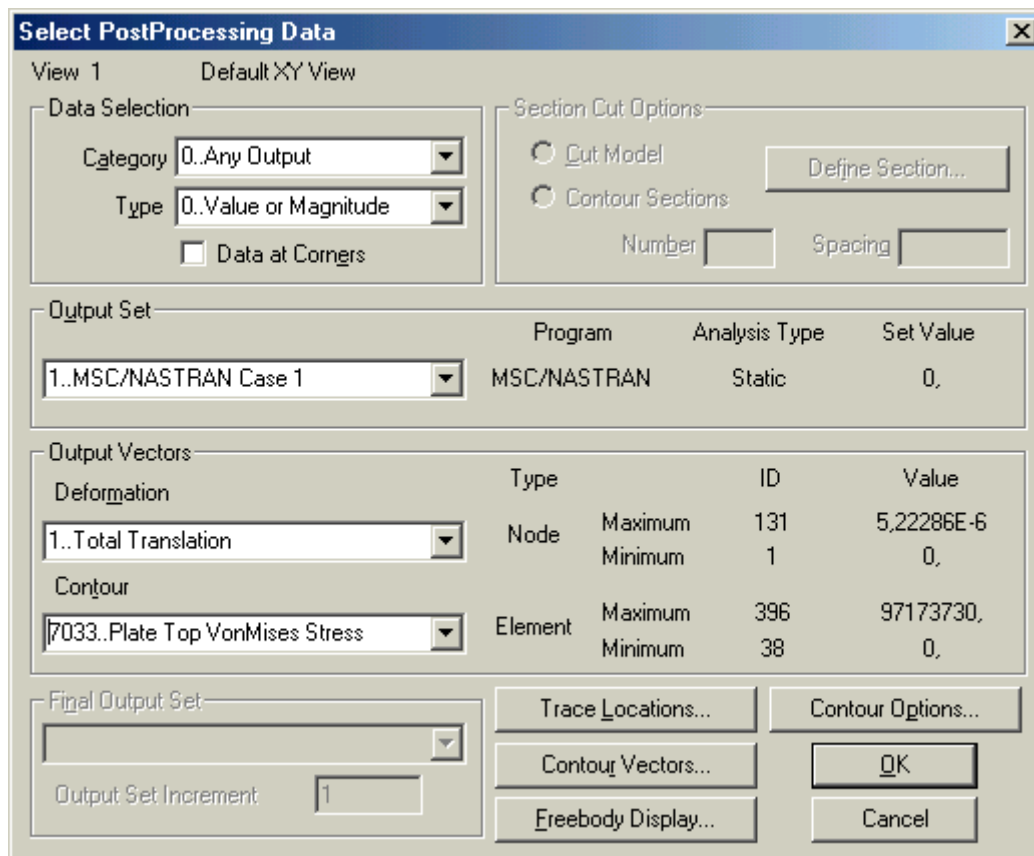




Рис. 6.43

У розділі **Output Set** (Вихідний набір даних) у списку, що розкривається, можна побачити три набори вихідних даних: 1.MSC/NASTRAN Case 1; 2.Eigenvalue 1 -17.2168; 3.Eigenvalue 2 19.52901. Перший набір вихідних даних відноситься до статичного розрахунку, другий - до розрахунку першого критичного навантаження, третій - до розрахунку другого критичного навантаження. Виконаємо спочатку статичний розрахунок, тобто виберемо зі списку 1.MSC/NASTRAN Case 1.

В розділі **Output Vectors** (Вихідні вектори) виберемо для **Deformation** (Деформації) – **Total Translation** (Сумарні переміщення), а для **Contour** (Контур) – **Plate Top VonMises Stress** (Еквівалентні напруги на верхньому боці пластини, обчислені за гіпотезою Мизеса). Тут же вказується максимальне (**Maximum**) і мінімальне (**Minimum**) значення (**Value**) вихідного вектора для вузлів (**Node**) і елементів (**Element**), а також номери вузлів й елементів (**ID**), у яких досягаються ці значення. Натиснемо кнопку ОК, а потім на кнопку **Contour** (Контур) .

На екрані з'явиться кольорове відображення розподілу напруг (рис. 6.44). У правій частині вікна розташована шкала значень (Паскалів) Па.

Якщо натиснути на кнопку **Deformed** (Деформований стан) , то можна побачити важіль у деформованому стані (рис. 6.45).

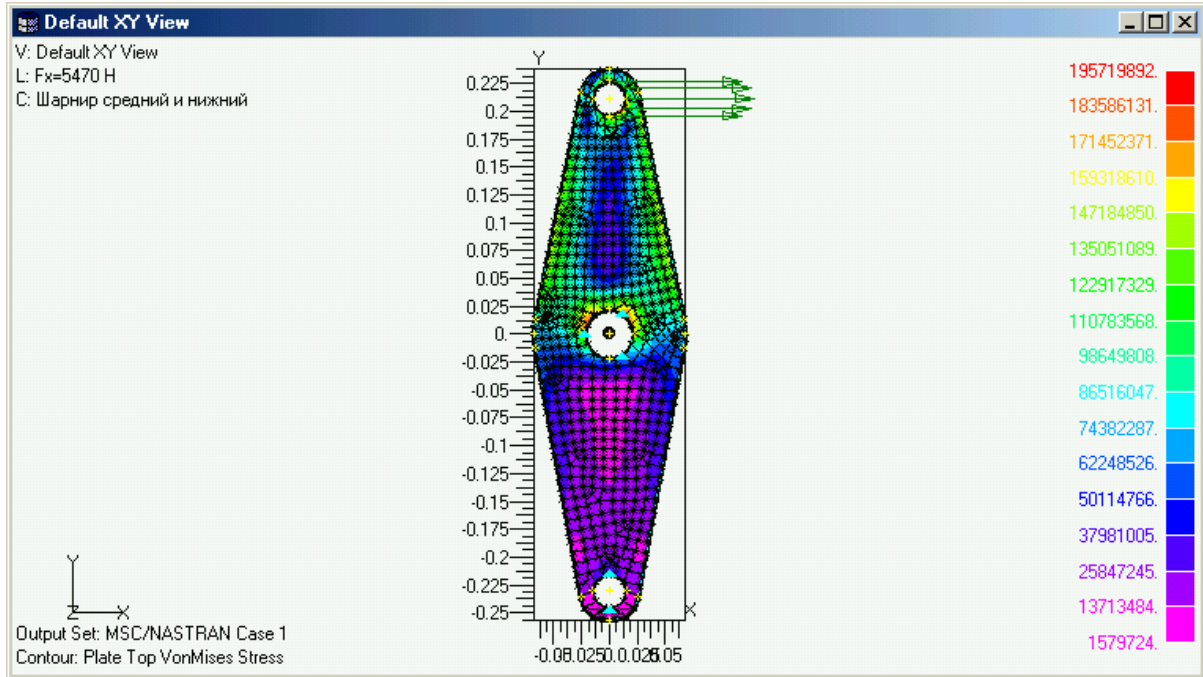


Рис.6.44

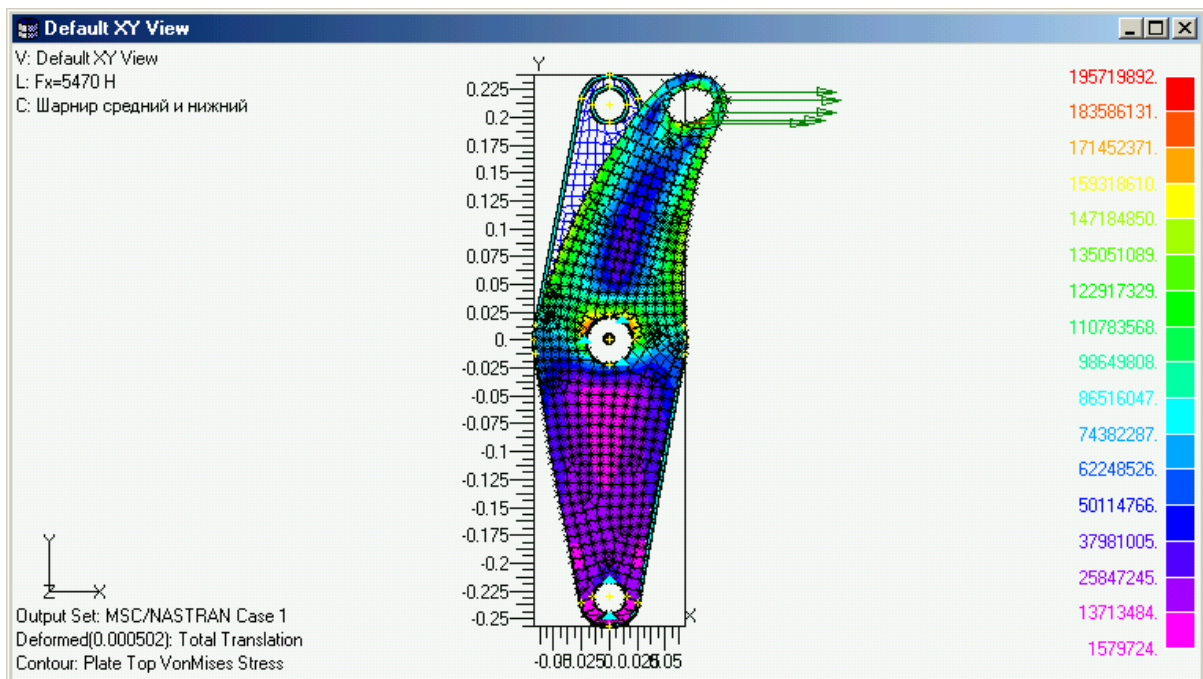



Рис.6.45

Відформатуємо зображення моделі. Натиснемо на панелі інструментів **View** кнопку  (Швидкі опції). З'явиться діалогове вікно, показане на рис. 6.46. Натиснемо кнопку **Geometry Off** (Виключити геометрію). Крім того, знімемо прапорець **Node** (Вузли) і натиснемо кнопку **Done**. В результаті із зображення моделі будуть вилучені точки (**Point**), криві (**Curve**), поверхні (**Surface**) і інші елементи геометрії.

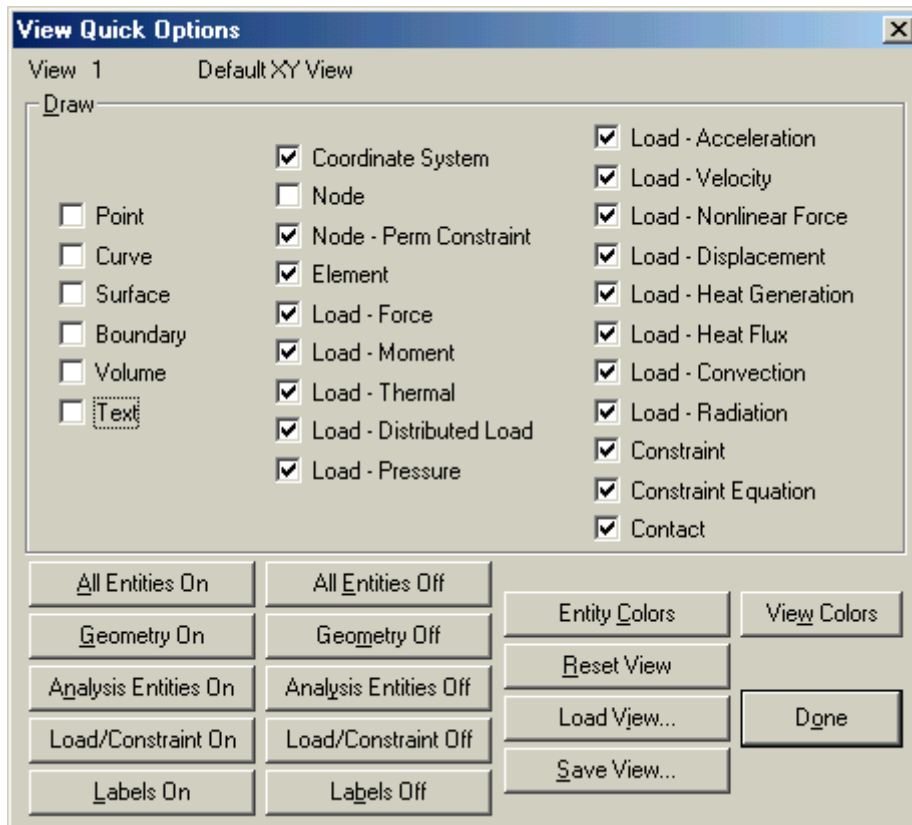



Рис.6.46

Можливо використати й інші засоби форматування. На панелі інструментів **View** натиснемо кнопку **View Style** (Стиль виду) . З'явиться меню (рис.6.47), у якому можна вибрати додаткові засоби відображення моделі.

Виберемо в меню пункт **Render Solid** (Тоноване тіло). Автоматично вклучаться опції **Render** (Тон), **Fill** (Заливання), **Shading** (Затінення).

У головному меню виберемо пункт **View Options** (Опції виду) або натиснемо клавішу **F6**. Встановимо категорію **Tools**

and View Style (Інструменти й стиль виду). У розділі **Options** (Опції) встановимо **Workplane and Rulers** (Робоча площина й шкали). У розділі **Rulers** виберемо пункт 1..Skip Rulers, у розділі **View Color** виберемо білі кольори - 149.

<u>W</u> ireframe	Каркас
<u>H</u> idden	Сховані лінії
<u>S</u> olid	Тіло
<u>R</u> endered Solid	Тоноване тіло
<input checked="" type="checkbox"/> <u>R</u> ender	Тон
<input checked="" type="checkbox"/> <u>F</u> ill	Заливка
<input checked="" type="checkbox"/> <u>S</u> hading	Затінення
<input checked="" type="checkbox"/> <u>M</u> esh Size	Розмір сітки
<u>S</u> hrink	Стиснути
<input checked="" type="checkbox"/> <u>O</u> ffsets	Зміщення
<u>O</u> rientation	Орієнтація
<u>O</u> ptions...	Опції (виклик вікна View Options)

Рис.6.47

Після цього модель набуде вигляду, показаного на рис. 6.48.

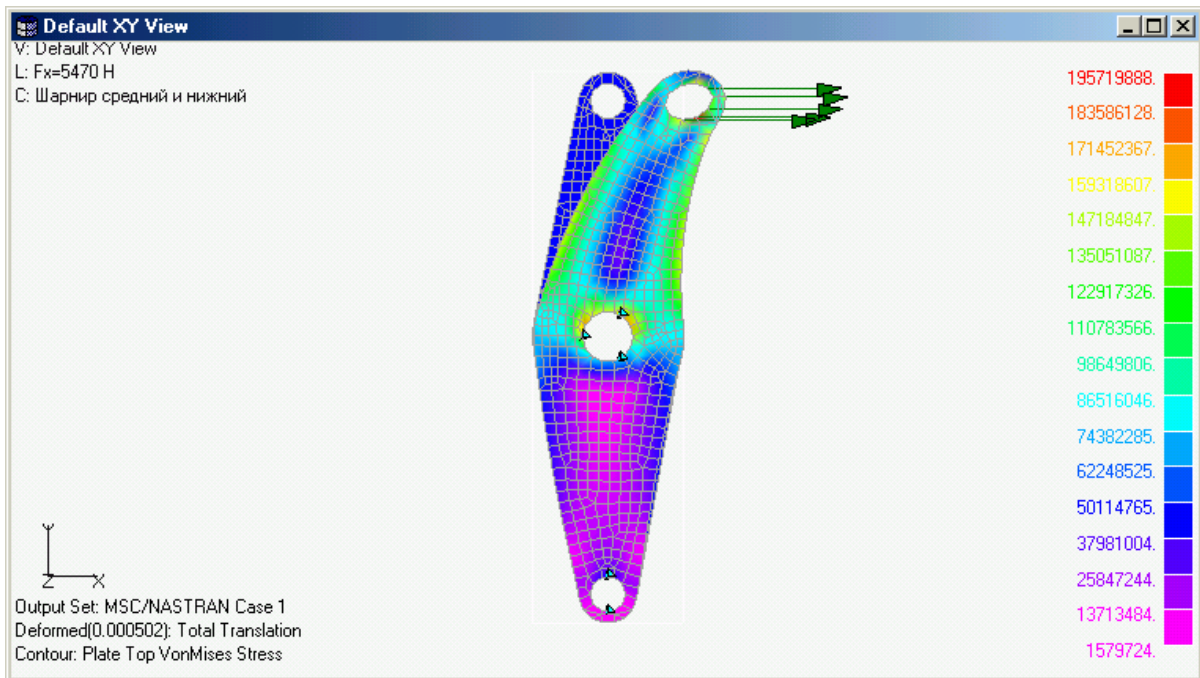




Рис.6.48

Найбільші напруги у важелі становлять $\sigma_{\max} = 195.7$ МПа.

