

Яблонь Л.С.

# **КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ФІЗИЧНИХ ЯВИЩ**

*Лабораторний практикум*

*для студентів спеціальності 014.08 Середня освіта (Фізика)*

Івано-Франківськ  
2020

Яблонь Л.С. Комп'ютерне моделювання фізичних явищ. Лабораторний практикум для студентів спеціальності 014.08 Середня освіта (Фізика). – Івано-Франківськ, 2020. – 87 с.

У пропонованому посібнику подано навчальну програму та опис лабораторних робіт зі спеціального курсу «Комп'ютерне моделювання фізичних явищ» для студентів спеціальності Середня освіта (Фізика)..

Рецензенти: к. ф.-м. н., доц. Федорів В.Д.,  
к.ф.-м.н., доц. Никируй Л.І.

## ВСТУП

Комп'ютерне моделювання є одним з ефективних методів вивчення фізичних систем. Часто комп'ютерні моделі простіше і зручніше досліджувати, вони дозволяють проводити обчислювальні експерименти, реальна постановка яких утруднена або може дати непередбачуваний результат. Логічність комп'ютерних моделей дозволяє виявити основні чинники, визначальні властивості об'єктів, що вивчаються, досліджувати поведінку фізичної системи при зміні її параметрів і початкових умов.

Створення моделі є «чарівним інструментом» пізнання, що дозволяє на різних ступенях дослідження виділити головні, найістотніші характеристики фізичного процесу. В цьому і полягає мета моделювання фізичного процесу. Необхідність врахування різноманітних властивостей системи, наявність кореляційних зв'язків між значним числом змінних, що характеризують процеси в системі створюють значні, а часом і непереборні перешкоди, в створенні аналітичних моделей системи. В такій ситуації єдиним шляхом вирішення поставленої задачі є імітаційне моделювання, яке можливо здійснити лише за допомогою ЕОМ.

Задачею курсу «Комп'ютерне моделювання фізичних явищ» є оволодіння навиками роботи в системах комп'ютерної математики Excel, MatLab та AutoCad, які останніми роками в наукових та інженерно-технічних колах отримали широке поширення, а в багатьох країнах прийняті як офіційні засоби оформлення інженерної документації і наукових публікацій.

Математичний апарат, який використовується в MATLAB, гранично наближений до сучасного математичного апарату інженера і ученого і спирається на обчислення, що проводяться з матрицями, векторами і комплексними числами. MATLAB надає широкі можливості для роботи з математичними моделями різноманітних фізичних явищ.

Програма передбачає вивчення студентами теоретичних основ одного із ефективних методів вивчення фізичних систем, яким є комп'ютерне моделювання, та оволодіння навиками роботи в системі комп'ютерної математики MATLAB, яка останніми роками в наукових і інженерно-технічних колах отримала широке розповсюдження, а в багатьох країнах прийнята як офіційний засіб оформлення інженерної документації і наукових публікацій. У програмі широко представлені питання, які стосуються теорії комп'ютерного моделювання фізичних явищ. Достатньо широко представлені питання, в яких висвітлюються можливості пакету MATLAB:

- математичні обчислення – символні і чисельні;
- розробка алгоритмів, додатків і призначеного для користувача інтерфейсу;
- моделювання;
- аналіз даних, дослідження і візуалізація; наукова і технічна графіка.

Система Autocad дозволяє працювати одночасно з декількома кресленнями, має потужні засоби візуалізації тривимірних об'єктів, забезпечує зв'язок графічних об'єктів із зовнішніми базами даних.

# ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

## Змістовий модуль 1.

### Комп'ютерні моделі та їх роль у вивченні фізичних явищ.

**Тема 1. Вступ. Аналітичне та імітаційне моделювання.** Історична довідка про виникнення та розвиток імітаційного моделювання (машинної імітації). Моделювання та його використання в науці і техніці. Математичне (аналітичне) моделювання. Макетне моделювання. Фізичне моделювання: критерії подібності, аналіз розмірностей. Аналогове моделювання: механічні та електричні системи аналогій; аналогії полів; аналогові обчислювальні машини. Машинна імітація. Основні напрями використання машинної імітації.