Коефіцієнт запасу міцності по границі текучості дорівнює $n_T = \frac{250}{195.7} = 1.28$, тобто, важіль задовольняє вимоги міцності.

Перейдемо до наступного варіанта розрахунку - 2.Eigenvalue 1 -17.57603. На панелі команд натиснемо кнопку **Post Data** (Вихідні дані) . З'явиться діалогове меню (рис. 7.65), у якому в розділі **Output Set** виберемо варіант 2..Eigenvalue 1 -17.57603 і натиснемо кнопку ОК. За допомогою кнопки  встановимо модель у положення, яке найбільш зручне для перегляду. У головному меню виберемо пункт **View - Regenerate** (Вид - Регенерувати). Результати розрахунку цього варіанта показані на рис. 6.49.

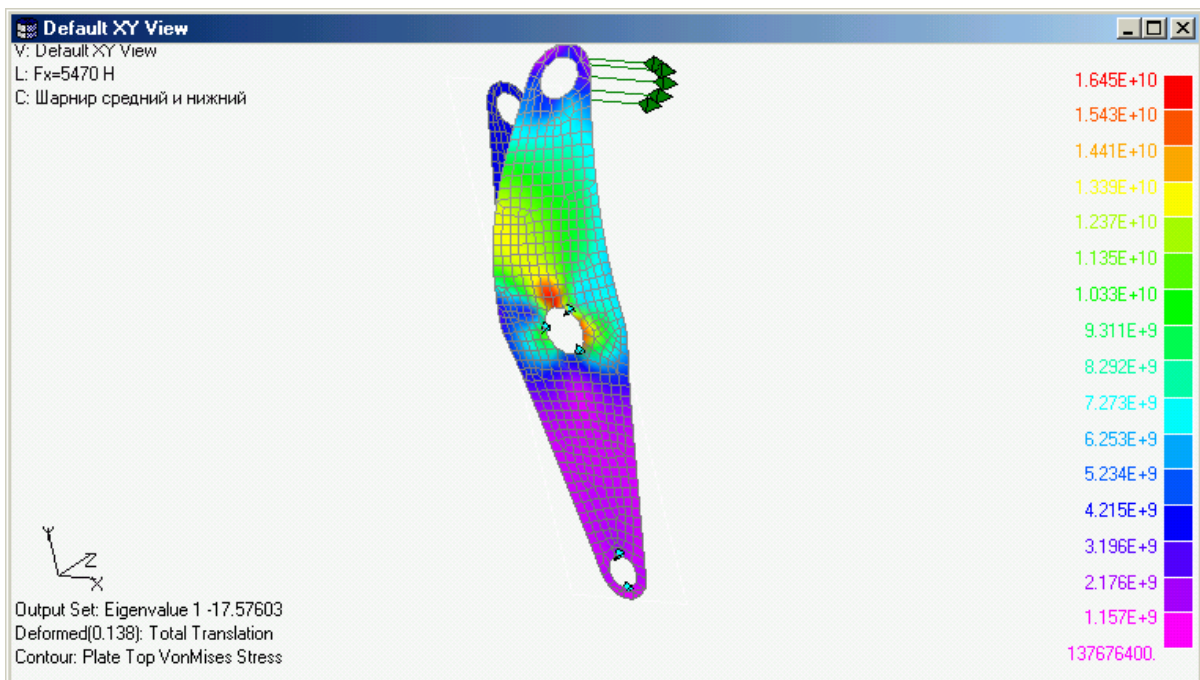



Рис. 6.49

Деформації важеля відбуваються як у площині важеля, так і поза площиною.

У лівому нижньому куті вікна зазначене перше власне значення параметра β_1 (Eigenvalue 1 -17.57603).

Перше критичне навантаження дорівнює $P_1^{kp} = \beta_1 P_1 = 17.576 \times 27.35 = 480.7$, тобто втрата стійкості важеля відбудеться при $P_1 = 480.7$ МПа.

Натиснемо на кнопку  й перейдемо до наступного розрахунку - 3.Eigenvalue 2 19.28197. Результати наведені на рис. 6.50.

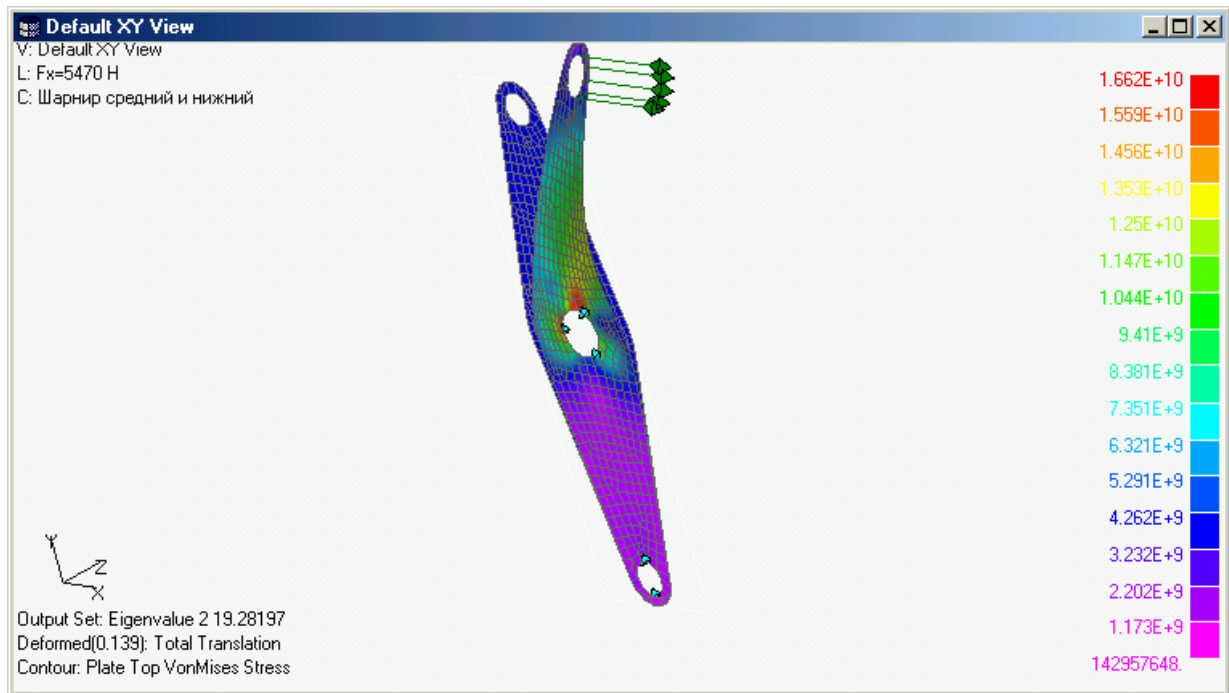


Рис. 6.50

На рис. 6.50 видно, що коефіцієнт β_2 другого критичного навантаження дорівнює $\beta_2 = 19.28$.

Друге критичне навантаження дорівнює $P_2^{kp} = \beta_2 P_1 = 19.28 \times 27.35 = 527.36$, тобто втрата стійкості за другою формою відбудеться при зусиллі $P_2 = 527.36$ МПа.

Таким чином, умови міцності й стійкості важеля виконуються.



Зменшимо товщину важеля. У головному меню виберемо пункт **Modify - Edit** (Змінити – Редагувати). З'явиться підменю (рис. 6.51), у якому наведений набір засобів модифікування моделі.

Point...	Точка
Boundary...	Межа
Coord Sys...	Система координат
Node...	Вузли
Element...	Елементи
Material...	Матеріал
Property...	Властивості
Load...	Навантаження
Constraint...	Граничні умови
Contact Segment/Surface...	Контактні сегменти/поверхні
Function...	Функції
Text...	Текст

Рис. 6.51

Виберемо пункт підменю **Property** (Властивості). З'явиться вікно, у якому за замовчуванням уведений ідентифікатор елемента **Plane** (ID = 1). Натиснемо кнопку ОК. Після цього з'явиться вікно (рис. 6.26), у якому в розділі **Property Values** (Значення параметрів), **Thicknesses** (Товщина) уведемо значення 0.012 м (12 мм) і натиснемо кнопку ОК.

У головному меню виберемо пункт **File - Analyze** і повторимо всі дії за розрахунками важеля.

За допомогою кнопок **PostProcess** (Результати)  і **Post Data** (Вихідні дані)  виведемо вікно **Select PostProcessing Data**. У списку **Output Set** з'явилася додаткова назва варіанта розрахунку – 4.MSC/NASTRAN Case 1.

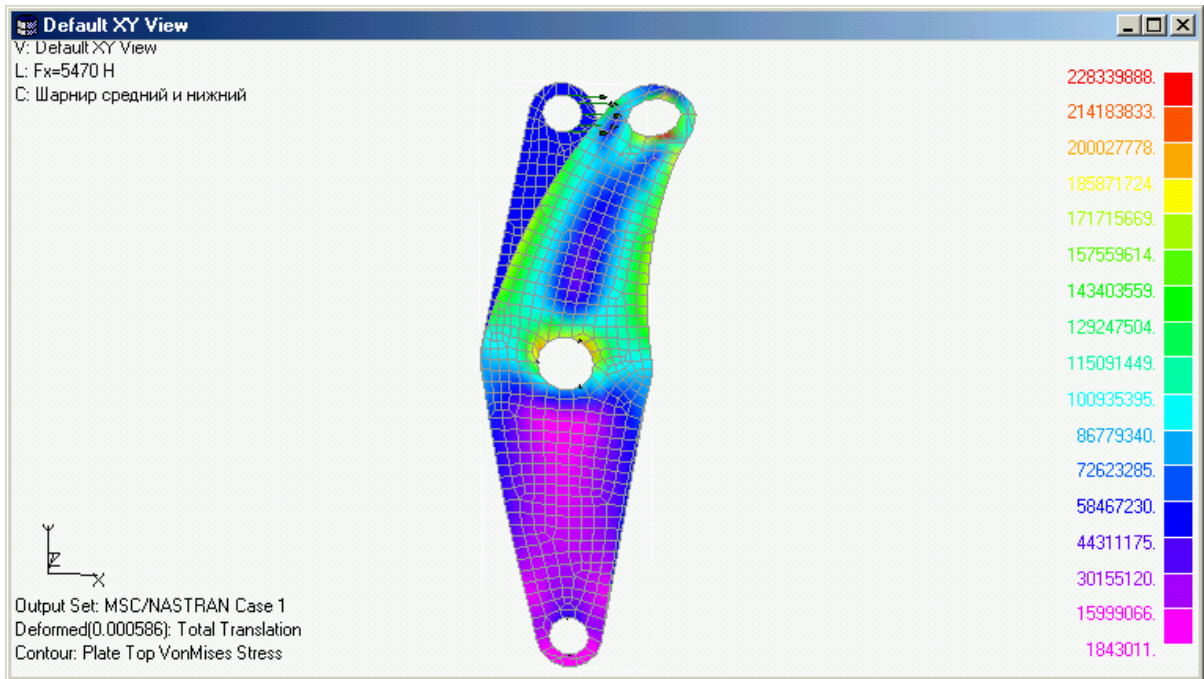
Результати розрахунку цього варіанта показані на рис. 6.52.

Максимальне значення напруг становить 228.3 МПа.

Коефіцієнт запасу міцності по межі текучості дорівнює $n_T = \frac{250}{228.3} = 1.09$, тобто при товщині пластини 12 мм напруги близькі до допустимих.

При вигині найбільш навантаженою частиною є вушко важеля поблизу точок внутрішньої циліндричної поверхні (рис. 6.52).

a)



б)

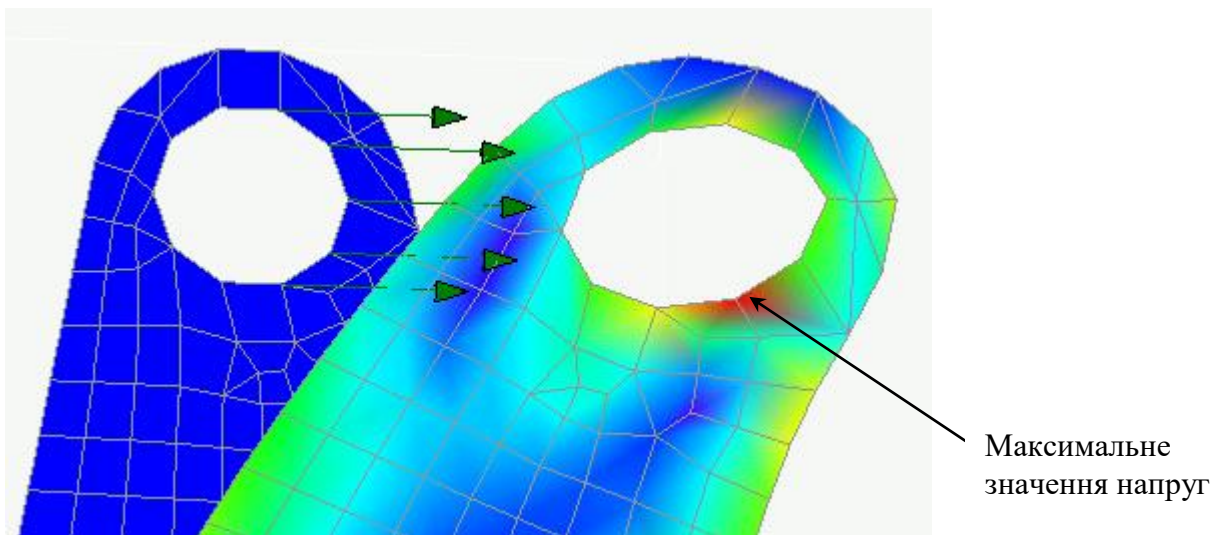


Рис. 6.52

7. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

7.1. Поняття імітаційної моделі

Людина використовує моделювання із самого дитинства. Коли маленька дівчинка грає зі своєю лялькою, вона моделює процеси, з якими їй прийдеться зіштовхнутися в подальшому житті. У такий спосіб вона пізнає навколишній світ і набуває практичних навичок роботи.

В інженерній практиці широко використовується імітаційне моделювання [17]. До нього вдаються у тих випадках, коли математичний опис досліджуваного реального процесу стає настільки складним, що не приводить до практичного результату.

У процесі імітаційного моделювання проводиться штучний експеримент, при якому замість проведення натурального експерименту з реальним об'єктом проводяться дослідження на математичних моделях. Імітаційне моделювання включає процеси розроблення моделі реальної системи і постановки експерименту на цій моделі. Такий експеримент проводиться з метою з'ясування поведінки системи або оцінення різних стратегій її функціонування, які забезпечують досягнення поставленої мети.

Імітаційне моделювання проводиться на ЕОМ. У процесі моделювання в ЕОМ вводяться математична модель, що зв'язує характеристики станів процесу з параметрами досліджуваної системи, вхідна інформація і початкові умови. Після проведення серії експериментів ЕОМ обчислює вихідні параметри процесу, який досліджується. Випадкові впливи на систему визначаються прийнятим теоретичним законом розподілу або на підставі експериментальних даних.

У процесі імітаційного моделювання проводяться статистичні експерименти [19]. У його основі лежить метод статистичних випробувань (метод Монте - Карло). Суть методу полягає в тім, що результати випробувань залежать від деякої випадкової величини, розподіленої за заданим законом. Тому результат кожного випробування теж є випадковим. Після проведення серії випробувань можна одержати безліч часткових значень досліджуваної характеристики (вибірку). У результаті

обробки цих статистичних даних отримають оцінку величин, що цікавлять дослідника (характеристик системи).

При моделюванні випадкових факторів використовуються випадкові числа, рівномірно розподілені на інтервалі $[0,1]$. Формування випадкових чисел проводиться за допомогою перетворювачів на основі сигналів фізичних генераторів, що використовують природні джерела випадкового шуму (електронні й напівпровідникові пристрої, радіоактивний розпад і т.п.). Випадкові числа можна генерувати на ЕОМ. У цьому випадку їх називають псевдовипадковими числами. Багато комп'ютерних програм (Excel, Mathcad й ін.) мають функції для одержання випадкових чисел.

Випадкові числа, рівномірно розподілені на інтервалі від a_1 до b_1 , можна одержати, використовуючи таку формулу:

$$u_i = a_1 + r_i (b_1 - a_1),$$

де r_i - випадкове число, рівномірно розподілене на інтервалі від 0 до 1.

Розглянемо простий приклад, що ілюструє застосування методу Монте-Карло.

Нехай потрібно визначити площу еліпса із центром у точці $(2, 1)$, що має півосі $a = 2$, $b = 1$.

Впишемо еліпс у прямокутник, вершини якого мають координати $(0,0)$, $(4, 0)$, $(4,2)$, $(0,2)$. Координати будь-якої точки, розташованої усередині прямокутника або на його границі, повинні задовольняти нерівності

$$0 \leq x \leq 4 \text{ й } 0 \leq y \leq 2.$$

Рівняння еліпса, вписаного в прямокутник, має вигляд

$$\frac{(x-2)^2}{a^2} + \frac{(y-1)^2}{b^2} = 1,$$

де a й b - півосі еліпса: $a = 2$, $b = 1$.

Розрахунок будемо проводити в такій послідовності:

1) за допомогою генератора випадкових чисел отримаємо випадкове число, рівномірно розподілене на інтервалі $[0,1]$;

2) визначимо випадкове число x , рівномірно розподілене на інтервалі

$$0 \leq x \leq 4;$$

3) визначимо випадкове число y , рівномірно розподілене на інтервалі

$$0 \leq y \leq 2;$$

4) перевіримо, чи перебуває точка з даними координатами усередині еліпса або на його межі;

5) проведемо деяку кількість випробувань і підрахуємо число точок, що потрапили усередину еліпса або на його межі.

Якщо провести n випробувань й m з n точок потрапить усередину еліпса і на його межі, то оцінку площі еліпса можна отримати за формулою

$$S_{\text{э}} = S_{\text{пр}} m/n.$$

Розрахунки виконаємо за допомогою табличного процесора Excel.

В комірку A3 введемо порядковий номер випробування. Генерувати випадкові числа, рівномірно розподілені на інтервалі $[0,1]$ будемо за допомогою функції СЛЧИС(). Для цього введемо в комірку B3 формулу

$$=\text{СЛЧИС}().$$

Введемо в комірку C3 формулу

$$=4*B3,$$

а в комірку D3 формулу

$$=2*B3.$$

В комірку E3 уведемо формулу

$$=\text{ЕСЛИ}(((33-2)^2)/4-(D3-1)^2 \leq 1; 1; 0).$$

Тут буде перевірятися, чи попадає точка усередину еліпса або на його межі. Якщо попадає, то в комірку E3 буде введена 1, якщо не попадає - 0.

Проведемо 200 випробувань. Виділимо осередок B3:E3. Після цього встановимо покажчик „миші” на маркер заповнення виділеного діапазону і протягнемо його вниз доти, поки не отримаємо результати 200 випробувань.

Діапазон комірок A3:A202 можна легко заповнити й використати для контролю числа точок. Введемо в комірку A3 число 1, а в комірку A4 формулу

$$=A3+1$$

і протягнемо вниз.

Фрагмент таблиці показаний на рис. 7.1

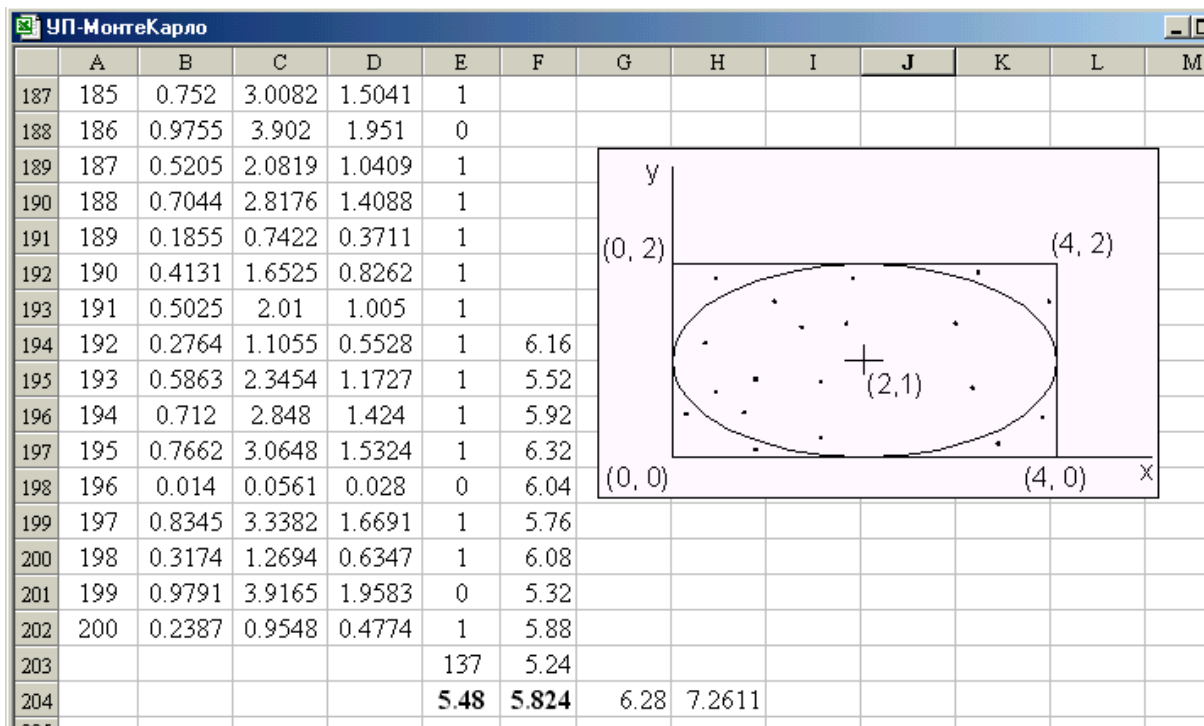


Рис. 7.1

В комірку E203 введемо формулу

$$=СУМ(E3:E202)$$

і підрахуємо число точок, що потрапили усередину еліпса.

Введемо в комірку E204 формулу

$$=(2*4)*E203/A202$$

і отримаємо оцінку площі еліпса для даної серії випробувань.

Запишемо отриманий результат в осередок F194.

Натиснемо клавішу F9. У результаті буде згенерована нова послідовність випадкових чисел (виконаний наступний прогін) і отримана нова оцінка площі еліпса.

Результати 10 прогонів наведені в осередках F194:F203. Середнє значення площі дорівнює 5.776. Точне значення площі, знайдене за формулою

$$S = \pi ab,$$

дорівнює 6.28.

Похибка становить близько 8 %.

Зі збільшенням числа згенерованих точок оцінка площі еліпса наближається до точного значення.

7.2. Системи масового обслуговування

Із системами масового обслуговування доводиться мати справу у всіх сферах діяльності людини. У побуті системами масового обслуговування є магазини, квиткові каси, телефонні станції, перукарні, майстерні й т.д. У сфері виробництва системами масового обслуговування є ділянки підприємств по виготовленню, ремонту й технічному утриманню різних машин.

Теорія систем масового обслуговування була розроблена датським математиком А. К. Ерлангом, що досліджував запити, що надходять на телефонну станцію.

У будь-якій системі масового обслуговування повинні бути визначені:

- 1) вхідний потік вимог або заявок на обслуговування;
- 2) дисципліна постановки в чергу;
- 3) правило, за яким здійснюється обслуговування;
- 4) вихідний потік вимог (заявок);
- 5) режими роботи.

Система масового обслуговування являє собою сукупність послідовно зв'язаних між собою вхідних вимог на обслуговування, накопичувачів, черг, каналів обслуговування й вихідних потоків вимог після обслуговування. Система масового обслуговування складається із тієї, що обслуговує, і тієї, що користується послугами. Система, що користується послугами, включає сукупність джерел вимог і вхідного потоку вимог.

Вимогою називають кожен окремий запит на виконання якої-небудь роботи.

Джерелом вимог може бути будь-який об'єкт (людина, машина й т.д.), який може направити в обслуговуючу систему тільки одну вимогу.

Обслуговуюча система складається з накопичувача й механізму обслуговування. Вимоги можуть відразу надходити в механізм обслуговування, якщо він вільний, або в накопичувач, де очікують початку обслуговування. Механізм обслуговування може складатися з декількох каналів обслуговування. Обслуговуючий канал здатний задовольнити тільки одну вимогу.

Вхідний потік характеризується моментами часу надходження вимог у систему й кількістю вимог, які надходять одночасно. Закон надходження вимог може бути детермінованим або імовірнісним. Детерміновані закони надходження вимог використовуються для організації роботи поточно-конвеєрних ліній. Наприклад, вимоги на обслуговування можуть надходити одна за іншою через 7 хв. При імовірнісному законі вимоги можуть надходити з рівною ймовірністю протягом деякого проміжку часу. Наприклад, одна вимога може надходити в інтервалі 7 ± 2 хв.

Потік називається найпростішим, якщо він має такі три властивості:

1) стаціонарність. Це означає, що ймовірність надходження k вимог у деякому проміжку часу дорівнює ймовірності надходження k вимог у будь-якому іншому проміжку часу такої ж тривалості. З нього виходить незмінність імовірнісного режиму потоку;

2) відсутність післядії. Ця властивість потоку означає, що ймовірність надходження k вимог у систему після довільного моменту часу не залежить від того, скільки надійшло вимог до цього моменту часу. Вона виражає взаємну незалежність надходження того або іншого числа вимог на обслуговування в проміжки часу, що не перетинаються;

3) ординарність. Потік має цю властивість, якщо ймовірність надходження більше однієї вимоги за малий проміжок часу є величина більш високого порядку малості, ніж ймовірність надходження тільки однієї вимоги. Іншими словами, практично неможливо одночасне надходження відразу двох і більше вимог.

Потік вимог, що володіє такими властивостями (стаціонарність, відсутність післядії, ординарність), називається найпростішим або пуассонівським.

Якщо вхідний потік вимог найпростіший, то число вимог k для будь-якого інтервалу часу розподілено за законом Пуассона

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k \geq 0, t \geq 0,$$

де λ - інтенсивність потоку вимог (число вимог за одиницю часу).

Для заданого числа $k = 0, 1, 2, \dots$ надходжень числа вимог можна визначити ймовірність того, що в систему надійде вимог не більше заданого числа K

$$P(k \leq K) = \sum_{k=0}^K P_k(t).$$

Якщо потік вимог найпростіший, то розподіл проміжків часу між моментами надходження вимог на обслуговування відповідає показниковому закону з функцією щільності розподілу ймовірностей

$$f(t) = \lambda e^{-\lambda t}.$$

Дисципліна постановки в чергу і вибір з неї характеризує порядок обслуговування вимог, що надходять на вхід системи. Якщо всі обслуговуючі пристрої зайняті, виникає черга. Найпростіша дисципліна обслуговування використовує чергу типу: першим прийшов - першим обслужений (ПЕРППО). Черга може бути організована за правилом: останнім прийшов - першим обслужений (ПОСППО). Дисципліна черги може визначатися правилом випадкового відбору вимог (СОТ).

На чергу можуть накладатися обмеження за довжиною черги або за часом перебування в ній.

Правила обслуговування встановлюють тривалість обслуговування (розподіл часу обслуговування), кількість вимог, які обслуговуються одночасно, і дисципліну обслуговування. Час обслуговування може бути детермінованим або імовірнісним.

Якщо обслуговування організоване за допомогою одного пристрою (каналу обслуговування), то така система називається одноканальною. При великій кількості однотипних каналів систему називають багатоканальною.

Ланцюжок послідовно зв'язаних пристроїв називають багатофазною системою масового обслуговування. У такій системі вимога послідовно проходить всі фази, перед тим, як покинути систему.

Дисципліна обслуговування визначає правила, при яких система припиняє обслуговування вимог, а також правило вибору для обслуговування наступної вимоги. Порядок обслуговування може не мати пріоритетів і визначатися дисципліною вибору із черги.

7.3. Система моделювання GPSS World

Прагнення зменшити трудомісткість програмної реалізації імітаційних моделей й експериментів з моделями привело до розроблення спеціальних мов моделювання. Однією з мов моделювання, що полегшує процес написання імітаційних програм, є мова GPSS.

Система моделювання GPSS World (GPSSW, General Purpose System Simulation World - Світова загальноцільова система моделювання), розроблена фірмою Minuteman Software, призначена для операційної системи Windows.

Модель складної системи на GPSS може бути наведена у вигляді сукупності елементів і логічних правил їхньої взаємодії. Передбачається, що система, яка моделюється, має обмежений набір елементів, які називаються об'єктами. Логічні правила взаємодії об'єктів можуть бути описані обмеженим набором стандартних операцій.

У мові використовуються різні типи об'єктів: транзакти, блоки, черги й ін.

Транзакти є динамічними об'єктами. Вони створюються в певній точці моделі, просуваються через блоки, а потім знищуються. Конкретний зміст транзакта визначається розроблювачем моделі. Наприклад, при моделюванні процесу роботи колісного цеху депо транзактом є колісна пара; при моделюванні роботи магазину транзактом є покупець.

Система GPSS World має 53 типи блоків. Блоки створюють і знищують транзакти, організовують тимчасову затримку, синхронізують рух двох або декількох транзактів, змінюють параметри й пріоритети, описують одноканальні пристрої (технічні засоби) і багатоканальні пристрої (пам'яті), зберігають необхідні значення для подальшого використання, забезпечують збір статистичної інформації й т.д.

Для опису імітаційної моделі мовою GPSSW можливо подати її у вигляді схеми, що називають блок - діаграмою. На діаграмі кожному блоку відповідає певне графічне зображення, причому види зображень блоків стандартизовані. Блоки з'єднуються між собою лініями. Приклад блок - діаграми показаний на рис. 7.2.

Кожен блок виконує певну функцію. Фактично блок являє собою підпрограму, написану мовою Си.

На початку моделювання в моделі немає жодного транзакта. Через блок **GENERATE** транзакти входять у модель, послідовно переміщується від одного блоку до іншого й наприкінці залишають модель. Якщо транзакт входить у блок, функцією якого є, наприклад, затримка транзакта на якийсь час, то транзакт

залишитися на місці, і почнеться переміщення в моделі іншого транзакта.

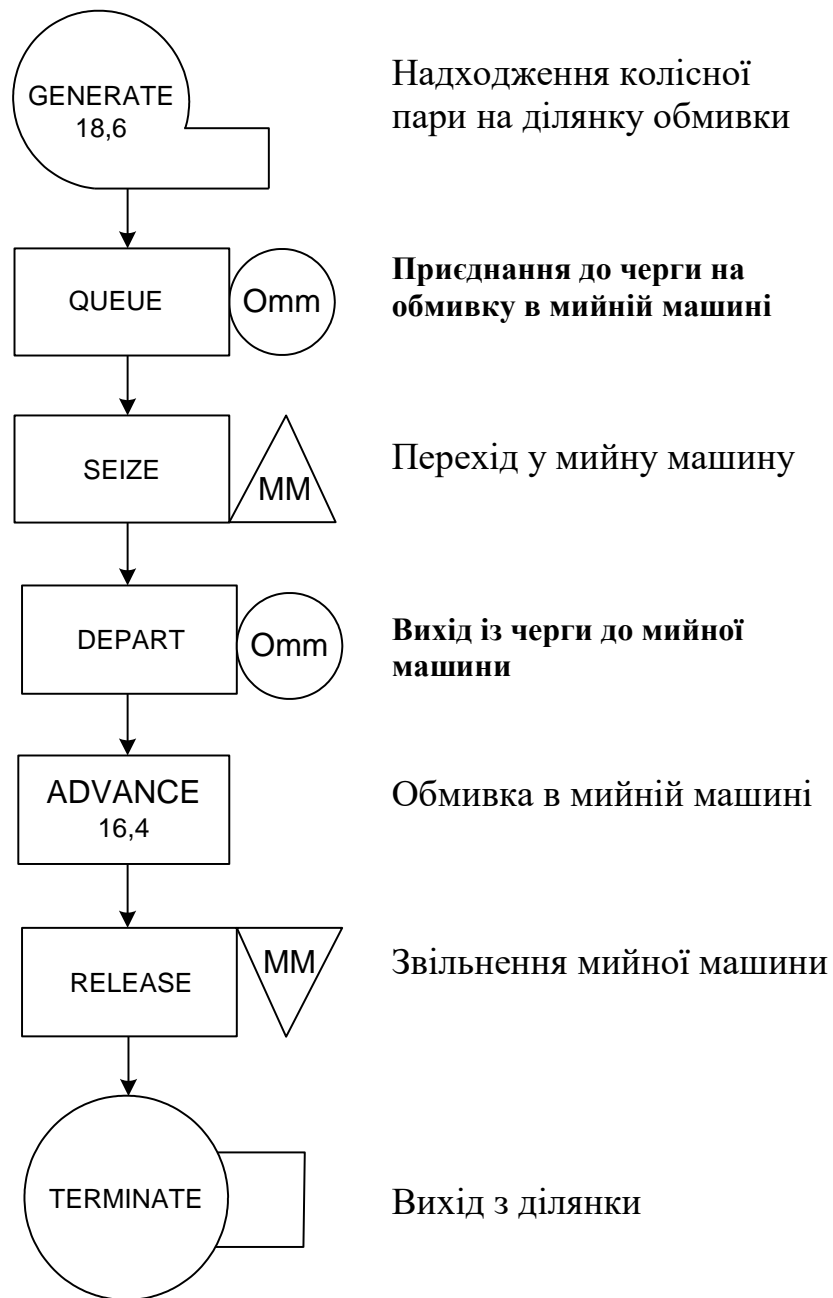


Рис. 7.2

В моделі існує велике число транзактів, однак у певний момент часу рухається тільки один транзакт.