**Тема 2. Сутність імітаційного моделювання.** Поняття імітаційного моделювання та машинної імітації. Переваги методу машинної імітації. Головні вади методу. Умови доцільності використання машинної імітації. Цілі машинної імітації: вивчення діючої системи; аналіз гіпотетичної системи; проектування більш досконалої системи. Встановлення адекватності імітаційної моделі еволюційних процесів; однорідне градуювання модельного (системного) часу – принцип часового приросту; неоднорідне градуювання модельного часу – принцип особливих станів. Програма реалізації імітаційної моделі. Мови машинного моделювання: мови моделювання неперервних процесів; мови моделювання неперервно-дискретних процесів, мови моделювання дискретних процесів. Відмінності мов імітаційного моделювання.

**Тема 3. Поняття про метод Монте-Карло.** Розвиток і застосування методу Монте-Карло. Деякі приклади застосування методу для розв'язування детермінованих задач. Точність оцінки ймовірності за допомогою відносної частоти, отриманої методом Монте-Карло. Квазірівномірні числа.

**Тема 4. Планування експериментів при дослідженні та оптимізації систем.** Планування експериментів при дослідженні систем. Схеми оцінки впливу факторів за допомогою коефіцієнтів регресії. Головний ефект. Змішування оцінок. Генеруюче співвідношення. Визначальний контраст.

Перший спосіб пошуку екстремуму функції відгуку. Другий спосіб пошуку екстремуму функції відгуку (метод Бокса – Уільсона). Рух у напрямі крутого сходження.

## **Змістовий модуль 2**

### **(Physics Education Technology), Excel, Autocad та MatLab як засоби комп'ютерного моделювання фізичних явищ**

**Тема 5.** Віртуальні лабораторії на PhET, що демонструють різні явища в галузі фізики. Лабораторія гравітаційних сил. Резонанс. Тиск в рідинах. Рух рідини. Геометрична оптика. Колір, як його бачить людина. Заломлення світла. Електропровідність. Лабораторія електрики: постійний струм. Закони Фарадея.

**Тема 6. Основні поняття і принципи роботи в програмі Excel.** Моделювання рівномірного руху тіла. Моделювання рівноприскореного руху тіла. Моделювання коливального руху на прикладі математичного маятника. Моделювання руху тіла, кинутого під кутом до горизонту. Розв'язування диференціального рівняння другого порядку чисельним методом. Коливання пружинного маятника. Моделювання падіння тіла з урахуванням опору середовища

**Тема 7. Основні поняття і принципи роботи в Autocad.** Запуск програми Autocad. Настроювання системного середовища. Інтерфейс програми Autocad. Засоби організації креслення. Введення команд. Введення даних. Команди системи Autocad, які використовуються при створення робочого середовища. Команди встановлення режимів креслення. Команда перегляду параметрів робочого середовища. Команди управління зображенням на екрані.

**Тема 8. Засоби креслення. Засоби редагування креслень. Нанесення розмірів на креслення. Додаткові засоби формування креслень.** Команди побудови ліній. Команди побудови багатокутників, кіл, еліпсів. Команди генерації тексту. Команди нанесення штрихувань.

Команди редагувань. Додатковий засіб редагування об'єктів. Редагування поліліній. Редагування штрихувань. Редагування тексту. Команди нанесення розмірів. Команди редагування розмірного блоку. Редагування розмірних стилів.

**Тема 9. Загальна характеристика системи MatLab.** Вхід в систему MATLAB. Основні відомості про правила роботи в режимі прямих обчислень. Поняття про математичний вираз. Типи і формати даних. Формування векторів і матриць. Особливості задання векторів і матриць. Операції сумування елементів, перетворення і діагоналізації матриці. Операції з робочою областю і текстом сесії.

**Тема 10. Основи графічної візуалізації обчислень.** Основні відомості. Функції активації і управління графічними вікнами. Інтерфейс графічних вікон. Функції побудови двовимірних графіків. Функції керування оформленням графіків. Побудова гістограми. Побудова трьохвимірних графіків.

**Тема 11. Основи програмування в середовищі MATLAB.** Структура і властивості файлів сценаріїв. Структура М-файла-функції. Глобальні змінні. Керування потоками.