Після розроблення блок - діаграми можна приступати до написання тексту програми на GPSSW.

7.4. Оператори мови GPSS World

Інформація про об'єкти GPSSW записується у вигляді послідовності операторів.

Оператор записується у вигляді одного текстового рядка довжиною не більше 250 символів.

В операторі виділяють окремі частини, які називаються полями. У загальному випадку оператор складається з таких полів:

номер рядка (необов'язково) | мітка (необов'язково) | дієслово (обов'язково) | операнди (залежить від дієслова) | коментарі.

Під дієсловом тут розуміються ключові слова GPSS, тобто або блок, або команда.

Коментарі – необов'язкове поле. Коментарі відокремлюються від поля операндів символом «;».

Нижче наводиться короткий опис деяких операторів GPSSW.

GENERATE (ГЕНЕРУВАТИ) – це блок, через який транзакт входить у модель. У моделі може бути кілька блоків **GENERATE**. Блок **GENERATE** має такий формат:

GENERATE [A], [B], [C], [D], [E]

Операнди мають такі значення:

A - середнє значення інтервалу надходження;

B - половина поля допуску рівномірно розподіленого інтервалу;

C - зсув інтервалів;

D - кількість транзактів, що генеруються;

E - пріоритет.

Приклад: **GENERATE 5, 2** - інтервали часу між появами транзактів розподілені рівномірно в діапазоні $[5 - 2, 5 + 2]$;

GENERATE 5, 2, 10, 4, 1 – генерація транзактів, інтервали часу між появами транзактів розподілені рівномірно в діапазоні $5 - 2, 5 + 2$, перший транзакт з'явиться із затримкою в 10 одиниць

модельного часу, усього буде створено 4 транзакти, пріоритет транзактів дорівнює одиниці.

GENERATE 5, 2, 10, , 1 – те ж, але кількість транзактів не обмежується.

TERMINATE (ЗАВЕРШИТИ) – видалення транзактів з моделі.

Формат блоку:

TERMINATE [A]

A - величина зменшення спеціального лічильника, що називається лічильником завершення.

Початкове значення цього лічильника встановлюється на початку моделювання.

START (ПОЧАТИ) – керування тривалістю процесу моделювання.

Можливі два способи керування тривалістю процесу моделювання.

Перший спосіб

1. У команді **START** операнду **A** присвоюється значення заданого числа транзактів.

2. У всіх блоках **TERMINATE**, через які транзакти залишають модель, операнду **A** присвоюється значення «1».

3. У всіх інших блоках **TERMINATE** використовується значення операнда за замовчуванням (**A** = 0).

Процес моделювання закінчиться, коли через модель пройде задану кількість транзактів, наприклад 1000.

GENERATE 15,3

...

TERMINATE 1

START 1000

Другий спосіб

Припустимо, що за одиницю модельного часу прийнята 1 хв і потрібно моделювати поведінку системи протягом 8 год.

Це можна зробити в такий спосіб:

1. Ввести в модель таймер - сегмент, що складається із двох блоків:

```
GENERATE 480  
TERMINATE 1  
START 1
```

2. У всіх інших блоках **TERMINATE** використати значення операнда **A = 0**.

3. У команді **START** операнд **A** прийняти рівним одиниці.

У процесі моделювання завершення руху транзактів в інших блоках **TERMINATE** не впливає на лічильник завершення. У момент часу 480 транзакт вийде із блоку **GENERATE** і відразу перейде в блок **TERMINATE**. Лічильник завершення зменшиться на одиницю й процес моделювання завершиться.

ADVANCE (ЗАТРИМАТИ) – затримка транзакта на час, визначений змістом полів **A** і **B**.

Формат: **ADVANCE A[,B]**

Зміст значень **A** і **B** такий же, як в операторі **GENERATE**.

Приклад: **ADVANCE 10,2**

Час затримки транзакта - випадкова величина, рівномірно розподілена на інтервалі [8, 12].

SEIZE (ЗАЙНЯТИ) – заняття пристрою.

Формат: **SEIZE A**

A – ім'я або номер пристрою.

Приклад: **SEIZE Mmash** – зайняти пристрій з ім'ям **Mmash**.

RELEASE (ЗВІЛЬНИТИ) – звільнення пристрою.

Формат: **RELEASE A**.

A – ім'я або номер пристрою.

Приклад: **RELEASE Mmash**

Часто використовується послідовність операторів **SEIZE – ADVANCE – RELEASE**.

Приклад:
SEIZE Mmash
ADVANCE 10,2
RELEASE Mmash

QUEUE (СТАТИ В ЧЕРГУ) – збір статистики про очікування.

Формат: **QUEUE** A[,B].

A – ім'я черги;

B – число одиниць, на яке збільшується черга.

Якщо операнд B не використовується, то довжина черги збільшується на одиницю.

Приклад:

QUEUE Ocher_MM

При вході транзакта в блок **QUEUE** черга збільшується на одиницю.

DEPART (Покинути чергу)

Формат: **DEPART** A[,B]

A – ім'я черги;

B – число одиниць, на яке зменшується черга.

Приклад:

QUEUE Ocher_MM - стати в чергу;

SEIZE Mmash - зайняти пристрій (мийну машину);

DEPART Ocher_MM - залишити чергу;

ADVANCE 10,2 - виконання операції (обмивка);

RELEASE Mmash - звільнити пристрій (мийну машину).

TRANSFER (ПЕРЕДАТИ) – передача транзакта в будь-який інший блок.

Формат: **TRANSFER** [A], [B], [C], [D]

Операнд A задає режим блоку. Існують такі 9 режимів роботи блоку **TRANSFER**:

за замовчуванням) - безумовний;

статистичний, вибір випадковим чином одного із двох блоків;

BOTH - послідовний вибір одного із двох блоків;

ALL - послідовний вибір одного з декількох блоків;
PICK - вибір випадковим чином одного з декількох блоків;
FN - функціональний;
P - параметричний;
SBR - підпрограмний;
SIM - одночасний.
B - номер або ім'я блоку (у режимі P - ім'я параметра);
C - номер або ім'я блоку. У режимі функції (FN) або параметра - збільшення;
D - збільшення номера блоку для режиму ALL. За замовчуванням дорівнює 1.

Приклади:

TRANSFER ,MET - безумовна передача керування операторові з міткою (номером) MET;

TRANSFER BOTH, Stanok1, Stanok2 - перехід до оператора з міткою Stanok1, якщо він неможливий, то до оператора з міткою Stanok2, якщо й він неможливий, то транзакт затримується до наступного моменту модельного часу, у який повторюються зазначені спроби;

TRANSFER .4, AAA, MM - транзакт із імовірністю 0.4 переходить до оператора з міткою MM і з імовірністю 0.6 до оператора з міткою AAA;

TRANSFER PICK, ST7, ST12 - рівномірний перехід до операторів з іменами ST7, ST8,..., ST12;

TRANSFER FN, AAA, 5 - перехід до оператора, мітка якого дорівнює сумі значення функції AAA і числа 5;

TRANSFER P, 4, 41 - перехід до оператора, мітка якого дорівнює сумі значення параметра № 4 транзакта й числа 41;

TRANSFER SBR, Prc, Inst - перехід до оператора Prc, номер блоку TRANSFER записується в параметр із ім'ям Inst; якщо такого параметра немає, він створюється.

ENTER (УВІЙТИ) – заняття багатоканального пристрою.

Формат блоку: ENTER A[,B].

LEAVE (ВИЙТИ) – звільнення багатоканального пристрою.

Формат блоку: LEAVE A[,B]

Операнд А в обох блоках використовується для вказівки імені багатоканального пристрою. Операнд В задає число пристроїв, які повинні бути зайняті в блоці ENTER або звільнені в блоці LEAVE. За замовчуванням операнд В = 1.

STORAGE (СХОВИЩЕ або ПАМ'ЯТЬ) – оператор, що визначає ємність багатоканального пристрою.

Формат блоку: STORAGE А.

А - ємність накопичувача.

Приклад: Nak STORAGE 15

.....
ENTER Nak, 2

ADVANCE 22,9

LEAVE Nak, 2

ASSIGN (ПРИСВОЇТИ) – модифікувати параметр вимоги.

Формат блоку: ASSIGN А,В

А - номер параметра активного транзакта;

В - присвоює значення.

Приклад: ASSIGN 1,CAN1

7.5. Розроблення моделі в GPSS World

Розроблення моделі передбачає виконання ряду етапів:

- 1) постановка завдання;
- 2) виявлення основних особливостей завдання;
- 3) створення імітаційної моделі процесу;
- 4) подання імітаційної моделі в GPSSW;
- 5) налагодження моделі;
- 6) моделювання системи;
- 7) модифікація системи, яка моделюється (при необхідності).

Запуск GPSS World здійснюється подвійним щигликом „миші” по його значку, що розташовується на робочому столі. З'явиться вікно Notice, у якому можна натиснути на кнопку Don't Download. Вікно Notice закриється й з'явиться головне вікно

GPSS World. Виберемо (Файл - Новий). Після цього з'явиться меню, у якому виберемо Model, оскільки потрібно створити об'єкт «Модель» (рис. 7.3).

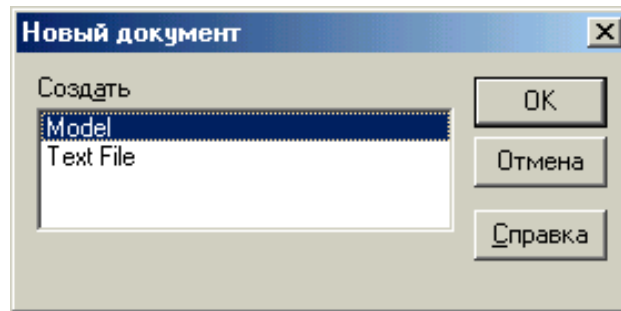


Рис. 7.3

Відкриється вікно текстового редактора (рис. 7.4).

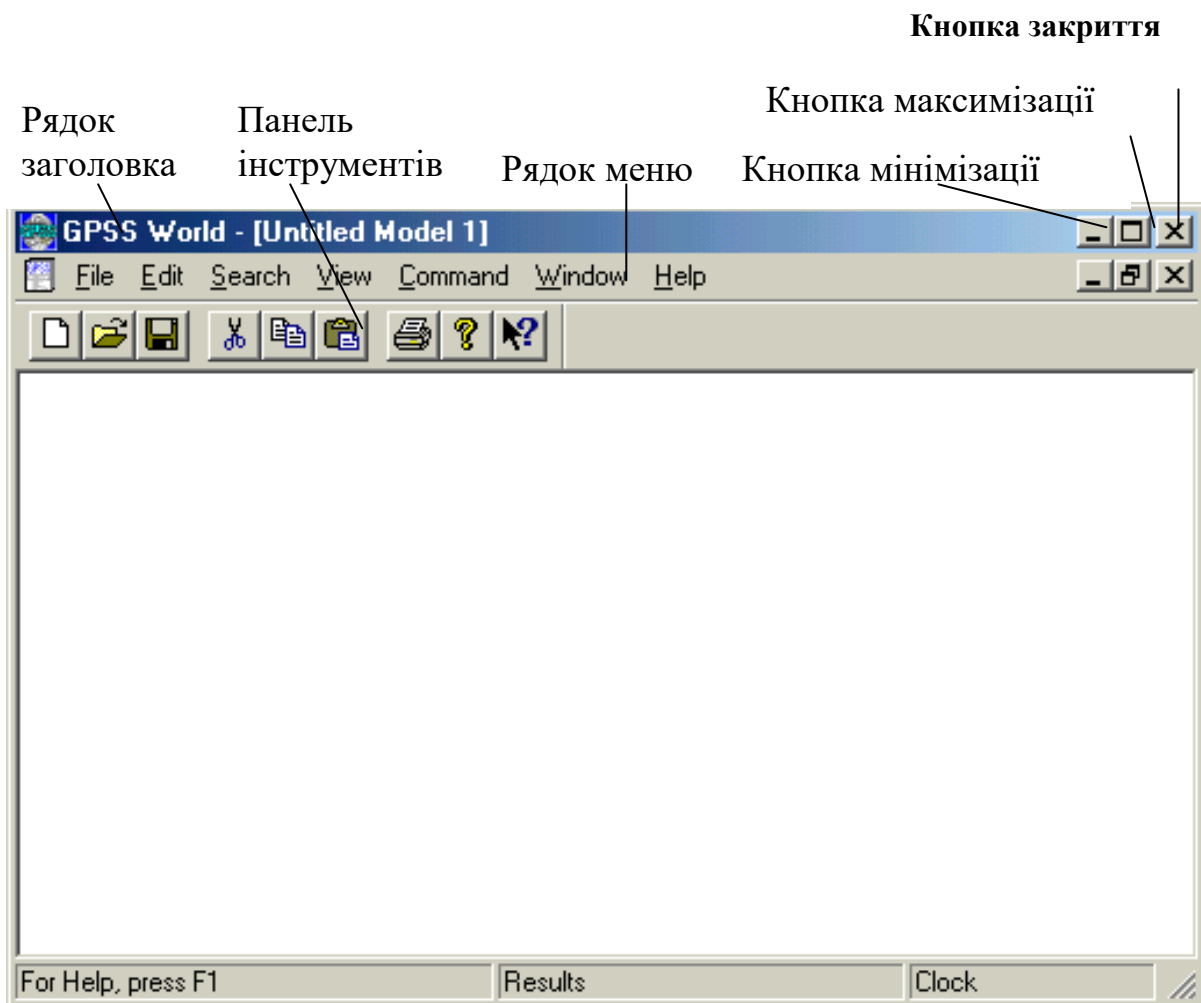


Рис. 7.4

Модель звичайно починається із заголовка. Заголовок вводиться як коментар. Коментар можна ввести після набору символу «;» або «*».

Текстовий редактор GPSS World використовує звичайні засоби форматування. Можна вибрати шрифт, установити накреслення, розмір і кольори символів. За замовчуванням використовується шрифт Courier New. Символи цього шрифту мають однакову ширину, що полегшує вирівнювання колонок тексту моделі. Це особливо важливо при виведенні стандартного звіту.

Установлення параметрів шрифту виконується в пункті головного меню Edit - Font (Виправлення - Шрифт).

Введемо у вікні текстового редактора такі коментарі:

```
*GPSSW           Mmachina.GPS
*****
*Моделювання роботи мийної машини           *
*****
```

Оператори GPSS World можна безпосередньо вводити із клавіатури. Введемо в наступному рядку ім'я t_obm. Для переходу до введення QTABLE варто скористатися клавішею [Tab]. Після введення оператора й коментарю буде виглядати так:

```
t_obm    QTABLE    Ocher_mm,0,2,32 ;Генерація потоку кол.пара
```

При введенні програми можна використовувати стандартні прийоми редагування: вирізати, копіювати, вставити.

Можна також скористатися вікном Insert GPSS Blocks (Вставити блок) і вікнами створення блоків.

Виберемо Edit - Insert GPSS Blocks... (Виправлення - Вставити блок GPSS). У вікні (рис. 7.5) виберемо GENERATE і натиснемо лівою клавішею „миші”.

Insert GPSS Block into Model Object		
ADOPT	ASSEMBLE	ALTER
ADVANCE	CLOSE	COUNT
ASSIGN	GATE	DISPLACE
BUFFER	JOIN	EXAMINE
DEPART	LINK	EXECUTE
ENTER	LOGIC	FAVAIL
GENERATE	LOOP	FUNAVAIL
LEAVE	MATCH	GATHER
MARK	OPEN	INDEX
MSAVEVALUE	PREEMPT	INTEGRATION
PLUS	PRIORITY	SAVAIL
QUEUE	READ	SCAN
RELEASE	REMOVE	SELECT
SAVEVALUE	RETURN	SUNAVAIL
SEIZE	SEEK	TABULATE
SPLIT	TEST	TRACE
TERMINATE	UNLINK	UNTRACE
TRANSFER	WRITE	

Рис. 7.5

З'явиться вікно створення блоку GENERATE (рис. 7.6) з курсором у вікні операнда А. Уведемо 15, натиснемо клавішу Tab і введемо 2 у вікні операнда В. Далі послідовним натисканням клавіші Tab або „мишею” переведемо курсор у вікно Comment і введемо коментар: Генерація потоку кол. пар. Натиснемо кнопку ОК. У моделі з'явиться оператор

GENERATE 15,2 ; Генерація потоку кол. пара

Вікно Insert GPSS Block into Model Object (рис. 7.5) знову стане активним і можна вводити наступний оператор.

Таким же чином можна ввести інші рядки програми. Після цього текст програми буде виглядати так, як показано на рис. 7.7.

Вибір у головному меню пункту Edit дозволяє внести зміни в текст програми. Для скасування неправильних дій можна скористатися командою Undo (Скасувати) або натиснути комбінацію клавіш Ctrl+Z. Команда Delete Line (Видалити рядок) видаляє виділений рядок. Команда Insert Line (Вставити рядок) поміщає новий рядок нижче рядка, на якому розташований курсор.

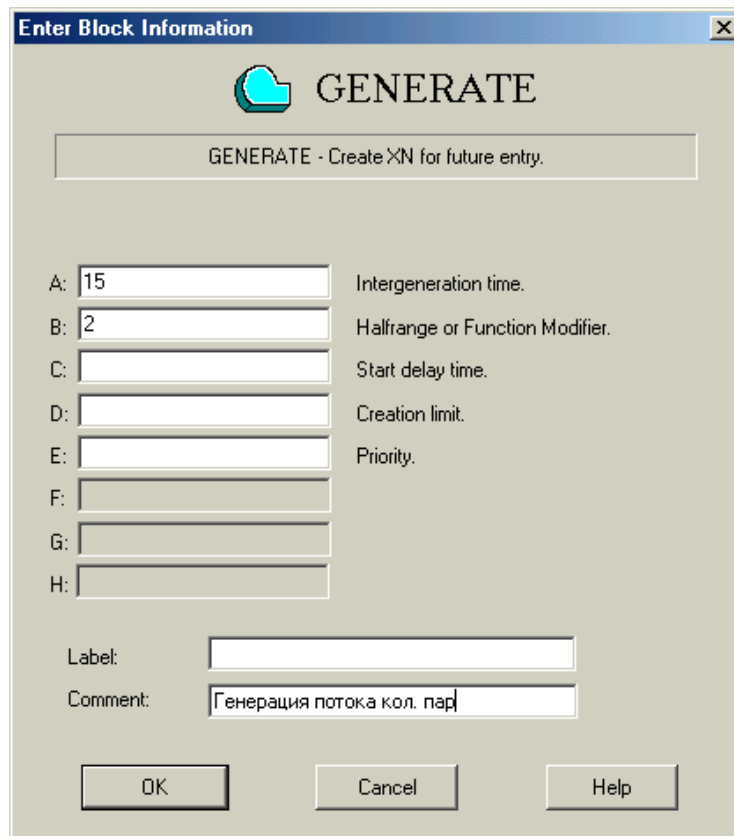


Рис. 7.6

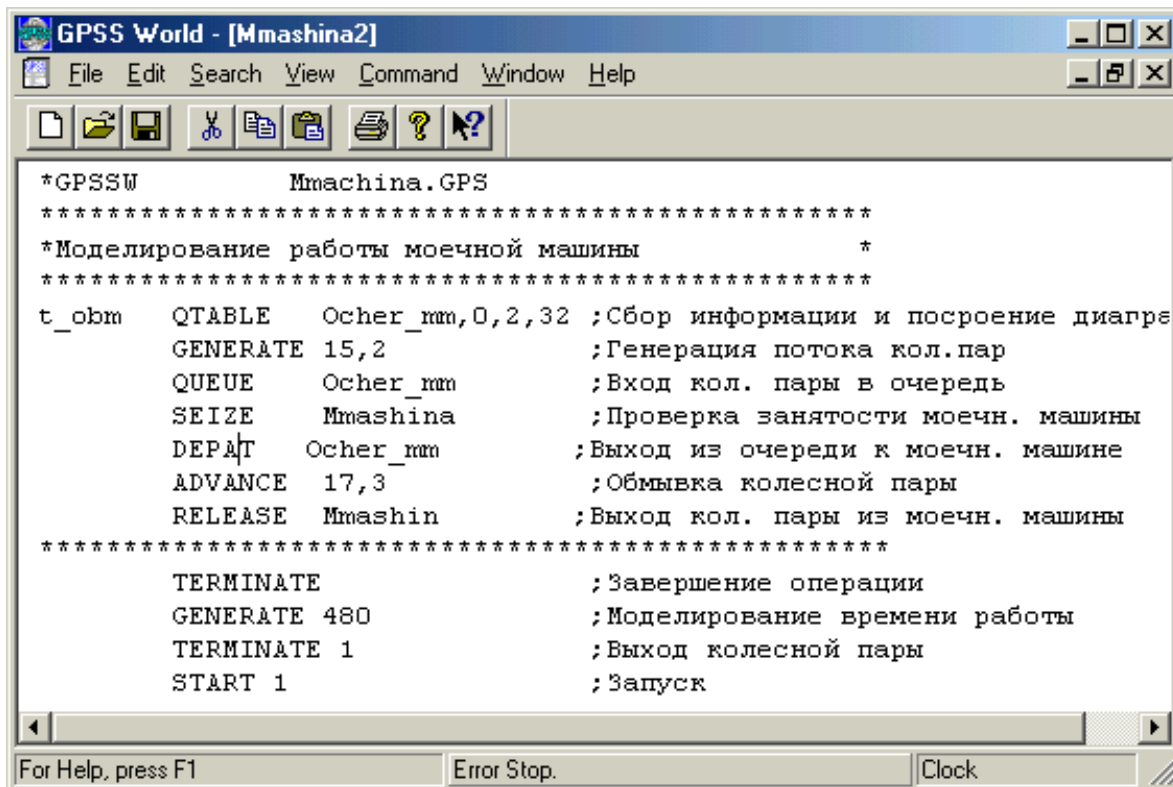


Рис. 7.7

Для редагування можливо також скористатися пунктом головного меню Search (Пошук), у якому вибрати пункт Find/Replace (Знайти/Замінити). У діалоговому вікні, що з'явилося, потрібно ввести новий текст, який повинен замінити видалений фрагмент тексту.

Виконаємо трансляцію вихідної моделі. Виберемо в головному меню пункт Command (Команда) і клацнемо по пункту Create Simulation (Створити процес моделювання). Буде дана команда на виклик транслятора GPSS, що перевірить модуль на наявність синтаксичних помилок. У більшості випадків у тексті програм є помилки.

Припустимо, що й у нашому випадку є синтаксичні помилки. При введенні оператора DEPART була пропущена буква R і набрана DEPAT. Трансляція буде перервана й на екрані з'явиться вікно JOURNAL (Журнал). У вікні JOURNAL (рис. 7.8) міститься інформація про дату й час початку трансляції. Тут же зазначено, що помилка перебуває в рядку 9, в 16 колонці. Далі наводиться рядок програми, що містить помилку, й повідомлення про те, що трансляція моделі перервана.

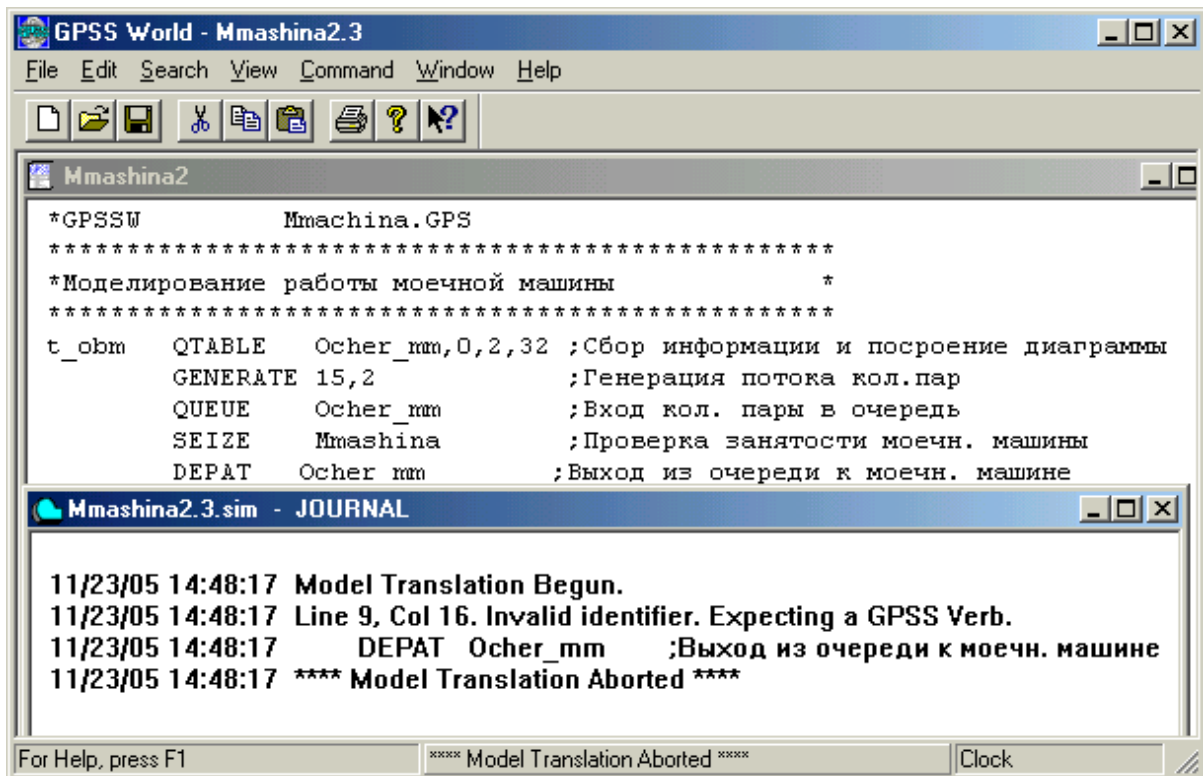


Рис. 7.8

Після виправлення синтаксичних помилок повторно виконується трансляція. Її можна виконати, як й у попередньому випадку, командою **Command - Create Simulation** (Команда - Створити процес моделювання). Можна також виконати команду **Command - Retranslate** (Команда - Повторна трансляція). У цьому випадку може відбутися зупинка внаслідок помилкового виконання і у вікні з'явиться повідомлення (рис. 7.9).

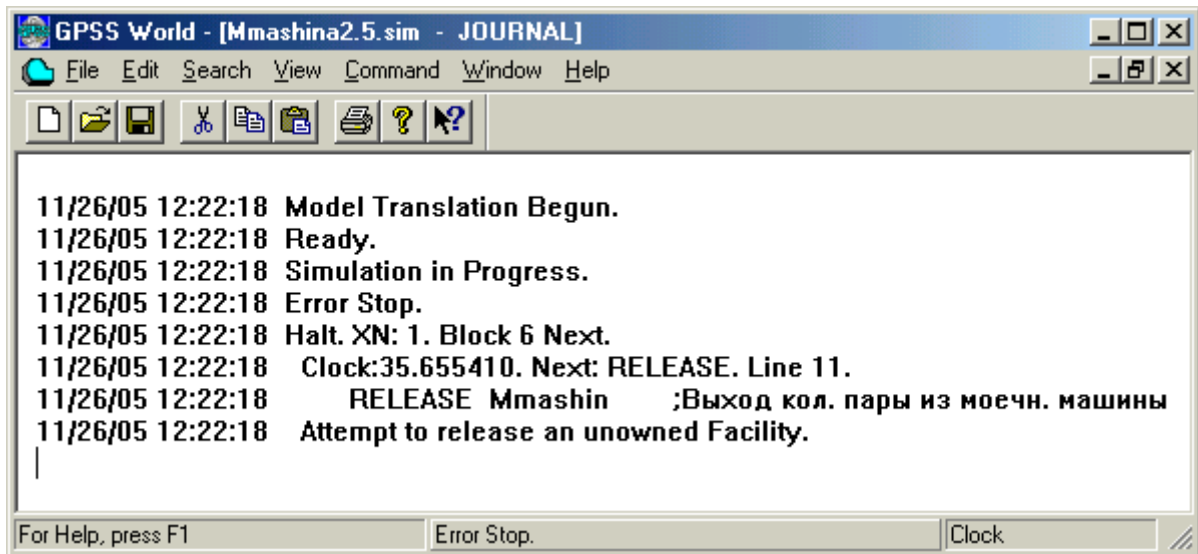


Рис. 7.9

В останньому рядку повідомлення зазначена причина зупинки: **Attempt to release an unowned Facility** (Спроба звільнити незайнятий пристрій). Інакше кажучи, транзакт намагається звільнити пристрій **Mmashin**, хоча такий пристрій зайнятий не був. У блоці **SEIZE** був зайнятий пристрій **Mmashina**. Для переходу до наступного рядка, що містить помилку виконання, необхідно скористатися командою **Search - Go To Line...** (Пошук - Перейти до рядка...). Номер рядка, що містить помилку, зазначений у повідомленні про помилку виконання (див. рис. 7.9, Line 11).

Після виправлення всіх помилок знову виконаємо трансляцію моделі. З'явиться вікно **REPORT** (Звіт), показане на рис. 7.10.

У рядку заголовка стандартного звіту вміщається ім'я файлу моделі, що створив звіт, а також дата й час прогону моделі.

Наступний рядок містить загальну інформацію про результати роботи моделі:

- START TIME (Початковий час) - 0.000;
- END TIME (Час закінчення) - 480.000;
- BLOCKS (Число блоків) - 9;
- FACILITIES (Число каналів обслуговування) - 1;
- STORAGES (Число накопичувачів) - 0.

У наступному рядку перераховуються блоки моделі й кількість входів у них вимог (колісних пар):

- ENTRIES (Число входів) - 28;
- UTIL. (Коефіцієнт використання) - 0.966;
- AVE.TIME (Середній час обслуговування) - 16.558;
- AVAIL (Доступність - стан пристрою) - 1;
- OWNER (Можливе число входів) - 29;
- PEND (Кількість транзактів, що очікує виконання з перериванням) - 0;
- INTER (Кількість транзактів, перерваних на даний момент) - 0;
- RETRY (Повтор) - 0;
- DELAY (Кількість транзактів, що очікує заняття пристрою) - 4.

В останньому рядку зазначені результати моделювання черги:

- MAX (Максимальний вміст черги) - 5;
- CONT (Поточний вміст черги) - 4;
- ENTRY (Число входів у чергу) - 32;
- ENTRY(0) (Число входів у чергу з нульовим часом очікування)- 1;
- AVE.CONT. (Середнє число входів) - 1.886;
- AVE.TIME (Середній час перебування в черзі) - 28.297;
- AVE.(-0) (Середній час перебування в черзі без обліку нульових входів у чергу) - 28.210;
- RETRY (Відмовлено) - 0.