## ПЕРЕЛІК ТЕМ ЛАБОРАТОРНИХ РОБІТ

1. Віртуальні лабораторії на PhET, що демонструють різні явища в галузі фізики. Лабораторія гравітаційних сил. Резонанс. Тиск в рідинах. Рух рідини. Геометрична оптика. Колір, як його бачить людина. Заломлення світла. Електропровідність. Лабораторія електрики: постійний струм. Закони Фарадея (8 год)
2. Моделювання рівномірного руху тіла. Моделювання рівноприскореного руху тіла (2 год)
3. Моделювання коливального руху на прикладі математичного маятника (2 год)
4. Моделювання руху тіла, кинутого під кутом до горизонту (2 год)
5. Розв'язування диференціального рівняння другого порядку чисельним методом. Коливання пружинного маятника (2 год)
6. Моделювання падіння тіла з урахуванням опору середовища (2 год)
7. Основні поняття і принципи роботи в Autocad. Підготовка робочого середовища (2 год)
8. Засоби креслення. Засоби редагування креслень. Нанесення розмірів на креслення. Додаткові засоби формування креслень (4 год)
9. Знайомство з режимом "прямих обчислень" пакета прикладних програм MatLab (2 год)
10. Знайомство з графічними можливостями пакета прикладних програм MatLab (2 год)
11. Побудова гістограм, трьохмірних графіків та застосування анімації (2 год)
12. Засоби програмування системи MatLab (2 год)
13. Моделювання фігур Ліссажу (2 год)
14. Моделювання коливань математичного маятника (2 год)
15. Моделювання руху частинок у центральному полі (2 год)
16. Моделювання розподілу Максвелла молекул ідеального газу за швидкостями у системі MatLab (2 год)

## Лабораторна робота № 1

### Віртуальні лабораторії на PhET, що демонструють різні явища в галузі фізики

Розглянути лабораторні роботи: Лабораторія гравітаційних сил. Резонанс. Тиск в рідинах. Рух рідини. Геометрична оптика. Колір, як його бачить людина. Заломлення світла. Електропровідність. Лабораторія електрики: постійний струм. Закони Фарадея. Підготувати теорію.

## Лабораторна робота № 2

### Моделювання рівномірного руху тіла

#### Теоретичні відомості

Розглянемо моделювання такого фізичного процесу, як рух тіла з деякою сталою швидкістю  $\vec{v} = const$ . Оскільки жодна з характеристик швидкості (напрямок і величина) не змінюються, то рух буде відбуватися вздовж прямої лінії, тобто є прямолінійним. Сумістимо з цією прямою вісь  $Ox$ . Кожну секунду координата  $x$  тіла буде отримувати один і той же приріст, тому в будь-який момент часу вона може бути знайдена як

$$x = v_x t,$$

де  $v_x$  – проекція вектора швидкості на вісь  $Ox$ . Якщо в початковий момент часу ( $t_0 = 0$ ) положення тіла не збігалось з початком відліку, то рівняння буде мати вигляд:

$$x = x_0 + v_x t.$$

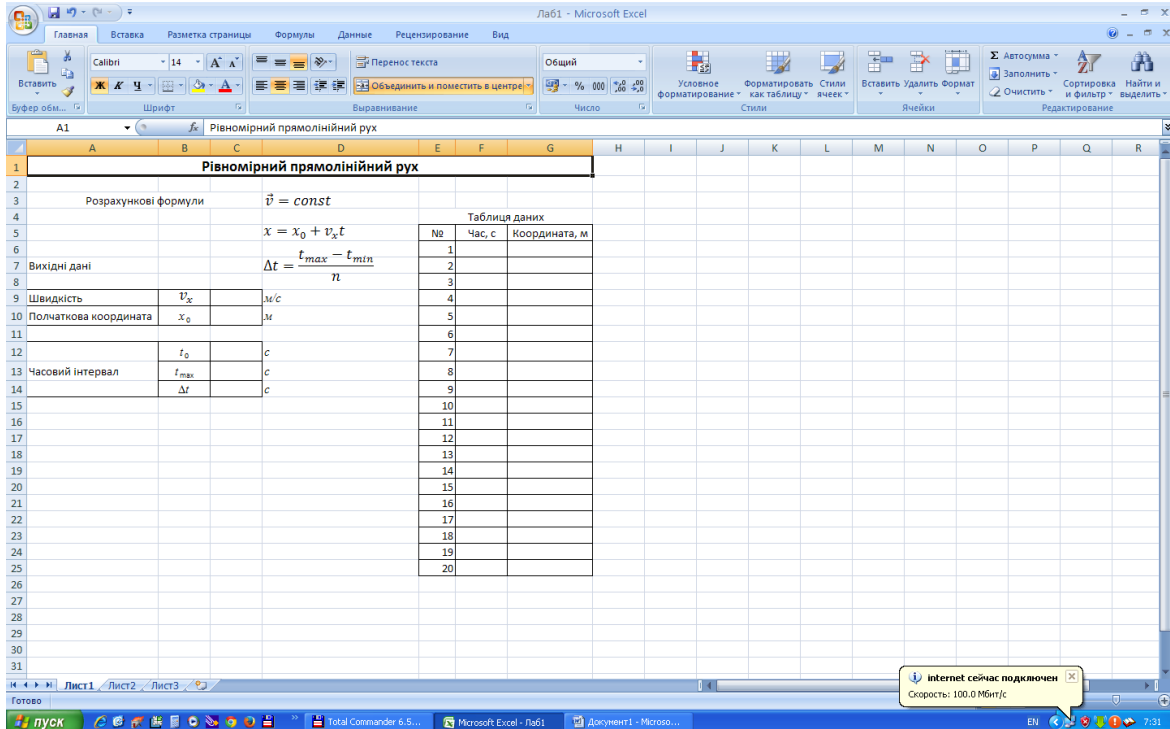
Проекція вектора швидкості – величина алгебраїчна, тобто вона може бути і позитивною, і негативною в залежності від того, який кут  $\alpha$  утворює вектор швидкості з напрямком осі  $Ox$ . Якщо  $\alpha = 0^\circ$ ,  $v_x > 0$  (тобто в цьому випадку  $v_x = v$ ), якщо  $\alpha = 180^\circ$ ,  $v_x < 0$  (і, отже,  $v_x = -v$ ).