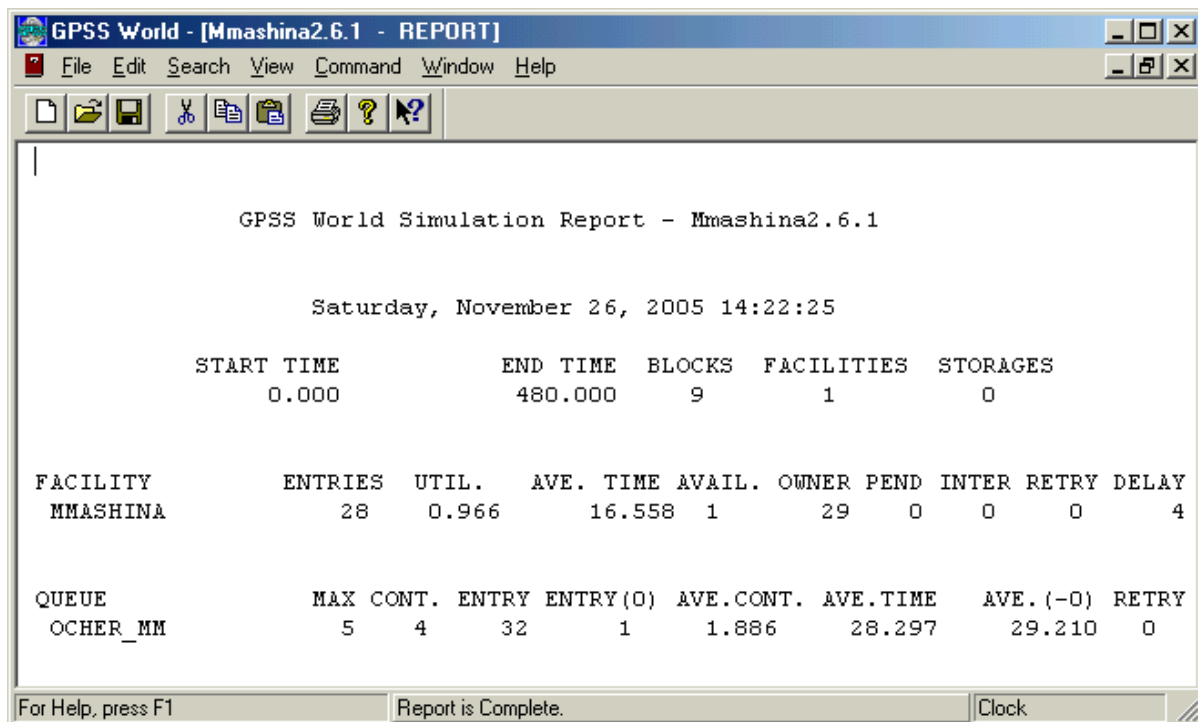


Рис. 7.10

7.6. Моделювання роботи ділянки цеху

Імітаційне моделювання широко застосовується для аналізу виробничих систем [8, 17].

Розглянемо задачу моделювання виробничої ділянки цеху [17].

На ділянці цеху встановлено три види встаткування, на яких обслуговуються два потоки виробів. На першому потоці вироби надходять на обробку з інтервалом часу 38 ± 4 хв, на другому потоці - 26 ± 5 хв. Закон розподілу інтервалів часу між надходженнями виробів кожного виду рівномірний.

Для першого потоку час виготовлення виробів на першому виді встаткування становить 18 ± 2 хв, на другому - 30 ± 4 хв, на третьому - 21 ± 3 хв. Для другого потоку час виготовлення виробів на першому виді встаткування становить 19 ± 4 хв, на другому виді встаткування - 25 ± 5 хв, на третьому виді встаткування - 28 ± 5 хв. Тривалість виготовлення виробів на всіх видах устаткування характеризується рівномірним законом розподілу.

Завдання полягає в такому:

- промоделювати роботу ділянки цеху протягом робочого дня (8 год) при двозмінній роботі;
- визначити середнє завантаження кожного виду встаткування, середній час обробки виробів кожного типу, середню довжину черг перед кожним видом устаткування;
- визначити розміри площ, необхідних для розміщення виробів на ділянці цеху.

За одиницю виміу часу приймемо хвилину. Час моделювання роботи цеху становить $8 \times 60 \times 2 = 960$ хв.

Побудову імітаційної моделі почнемо зі створення заголовка, який запишемо в такому вигляді:

```
*GPSSW                                PR_UCHASTOK.GPS
*****
*   Моделювання роботи ділянки цеху   *
*****
```

Розіб'ємо модель на декілька секторів.

У першому секторі будемо моделювати перший потік виробів.

Моделювання потоку виробів будемо виконувати за допомогою оператора GENERATE (Генерувати). Це блок, через який виріб входить у модель. У нашому прикладі він буде виглядати так:

```
GENERATE    38,4
```

Операнди: $A = 38$, $B = 4$. Операнд A вказує середній інтервал часу між надходженням на ділянку двох виробів (вимог, транзактів) , які йдуть один за другим. У нашому випадку він становить 38 хв. Операнд B задає відхилення часу надходження виробів від середнього, яке в нашому прикладі дорівнює 4. При рівномірному законі розподілу буде випадковим чином обрано одне з дев'яти різних значень: 34, 35, 36, 37, 38, 39, 40, 41, 42.

Після входу на ділянку виріб спочатку стає в чергу до встаткування першого виду. Це можна промодельовати оператором QUEUE (Черга), що разом з оператором DEPART (Вийти) збирає статистичну інформацію про роботу черги, яка моделюється.

У нашому прикладі оператор виглядає таким чином:

```
QUEUE      OCHER1
```

При вході транзакта в блок QUEUE (Черга) виконуються такі дії:

- 1) лічильник входів для даної черги збільшується на В - другий операнд в операторі QUEUE; у нашому випадку на $V = 1$;
- 2) довжина черги (лічильник поточного вмісту) для даної черги збільшується на В;
- 3) значення поточної довжини черги зберігається в стандартному числовому атрибуті Q\$< ім'я черги>;
- 4) транзакт приєднується до черги із запам'ятовуванням поточного модельного часу.

Виріб може вийти із черги тільки тоді, коли звільниться встаткування першого виду (канал обслуговування). Перевірити зайнятість каналу обслуговування можна за допомогою оператора SEIZE. Це може виглядати так:

```
SEIZE      OBOR1
```

В полі операнда А вказано символічне ім'я каналу обслуговування.

Вихід виробу із черги до встаткування першого виду фіксується оператором DEPART. У полі операнда В вказується назва черги:

```
DEPART     OCHER1
```


Далі повинний бути промодельований час обробки виробу на встаткуванні першого виду, що становить 18 ± 2 хв. Моделювання цього процесу здійснюється оператором ADVANCE (Затримати), що у нашому прикладі виглядає так:

```
ADVANCE      18,2
```

Після обробки на встаткуванні першого виду виріб направляється до встаткування другого виду. Але перед цим у систему повинне надійти повідомлення про звільнення каналу обслуговування. Це можна зробити за допомогою оператора RELEASE у такий спосіб:

```
RELEASE      OBOR1
```

Необхідно зазначити, що оператори QUEUE й DEPART створюють пару взаємозалежних блоків, які повинні мати однакове ім'я черги. Це ж відноситься до операторів SEIZE й RELEASE, які в нашому прикладі повинні мати однакову назву встаткування.

Потім виріб надходить до другого виду встаткування. Тут процес може бути промодельований так:

```
QUEUE        OCHER2
SEIZE        OBOR2
DEPART       OCHER2
ADVANCE      30,4
RELEASE      OBOR2
```

Далі виріб надходить в обробку на третій вид устаткування. Це можна промодельувати аналогічно:

```
QUEUE        OCHER3
SEIZE        OBOR3
DEPART       OCHER3
ADVANCE      21,3
RELEASE      OBOR3
```

Після цього виріб виходить із ділянки цеху. Це можливо промодельувати за допомогою оператора TERMINATE (Завершити).

Тепер можна перейти до моделювання другого потоку виробів. Це будемо робити в другому секторі програми.

Сформуємо другий потік за допомогою оператора GENERATE, що у нашому завданні запишемо так:

```
GENERATE    28,5
```

У полі операнда A вказується середній час надходження виробів другого потоку.

Для збору статистичної інформації введемо пару операторів: спочатку QUEUE, а потім через декілька операторів RELEASE, які використовуються для збору статистичної інформації.

Якщо перший вид устаткування вільний, то виріб може покинути чергу, використовуючи оператор DEPART:

```
DEPART     OCHER1
```

Моделювання обробки виробів на першому виді встаткування виконується оператором ADVANCE:

```
ADVANCE    19,4
```

Після обробки виріб другого потоку звільняє перший вид устаткування. Це можна промоделювати так:

```
RELEASE    OBOR1
```

Оператори SEIZE, ADVANCE, RELEASE створюють послідовність

```
SEIZE      OBOR1
```

```
ADVANCE    19,4
```

```
RELEASE    OBOR1
```

Транзакт, рухаючись по цьому ланцюжку блоків, займе пристрій з ім'ям OBOR1, затримається там на 19 ± 4 хв і потім замінить цей пристрій. Після того, як транзакт увійде в блок і відповідно цьому блоку підпрограма закінчиться, інтерпретатор спробує перемістити транзакт у наступний блок моделі й наступний транзакт може вже використати пристрій OBOR1.

Потім виріб надходить на обробку на другий вид устаткування. Відповідну послідовність операторів можна записати в такому вигляді:

```

QUEUE      OCHER2
SEIZE      OBOR2
DEPART     OCHER2
ADVANCE    25,5
RELEASE    OBOR2

```

Аналогічно можна промодельювати обробку виробів на встаткуванні третього виду:

```

QUEUE      OCHER3
SEIZE      OBOR3
DEPART     OCHER3
ADVANCE    28,5
RELEASE    OBOR3

```

На цьому обробка виробу на всіх трьох видах устаткування закінчується, і виріб другого потоку залишає ділянку. Це моделюється за допомогою оператора TERMINATE.

Третій сектор програми містить оператори, за допомогою яких моделюється час роботи систем:

```

GENERATE    960
TERMINATE   1
START       1

```

Остаточно програма має такий вигляд:

```

*GPSSW                PR_UCHASTOK.GPS
*****
*   Моделювання роботи ділянки цеху   *
*****
GENERATE    38,4      ;Генерація першого потоку виробів
QUEUE       OCHER1   ;Вхід у чергу до встаткування першого виду
SEIZE       OBOR1   ;Визначення заняття встаткування першого виду
DEPART      OCHER1   ;Вихід із черги до встаткування першого виду
ADVANCE     18,2     ;Обробка на встаткуванні першого виду
RELEASE     OBOR1   ;Звільнення встаткування першого виду
QUEUE       OCHER2   ;Вхід у чергу до встаткування другого виду
SEIZE       OBOR2   ;Визначення заняття встаткування другого виду
DEPART      OCHER2   ;Вихід із черги до встаткування другого виду
ADVANCE     30,4     ;Обробка на встаткуванні другого виду
RELEASE     OBOR2   ;Звільнення встаткування другого виду
QUEUE       OCHER3   ;Вхід у чергу до встаткування третього виду
SEIZE       OBOR3   ;Визначення заняття встаткування третього виду
DEPART      OCHER3   ;Вихід із черги до встаткування третього виду

```

```

ADVANCE      21,3      ;Обробка на встаткуванні третього виду
RELEASE      OBOR3     ;Звільнення встаткування третього виду
TERMINATE    ;Вихід виробу з ділянки
*****
GENERATE     28,5      ;Генерація другого потоку виробів
QUEUE       OCHER1    ;Вхід у чергу до встаткування першого виду
SEIZE       OBOR1     ;Визначення заняття встаткування першого виду
DEPART      OCHER1    ;Вихід із черги на обробку
ADVANCE     19,4      ;Обробка на встаткуванні першого виду
RELEASE     OBOR1     ;Звільнення встаткування першого виду
QUEUE      OCHER2    ;Вхід у чергу до встаткування другого виду
SEIZE      OBOR2     ;Визначення заняття встаткування другого виду
DEPART     OCHER2    ;Вихід із черги на обробку
ADVANCE    25,5      ;Обробка на встаткуванні другого виду
RELEASE    OBOR2     ;Звільнення встаткування другого виду
QUEUE     OCHER3    ;Вхід у чергу до встаткування третього виду
SEIZE     OBOR3     ;Визначення заняття встаткування третього виду
DEPART    OCHER3    ;Вихід із черги на обробку
ADVANCE   28,5      ;Обробка на встаткуванні третього виду
RELEASE   OBOR3     ;Звільнення встаткування третього виду
TERMINATE ;Вихід виробу з ділянки
*****
GENERATE    960      ;Моделювання часу роботи протягом дня
TERMINATE   1        ;Зменшення часу роботи ділянки на 1 хв
START      1

```

Для виведення параметрів моделювання натиснемо на пункт Edit (Виправлення) і в меню, що випадає, виберемо пункт Settings. З'явиться діалогове вікно SETTINGS (рис. 7.11), у якому на вкладці Reports (Звіти) відзначимо пункти Facilities (Канали обслуговування) і Queues (Черги).

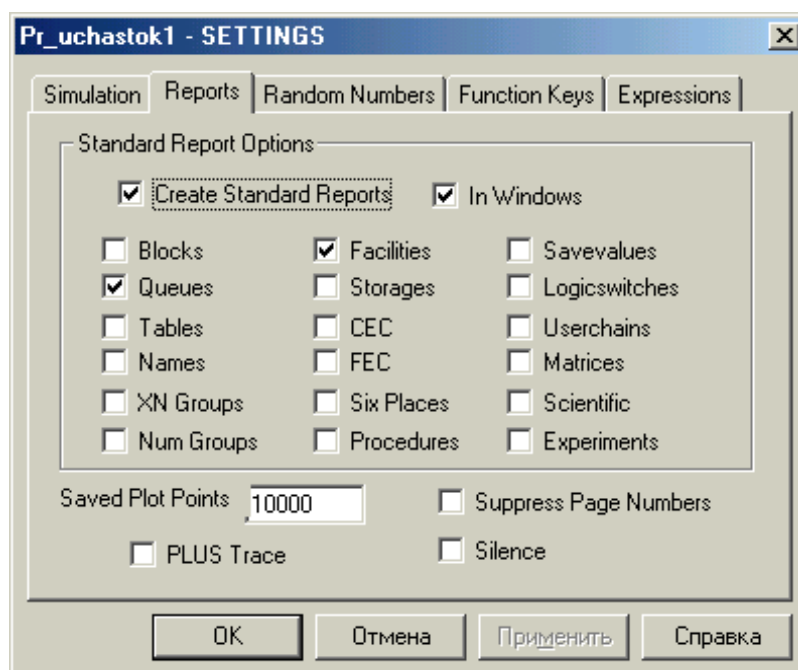


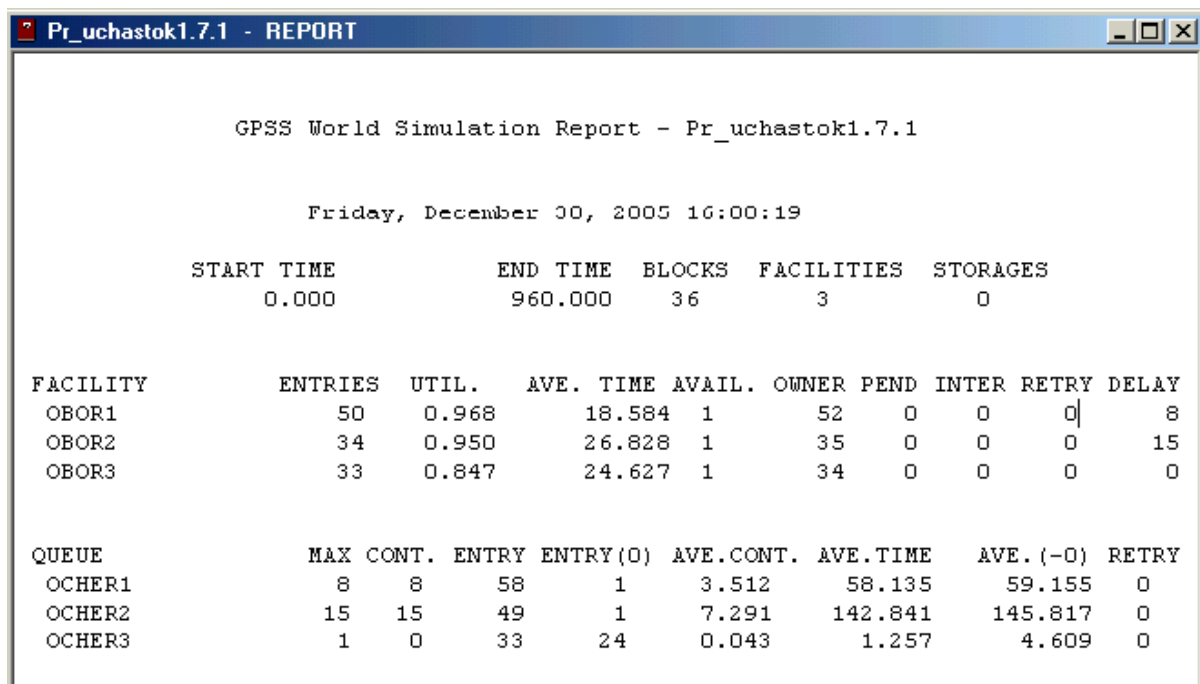
Рис. 7.11

Для початку моделювання необхідно:

натиснути на пункт **Command** головного меню системи;
у меню, що випадає, натиснути на пункт **Create Simulation**
(Створити модель, яка виконується).

Почнеться трансляція вихідної моделі. Тому що в програмі є керуюча команда **START**, автоматично почнеться виконання відтрансльованої програми.

Результати моделювання подані на рис. 7.12.



```
Pr_uchastok1.7.1 - REPORT

GPSS World Simulation Report - Pr_uchastok1.7.1

Friday, December 30, 2005 16:00:19

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
0.000              960.000    36       3           0

FACILITY           ENTRIES  UTIL.   AVE. TIME  AVAIL.  OWNER  PEND  INTER  RETRY  DELAY
OBOR1              50      0.968   18.584    1       52     0     0      0      8
OBOR2              34      0.950   26.828    1       35     0     0      0     15
OBOR3              33      0.847   24.627    1       34     0     0      0      0

QUEUE             MAX CONT.  ENTRY  ENTRY(0)  AVE.CONT.  AVE.TIME  AVE.(-0)  RETRY
OCHER1             8      8      58        1      3.512    58.135    59.155  0
OCHER2            15     15     49        1      7.291   142.841   145.817  0
OCHER3             1      0     33        24     0.043    1.257     4.609   0
```

Рис. 7.12

У верхньому рядку вікна **REPORT** (Звіт) вказується:

- **START TIME** (Початковий час) - 0.000;
- **END TIME** (Час закінчення) - 960.000;
- **BLOCKS** (Число блоків) - 36;
- **FACILITIES** (Число каналів обслуговування) - 3;
- **STORAGES** (Число накопичувачів) - 0.

Далі вказуються результати моделювання каналів обслуговування (**FACILITY**), які мають імена **OBOR1**, **OBOR2** й **OBOR3**:

- ENTRIES (Число входів) - 50, 34, 33;
- UTIL. (Коефіцієнт використання) - 0.968, 0.950, 0.847;
- AVE.TIME (Середній час обслуговування) -18.584, 26.828, 24.627;
- AVAIL. (Доступність - стан пристрою) - 1, 1, 1;
- OWNER (Можливе число входів) - 52, 35, 34;
- PEND (Кількість транзактів, що очікує виконання з перериванням) - 0, 0, 0;
- INTER (Кількість транзактів, перерваних на даний момент) - 0, 0, 0;
- RETRY (Повтор) - 0, 0, 0;
- DELAY (Кількість транзактів, що очікує заняття пристрою) - 8, 15, 0.

Нижче зазначені результати моделювання черг OSHER1, OSHER2 й OSHER3:

- MAX (Максимальний вміст черги) - 8, 15, 1;
- CONT. (Поточний вміст черги) - 8, 15, 0;
- ENTRY (Число входів у чергу) - 58, 49, 33;
- ENTRY(0) (Число входів у чергу з нульовим часом очікування) - 1, 1, 24;
- AVE.CONT. (Середнє число входів) - 3.512, 7.291, 0.043;
- AVE.TIME (Середній час перебування в черзі) - 58.135, 142.814, 1.257;
- AVE.(-0) (Середній час перебування в черзі без обліку нульових входів у чергу) - 59.155, 145.817, 4.609;
- RETRY (Відмовлено) - 0, 0, 0.

На підставі результатів моделювання можливо зробити наступні висновки.

1. Устаткування використовується ефективно. Коефіцієнти завантаження (**UTIL.**) першого, другого й третього видів устаткування становлять відповідно 0.968, 0.950 й 0.847.

2. Найбільша черга (**MAX**) утвориться перед другим видом устаткування (15 виробів) і першим видом устаткування

(8 виробів). Отже, перед цими видами встаткування необхідно передбачити місце для розміщення виробів, які обробляються. Перед третім видом устаткування черги немає.

3. Середній час перебування в черзі біля устаткування другого виду (**AVE.TIME**) становить 142.811 хв.

7.7. Моделювання ремонту колісних пар вагонів

Використаємо систему GPSSW для моделювання роботи ділянки ремонту колісних пар вагонів.

У колісному відділенні ремонт колісних пар здійснюється потоковим методом. Колісні пари подаються у відділення на оглядову площадку для визначення виду огляду й ремонту. Бригадир відділення з ремонту колісних пар робить попередній огляд й оприбуткування колісної пари. Потім колісна пара за допомогою мостового крана подається на позицію очищення. Після очищення колісна пара надходить у мийну машину. З мийної машини колісна пара надходить на позицію діагностики, де перевіряють її розміри. На наступній позиції вісь контролюють магнітним дефектоскопом, а також ультразвуковим дефектоскопом. Потім колісна пара направляється до двох колісноточарних верстатів, на одному з яких обточують поверхню кочення коліс.

Система, яка моделюється, має наступні параметри:

- потік колісних пар, що надходять у відділення, рівномірний;
- інтервал часу надходження колісних пар становить 7 ± 3 хв, тобто перебуває в межах від 4 до 10 хв;
- час, затрачений на попередній огляд і оприбуткування колісної пари, становить 9 ± 2 хв;
- час на очищення колісної пари дорівнює 10 ± 1 хв;
- час обмивки колісної пари в мийній машині становить 13 ± 1 хв;
- час на діагностику колісної пари становить 17 ± 2 хв;
- час на дефектоскопію колісної пари дорівнює 22 ± 2 хв;

- час на обточування колісної пари на верстаті становить 25 ± 2 хв;
- час на переміщення колісної пари між позиціями огляду, очищення, обмивки й діагностики становить 3 ± 1 хв, між позиціями діагностики, дефектоскопії й обточування - 5 ± 1 хв.

Потрібно визначити параметри функціонування ділянки цеху:

- коефіцієнт завантаження позицій ремонту;
- коефіцієнт завантаження мийної машини й колесотокарних верстатів;
- максимальне, середнє й поточне число колісних пар у кожній черзі;
- середній час обслуговування в кожному каналі обслуговування (на кожній позиції);
- середній час знаходження колісної пари в кожній черзі й ін.

Процес функціонування ділянки графічно поданий на рис. 7.13.

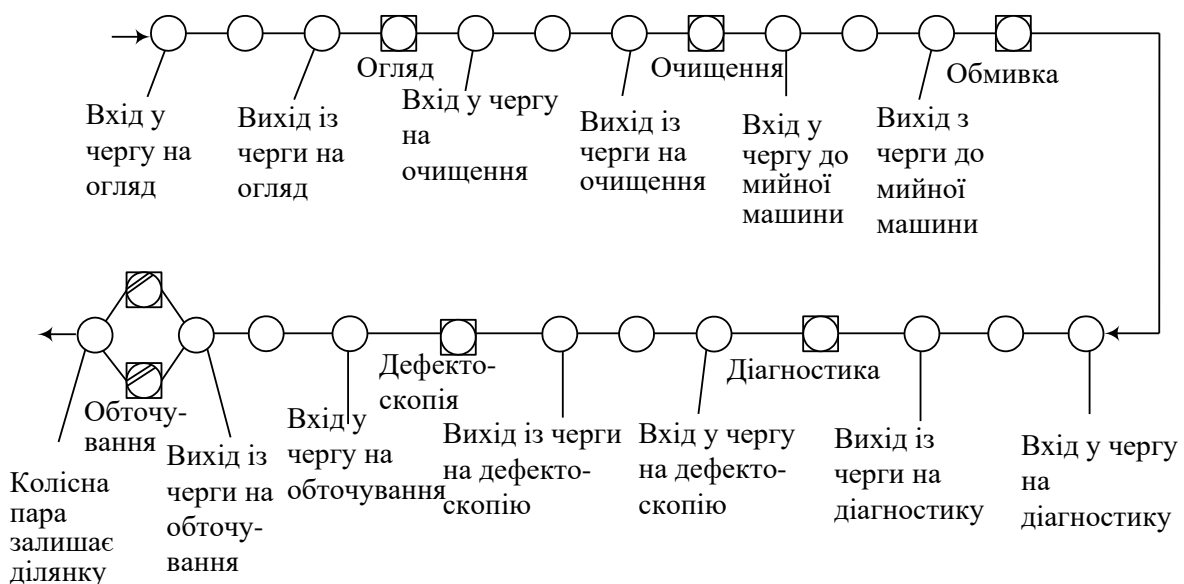


Рис. 7.13

Відповідно до технологічного процесу колісна пара, викочена з-під візка, надходить у колісне відділення на оглядову площадку для попереднього огляду. Якщо оглядова площадка

вільна, бригадир відділення з ремонту колісних пар робить попередній огляд й оприбуткування колісної пари. Якщо площадка зайнята, колісна пара стає в чергу. Після звільнення колісна пара виходить із черги й надходить на оглядову площадку. Протягом 9 ± 2 хв здійснюється попередній огляд і колісна пара залишає оглядову площадку (канал обслуговування).

Якщо наступний канал обслуговування (мостовий кран і позиція очищення) вільний, колісна пара надходить на позицію очищення, яка розташована на спеціально виділеній площадці. Якщо канал обслуговування зайнятий, колісна пара стає в чергу. Після звільнення позиції очищення колісна пара залишає чергу і надходить на позицію очищення. Те ж відбувається перед позицією обмивки, діагностики й дефектоскопії колісної пари. Потім колісна пара направляється до першого верстата. Якщо канал обслуговування (верстат) зайнятий, вона направляється до другого каналу обслуговування (верстату). Якщо другий канал обслуговування зайнятий, колісна пара стає в чергу.

Охарактеризуємо кожну подію в системі (у дужках зазначений відповідний оператор мови GPSS):

1. Надходження вимоги в систему (колісної пари на ділянку) - (GENERATE).
2. Вхід колісної пари в чергу до позиції попереднього огляду (QUEUE).
3. Перевірка зайнятості каналу обслуговування (позиції огляду) - (SEIZE).
4. Вихід колісної пари із черги до позиції огляду (DEPART).
5. Проведення попереднього огляду (ADVANCE).
6. Вихід колісної пари з позиції огляду (RELEASE).
7. Передача колісної пари на наступну позицію (ADVANCE).

Починаючи від входу колісної пари в наступну чергу, ці події повторюються на ділянках очищення, обмивки, діагностики й дефектоскопії колісної пари.

8. Вхід вимоги в накопичувач (ENTER).
9. Визначення каналу обслуговування (TRANSFER).

10. Очікування звільнення одного з каналів обслуговування (SEIZE).
11. Вихід з накопичувача (LEAVE).
12. Обслуговування вимоги (ADVANCE).
13. Звільнення каналу обслуговування (RELEASE).
14. Вихід вимоги із системи (TERMINATE).

Програма імітаційного моделювання починається із загальної характеристики моделі:

```
* GPSSW                                ОВТОСНКА.GPS
*****
* Моделювання ремонту колісних пар вагонів
*****
```

Оператор GPSS включає три частини: мітку, назву, поле змінних. Поле змінних може містити підполя, розділені комами для вказівки чисел, стандартних числових атрибутів, символів, ідентифікаторів і т.п.

Помістимо в наступному рядку оператор QTABLE , призначений для опису таблиці часу перебування в черзі до мийної машини:

```
t_mm QTABLE Ocher_mm,0,2,32 ;Збір інформації й побудова гістограми
```

У цьому рядку:

t_mm – мітка;
 QTABLE – назва оператора;
 Ocher_mm – ім'я черги до мийної машини.

Далі слідує ліва межа першого інтервалу таблиці, ширина інтервалу, кількість інтервалів таблиці, збільшена на 2.

Коментарі не ставляться до числа обов'язкових атрибутів і можуть бути відсутні.

Моделювання потоку колісних пар, що надходять на ділянку, виконується за допомогою оператора GENERATE :

```
GENERATE 7,3 ;Генерація потоку колісних пар
```

Тут число 7 - середній інтервал часу між прибуттям на ділянку двох колісних пар, які йдуть одна за іншою; число 3 - відхилення часу надходження колісної пари від середнього значення.

Колісна пара, що надійшла на ділянку, стає в чергу до позиції огляду, якщо черга є. Присвоїмо цій черзі ім'я `Ocher_osm`. Тоді оператор буде виглядати так:

```
QUEUE      Ocher_osm
```

Колісна пара може вийти із черги тільки після того, як звільниться оглядова площадка. Дано позиції огляду ім'я `Osmotr`. Перевірку зайнятості оглядової площадки можна виконати за допомогою такого оператора:

```
SEIZE      Osmotr
```

Вихід колісної пари із черги до позиції огляду моделюється таким оператором:

```
DEPART     Ocher_osm
```

Тепер промодельуємо час перебування колісної пари на оглядовій площадці. За умовою цей час становить 9 ± 2 хв.

```
ADVANCE    9,2
```

Після проведення попереднього огляду колісна пара залишає оглядову площадку. Це моделюється за допомогою оператора

```
RELEASE    Osmotr
```

На цьому перший сегмент програми можна вважати закінченим і перейти до другого сегмента.

Необхідно відмітити, що оператори `QUEUE` і `DEPART` мають однакове ім'я й виконують схожі функції: перший – фіксує вхід колісної пари в чергу, другий – вихід із черги. Ці оператори називаються парними. Те ж відноситься до операторів `SEIZE` й `RELEASE`, які мають своє унікальне ім'я.

Відповідно до технологічного процесу після попереднього огляду колісна пара надходить на позицію очищення. Послідовність подій у цьому каналі обслуговування аналогічна тій, яка розглянута для позиції огляду. Вона може бути наведена в такому вигляді:

QUEUE	Ocher_ochist
SEIZE	Ochistka
DEPART	Ocher_ochist
ADVANCE	10,1
RELEASE	Ochistka

Після виконання робіт на позиції очищення колісна пара надходить у наступний канал обслуговування - мийну машину. Моделювання процесу обмивки колісної пари може бути подане в такому вигляді:

QUEUE	Ocher_mm
SEIZE	Mmashina
DEPART	Ocher_mm
ADVANCE	13,1
RELEASE	Mmashina

Потім колісна пара надходить на позицію діагностики:

ADVANCE	4,1
QUEUE	Ocher_diag
SEIZE	Diagnos
DEPART	Ocher_diag
ADVANCE	17,2
RELEASE	Diagnos

Таким же чином виконується моделювання процесу на позиції дефектоскопії:

ADVANCE	4,1
QUEUE	Ocher_defect
SEIZE	Defect
DEPART	Ocher_defect
ADVANCE	22,2
RELEASE	Defect

Потім колісна пара надходить на позицію обточування. Обточування може здійснюватися на одному із двох колесотокарних верстатів.

Заняття й звільнення багатоканального пристрою імітується блоками ENTER і LEAVE. Оператор можна буде записати в такому вигляді:

```
ENTER NAK
```

У цьому операторі NAK - ім'я накопичувача, місткість якого заздалегідь визначена в програмі.