Графічне моделювання процесу рівномірного прямолінійного руху буде полягати в побудові графіка залежності  $x = f(t)$  при різних значеннях і напрямках швидкості.

#### Завантаження файлу таблиці



# 1. Запустити Excel.



## 2. Створити модель 1 (рис.), в якій вихідними даними будуть:

- початкове положення (координата) тіла;
- проекція швидкості на обрану вісь;
- часовий інтервал (задається початковим і кінцевим моментами часу), протягом якого розглядається рух (час протікання процесу).

*Змінні* величини: час і координата тіла; *постійні*: проекція швидкості на вісь  $Ox$  (це означає, що її числове значення в процесі руху не змінюється).

Для побудови графіка  $x = f(t)$  необхідно отримати певне число точок  $(t; x)$ . Отже, необхідно задати крок змінної  $t$ , а саме  $\Delta t$ :

$$\Delta t = \frac{t_{max} - t_{min}}{n}, \quad (1.1)$$

Якщо в даній формулі  $n$  означає кількість підінтервалів, на яку розбито обраний проміжок часу, то таблиця буде містити  $n + 1$  точок.

Зберегти файл під назвою *Лабораторна робота 1.xls*.

### Заповнення таблиці

1. Занести вихідні числові дані в комірки C9, C10 і C13.
2. У комірці C14 розрахувати крок зміни часу для 20 підінтервалів за формулою (1) ( $t_{min} = t_0$ ).
3. Заповнити таблицю даних:
  - У комірку F6 занести початковий момент часу з комірки C12 способом копіювання;

- У комірці F7 вводимо формулу, яка розраховує наступний (після початкового) момент часу:  $= F6 + \$ C \$ 14$  , після чого скопіюємо її в інші комірки колонки таблиці з заголовком "Час, с";
- скласти і занести в комірку G7 формулу, що дозволяє розрахувати координату у відповідний їй момент часу.
- Скопіювати формулу з комірки G7 в діапазон комірок G8 : G26 .

4 Назвіть аркуш, наприклад, *Рівномірний рух*.

### ***Побудова графіка залежності $x(t)$***

Дані для побудови графіка залежності координати рівномірного руху тіла від часу знаходяться в діапазоні комірок F6 : G26, створеної таблиці. Значення в стовпчику F (діапазон F6 : F26) будуть відкладатися по горизонтальній осі (вісь  $x$ ), значення в стовпчику G (діапазон G6 : G26) – по вертикальній осі (вісь  $y$ ).

1. Виділити діапазон даних.
2. Вибрати команду Вставка - Діаграма ...
3. За допомогою майстра діаграм побудувати та оформити графік.

#### **Основні вимоги:**

- тип діаграми: *точкова*;
- ім'я ряду даних: *залежність  $x$  від  $t$*
- назва діаграми: *графік рівномірного руху*;
- вісь  $x$ : *час  $t$  , с*;
- вісь  $y$ : *координата  $x$  , м*;
- включити основні лінії сітки;
- відключити легенду;
- розташувати діаграму на окремому аркуші.

Зберегти дані.

### ***Завдання***

1. Відформатувати числові дані в таблиці, призначивши діапазону F6 : G26 числовий формат з двома десятковими знаками після коми.
2. Простежити, чи впливає крок програми (величина  $\Delta t$  ) на вигляд графіка.
3. Змінюючи початкові дані в комірках C9 , C10 , C13 і C14 , простежити за змінами вигляду графіка.
4. Задати від'ємне значення проекції швидкості. Який зміст знаку "-" у значенні проекції швидкості? у значенні координати?