Для забезпечення можливості напрямку вимоги до вільного каналу використовується оператор TRANSFER:

```
TRANSFER Both, St_1, St_2
```

Спочатку колісна пара направляється до оператора з міткою St_1. Цим оператором є SEIZE, що записаний у такому вигляді:

```
St_1 SEIZE Stanok_1
```

Якщо канал обслуговування з міткою St_1 зайнятий, то вимога направляється до оператора із символічною міткою St_2. Цим оператором також є оператор SEIZE, що записується в такому вигляді:

```
St_2 SEIZE Stanok_2
```

Таким чином, вимога буде обслужена в одному із вільних каналів обслуговування. Але попередня вимога повинна запам'ятати канал, у який вона потрапила на обслуговування. Це можна зробити за допомогою оператора ASSIGN (Присвоїти). Оператор ASSIGN у параметрі вимоги під номером 1 запам'ятовує ім'я каналу, у який вимога пішла на обслуговування:

```
ASSIGN 1, Stanok_1
```

Після визначення вільного каналу й запису його імені за допомогою оператора TRANSFER вимога направляється до оператора з міткою COME:

TRANSFER , COME

Потім вимога попадає в оператор LEAVE (Вийти), де зменшується ємність накопичувача:

COME LEAVE NAK

Після виходу з накопичувача вимога надходить на обслуговування. Моделювання цього процесу в нашому завданні буде виглядати так:

ADVANCE 25,2

Після обслуговування вимога виходить із каналу обслуговування, і повинен з'явитися сигнал про звільнення каналу. Це робиться за допомогою оператора RELEASE (Звільнити):

RELEASE P1

Оператор вимоги під номером P1 містить ім'я каналу обслуговування, що звільняє.

Далі колісна пара залишає ділянку. Для цього використовується оператор TERMINATE (Завершити)

TERMINATE ; Завершення виконання операцій

Останній сегмент моделює роботу ділянки протягом робочого дня. Оскільки за одиницю часу обрана хвилина, то час моделювання системи становить $8 \times 60 = 480$ хв. Завершальний сегмент може виглядати так:

GENERATE 480
TERMINATE 1
START 1

Остаточно програма має такий вигляд:

```

* GPSSW                                ОВТОЧКА.GPS
*****
* Моделювання ремонту колісних пар вагонів
*****
NAK STORAGE 3
t_mm QTABLE Ocher_mm,0,2,32 ;Збір інформації й побудова гістограми
GENERATE 7,3 ;Генерація потоку колісних пар
QUEUE Ocher_osm ;Вхід кол. пари в чергу до позиції огляду
SEIZE Osmotr ;Перевірка зайнятості позиції огляду
DEPART Ocher_osm ;Вихід кол. пари із черги до позиції огляду
ADVANCE 9,2 ;Попередній огляд колісної пари
RELEASE Osmotr ;Вихід колісної пари з позиції огляду
*****
ADVANCE 4,1 ;Переміщення кол. пари на позицію очищення
QUEUE Ocher_ochist ;Вхід кол. пари в чергу до позиції очищення
SEIZE Ochistka ;Перевірка зайнятості позиції очищення
DEPART Ocher_ochist ;Вихід кол. пари із черги до позиції очищення
ADVANCE 10,1 ;Очищення колісної пари
RELEASE Ochistka ;Вихід кол. пари з позиції очищення
*****
ADVANCE 4,1 ;Переміщення кол. пари до мийної машини
QUEUE Ocher_mm ;Вхід кол. пари в чергу до мийної машини
SEIZE Mmashina ;Перевірка зайнятості мийної машини
DEPART Ocher_mm ;Вихід кол. пари із черги до мийної машини
ADVANCE 13,1 ;Обмивка колісної пари
RELEASE Mmashina ;Вихід кол. пари з мийної машини
*****
ADVANCE 4,1 ;Переміщення кол. пари до позиції діагностики
QUEUE Ocher_diag ;Вхід кол. пари в чергу на діагностику
SEIZE Diagnos ;Перевірка зайнятості позиції діагностики
DEPART Ocher_diag ;Вихід із черги на діагностику
ADVANCE 17,2 ;Діагностика
RELEASE Diagnos ;Вихід кол. пари з позиції діагностики
*****
ADVANCE 4,1 ;Переміщення кол. пари до позиції дефектоскоп.
QUEUE Ocher_defect ;Вхід кол. пари в чергу на дефектоскоп.
SEIZE Defect ;Перевірка зайнятості позиції дефектоскопир.
DEPART Ocher_defect ;Вихід із черги на діагностику
ADVANCE 22,2 ;Дефектоскопирування
RELEASE Defect ;Вихід кол. пари з позиції дефектоскопир.
*****
ADVANCE 5,1 ;Переміщення кол. пари до позиції обточування
ENTER NAK ;Збільшити місткість накопичувача
TRANSFER Both,St_1,St_2;Перехід кол. пари до вільного верстата
St_1 SEIZE Stanok_1 ;Визначення зайнятості 1 верстата
ASSIGN 1,Stanok_1 ;Запам'ятати канал обслуговування
TRANSFER ,COME ;Направити до оператора з міткою COME
St_2 SEIZE Stanok_2 ;Визначення зайнятості 2 верстатів
ASSIGN 1,Stanok_2 ;Запам'ятати канал обслуговування
COME LEAVE NAK ;Зменшити місткість накопичувача
ADVANCE 25,2 ;Обточування кол. пари на верстаті
RELEASE P1 ;Звільнення верстата
*****
TERMINATE ;Завершення виконання операцій
GENERATE 480 ;Моделювання часу роботи ділянки
TERMINATE 1 ;Вихід кол. пари з ділянки
START 1 ;Початок моделювання
*****

```

Після запуску програми автоматично з'являється вікно REPORT зі звітом про результати моделювання. Основні результати моделювання подані в такому вигляді:

GPSS World Simulation Report - ОБТОЧКА4.17.1

Saturday, December 24, 2005 13:00:09

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	480.000	44	7	1

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
DELAY								
OSMOTR	52	0.981	9.057	1	53	0	0	14
OCHISTKA	47	0.951	9.713	1	48	0	0	4
MMASHINA	34	0.919	12.981	1	35	0	0	11
DIAGNOS	26	0.886	16.352	1	27	0	0	7
DEFECT	19	0.843	21.299	1	20	0	0	6
STANOK_1	10	0.471	22.589	1	19	0	0	0
STANOK_2	8	0.417	25.029	1	18	0	0	0

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE. CONT.	AVE. TIME	AVE. (-0)
RETRY							
OCHER_MM	11	11	45	1	4.973	53.044	54.250 0
OCHER_OSM	15	14	66	1	7.701	56.007	56.868 0
OCHER_OCHIST	5	4	51	1	2.012	18.938	19.317 0
OCHER_DIAG	8	7	33	1	3.163	46.007	47.444 0
OCHER_DEFECT	6	6	25	1	2.467	47.361	49.334 0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY
DELAY									
NAK	3	3	0	1	18	1	0.000	0.000	0 0

У третьому рядку звіту вказується:

- START TIME (Початковий час) - 0.000;
- END TIME (Час закінчення) - 480.000;
- BLOCKS (Число блоків) - 44;
- FACILITIES (Число каналів обслуговування) - 7;
- STORAGES (Число накопичувачів) - 1.

Далі наводяться результати моделювання каналів обслуговування OSMOTR, OCHISTKA, MMASHINA, DIAGNOS, DEFECT, STANOK_1, STANOK_2:

- ENTRIES (Число входів) - 52, 47, 34, 26, 19, 10, 8;
- UTIL. (Коефіцієнт використання) - 0.981, 0.951, 0.919, 0.866, 0.843, 0.471, 0.417;
- AVE.TIME (Середній час обслуговування) - 9.057, 9.713, 12.981, 16.352, 21.299, 22.589, 25.029;
- AVAIL. (Доступність - стан пристрою) - 1, 1, 1, 1, 1, 1, 1;
- OWNER (Можливе число входів) - 52, 48, 35, 27, 20, 19,18;
- PEND (Кількість транзактів, що очікує виконання з перериванням) - 0, 0, 0, 0, 0, 0, 0;
- INTER (Кількість транзактів, перерваних на даний момент) - 0, 0, 0, 0, 0, 0, 0;
- RETRY (Повтор) - 0, 0, 0, 0, 0, 0, 0;
- DELAY (Кількість транзактів, що очікує заняття пристрою) - 14, 4, 11, 7, 6, 0, 0.

Нижче зазначені результати моделювання черг OSHER_MM, OSHER_OSM, OSHER_OCHIST, OSHER_DIAG, OSHER_DEFECT:

- MAX (Максимальний вміст черги) - 11, 15, 5, 8, 6;
- CONT. (Поточний вміст черги) - 11, 14, 4, 7, 6;
- ENTRY (Число входів у чергу) - 45, 66, 51, 33, 25;
- ENTRY(0) (Число входів у чергу з нульовим часом очікування)- 1, 1, 1, 1, 1;
- AVE.CONT. (Середнє число входів) - 4.973, 7.701, 2.012, 3.163, 2.467;
- AVE.TIME (Середній час перебування в черзі) - 53.044, 56.007, 18.937, 46.007, 47.361;
- AVE.(-0) (Середній час перебування в черзі без обліку нульових входів у чергу) - 54.250, 56.868, 19. 137, 47.444, 49.334;
- RETRY (Відмовлено) - 0, 0, 0, 0, 0.

Нижче зазначені результати функціонування накопичувача (STORAGE) під ім'ям NAK:

- CAP. (Місткість) - 3;
- REM. (Вилучений) - 3;
- MIN. (Мінімальний уміст) - 0;
- MAX. (Максимальний уміст) - 1;
- ENTRIES (Число входів) - 18;
- AVL. (Доступність) - 1;
- AVE.C. (Середня місткість) - 0.000;
- UTIL. (Коефіцієнт використання) - 0.000;
- RETRY (Кількість транзактів, що очікує виконання спеціальної умови, що залежить від стану пам'яті) - 0;
- DELAY (Кількість транзактів, що очікують у блоках ENTER, пов'язаних з даною пам'яттю) - 0.

Аналіз результатів дозволяє зробити такі висновки:

1. Найбільш завантаженою є позиція огляду, що використовувалася протягом 98.1% часу. Коефіцієнт використання (**UTIL.**) позиції огляду дорівнює 0.981, позиції очищення - 0.959, мийної машини - 0.919.

2. Найменш завантаженими є колесотокарні верстати. Коефіцієнт використання верстата STANOK_1 дорівнює 0.471, верстата STANOK_2 - 0.417. Інакше кажучи, верстат STANOK_1 використовувався протягом 47.1% часу, верстат STANOK_2 - протягом 41.7% часу.

3. Позиція огляду була зайнята обслуговуванням 52 рази. Для позиції огляду число входів (**ENTRIES**) дорівнює 52, для позиції очищення – 47, для мийної машини – 34. Верстат STANOK_1 був зайнятий обслуговуванням 10 разів, верстат STANOK_2 - 8 разів. За зміну (8 год) на обох верстатах оброблено 18 колісних пар.

4. Середній час обслуговування (**AVE.TIME**) на позиції огляду становить 9.057 хв, на позиції очищення - 9.713 хв, на позиції обмивки - 12.981 хв. Середній час обробки колісних пар на верстаті STANOK_2 становить 25.029 хв.

5. Відмовлено в обслуговуванні (**DELAY**) на позиції огляду – 14 колісним парам, на позиції діагностики – 7 колісним парам, на позиції обмивки – 11 колісним.

6. Черга до мийної машини ніколи не перевищувала (**MAX**) 11 колісних пар. Максимальна довжина черги до позиції огляду – 15 колісних пар, до позиції діагностики – 8 колісних пар. Черги до колесотокарних верстатів не було.

7. Поточна довжина черги (**CONT**) до мийної машини становить 11 колісних пар, до позиції огляду – 14 колісних пар, до позиції діагностики – 7 колісних пар.

8. Середня довжина черги (**AVE.CONT.**) до позиції огляду дорівнює 7.701 колісних пар, до мийної машини - 4.973 колісних пар.

9. Середній час перебування в черзі (**AVE.TIME**) до мийної машини дорівнює 53.044 хв, до позиції огляду – 56.007 хв, до позиції дефектоскопії - 47.361 хв.

СПИСОК ЛІТЕРАТУРИ

1. Системы автоматизированного проектирования: Учеб. пособие для вузов: В 9 кн. - Кн. 1: Принципы построения и структура / Под ред. И. П. Норенкова. – М.: Высш. шк., 1986. – 127 с.
2. Системы автоматизированного проектирования: Учеб. пособие для вузов: В 9 кн. - Кн. 2: Технические средства и операционные системы / Д.М. Жук, А.В. Мартынюк, П.А. Сомов; Под ред. И. П. Норенкова.– М.: Высш. шк., 1986. – 159 с.
3. Системы автоматизированного проектирования: Учеб. пособие для вузов: В 9 кн. - Кн. 4: Математические модели технических объектов / В.А. Трудоношин, Н.В. Пивоварова; Под ред. И. П. Норенкова. – М.: Высш. шк., 1986. – 166 с.
4. Системы автоматизированного проектирования: В 9 кн.- Кн. 5: Автоматизация функционального проектирования: Учеб. пособие для вузов/ П.К. Кузьмик, В.Б. Маничев; Под ред. И. П. Норенкова. – М.: Высш. шк., 1986. – 144 с.
5. Системы автоматизированного проектирования: В 9 кн. - Кн. 6. Автоматизация конструкторского и технологического проектирования: Учеб. пособие для вузов/ Н.М. Капустин, Г.В. Васильев; Под ред. И. П. Норенкова. – М.: Высш. шк., 1986. – 191 с.
6. Разработка САПР: Практ. пособие: В 10 кн. - Кн. 1: Проблемы и принципы создания САПР / А.В. Петров, В.М. Черненький; Под ред. А.В. Петрова. - М.: Высш. шк., 1990.- 143 с.
7. Разработка САПР: Практ. пособие: В 10 кн. - Кн. 2: Системотехнические задачи создания САПР / А.Н. Данчул, Л.Я. Полуян; Под ред. А.В. Петрова. - М.: Высш. шк., 1990.- 144 с.
8. Шевченко В.В., Головки В.Ф. Дослідження операцій у виробництві, ремонті та експлуатації вагонів: Навч. посібник .- Харків: УкрДАЗТ, 2006. - 171 с.
9. Банди Б. Методы оптимизации. Вводный курс / Пер. с англ. – М.: Радио и связь, 1980.- 128 с.
10. Реклейтис Г., Рейвиндоан А., Рэгсдел К. Оптимизация в технике: В 2 кн. / Пер. с англ. – М.: Мир, 1986. - Кн. 1. – 349 с.

11. Реклейтис Г., Рейвиндоан А., Рэгсдел К. Оптимизация в технике: В 2 кн / Пер. с англ. – М.: Мир, 1986. - Кн. 2. – 320 с.
12. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация / Пер. с англ. – М.: Мир, 1985. – 598 с.
13. Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ: Учеб. пособие для втузов. – М.: Высш. шк., 1989. – 367 с.
14. Советов Б.Я., Яковлев С.А. Моделирование систем: Учеб. пособие для втузов. - 3-е изд., перераб. и доп. – М.: Высш. шк., 2001. – 343 с.
15. Грабовский Ральф: Иллюстрированный справочник по AutoCAD 2004 / Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 880 с.
16. Зарубин В.С. Математическое моделирование в технике: Учеб. для втузов/ Под ред. В.С. Зарубина, А.П. Крищенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 496 с.
17. Рыжиков Ю.И. Имитационное моделирование. Техника и технологии. – С.Пб.: КОРОНА принт; М.: Альтекс-А, 2004.- 384 с.
18. Шимкович Д.Г. Расчет конструкций в visualNastran for Windows. – М.: ДМК Пресс, 2004. – 704 с.
19. Разработка САПР: Практ. пособие: В 10 кн. - Кн. 9: Имитационное моделирование / В.М. Черненко. - М.: Высш. шк., 1990.- 112 с.

В.В. Шевченко, В.Ф. Головка
АВТОМАТИЗОВАНЕ ПРОЕКТУВАННЯ
ВАГОНІВ
НАВЧАЛЬНИЙ ПОСІБНИК

Відповідальний за випуск Шевченко В.В.

Редактор Решетилова В.В.

Підписано до друку 18.07.07 р.

Формат паперу 60x84 1/16 . Папір писальний.

Умовн.-друк.арк. 13,25 Обл.-вид.арк. 13,5.

Замовлення № Тираж 300 Ціна

Видавництво УкрДАЗТу, свідоцтво ДК 2874 від 12.06.2007 р.
Друкарня УкрДАЗТу,
61050, Харків - 50, пл. Фейєрбаха, 7