# Лабораторна робота № 3

## Моделювання рівноприскореного руху тіла

### Теоретичні відомості

Розглянемо прямолінійний рівноприскорений рух ( $\vec{a} = const$ ). Оскільки рух відбувається вздовж прямої, то для його опису достатньо однієї координати. Нехай тіло рухається вздовж осі Oy. За визначенням прискорення

$$\vec{a} = \frac{\vec{v} - \vec{v}_0}{t},$$

звідси

$$\vec{v} = \vec{v}_0 + \vec{a}t.$$

Оскільки рух рівноприскорений, кожену секунду швидкість отримує один і той же приріст. Перепишемо цей вираз у проекції на обраний напрям осі Oy:

$$v_y = v_{0y} + a_y t.$$

Координата тіла змінюватиметься за законом:

$$y(t) = y_0 + v_{0y}t + \frac{a_y t^2}{2},$$

де  $y_0$  – початкове положення тіла, тобто його координата в момент часу  $t_0 = 0$ .

Таким чином, ми розглянули математичну модель рівноприскореного руху, а графічне моделювання буде полягати у побудові графіків залежності  $v_y = f(t)$  та  $y = f(t)$  при різних значеннях  $a_y$  і  $v_{0y}$ .

### Завантаження файлу таблиці

#### 1. Запустити Excel.

The screenshot shows a Microsoft Excel spreadsheet titled 'Ла62 - Microsoft Excel'. The spreadsheet is titled 'Рівноприскорений рух' (Uniformly accelerated motion). It contains the following content:

**Розрахункові формули:**

- $\vec{a} = const$
- $v_y = v_{0y} + a_y t$
- $y(t) = y_0 + v_{0y}t + \frac{a_y t^2}{2}$
- $\Delta t = \frac{t_{max} - t_{min}}{n}$

**Вихідні дані**

Початкова координата	$y_0$	м
Початкова швидкість	$v_{0y}$	м/с
Прискорення	$a_y$	м/с <sup>2</sup>
Часовий інтервал	$t_{min}$	с
	$t_{max}$	с
Крок	$\Delta t$	с

**Таблиця даних**

№	Час, с	Швидкість, м/с	Координата, м
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			

2. Створити *модель 2* (рис.), в якій вихідними даними будуть:

- координата тіла  $y_0$ ;
- 2) проекція початкової швидкості на обрану вісь;
- 3) проекція прискорення на обрану вісь;
- часовий інтервал  $t_{min} \div t_{max}$  протягом якого розглядається рух.

*Змінні* величини: час, швидкість і координата тіла; *постійні*: проекція прискорення та початкової швидкості на вісь  $Oy$

3. Зберегти файл під назвою *Лабораторна робота 2.xls*.

### **Заповнення таблиці**

1. Заповнити таблицю вихідних даних.
2. У комірці C14 розрахувати крок зміни часу для 20 підінтервалів за формулою  $\Delta t = \frac{t_{max} - t_{min}}{n}$ .
3. Заповнити таблицю даних:
  - комірки G4, H4, I4 повинні бути зв'язані з відповідними комірками таблиці вихідних даних, відповідно, C14, C12, C11 (спосіб копіювання);
  - у комірках G5, H5, I5 таблиці необхідно занести розрахункові формули і скопіювати у весь діапазон відповідних стовпчиків.
4. Відформатуйте числові дані у діапазоні комірок G4:I24.
5. Назвіть аркуш, наприклад, *Рівноприскорений рух*.
6. Побудувати графіки залежності  $v(t)$  та  $y(t)$ . Діаграми повинні мати: назви, підписи до осей, легенду. Зберегти дані.

### **Завдання**

1. Перевірити, чи впливає крок зміни часу на вигляд графіків.
2. Змінюючи початкові дані в комірках C11, C12, C13 і C15, простежити за змінами вигляду графіків. Який фізичний зміст негативних значень величин?
3. Порівняти рух тіл з різним прискоренням. Для цього необхідно:
  - провести вставку комірок над діапазоном A13 : D13 для задання другого значення прискорення ;
  - ввести в комірку C13 ще одне значення прискорення руху тіла, яке відрізняється від наявного, і оформити відповідним чином діапазон комірок B13 : D14;
  - додати дві колонки до таблиці (стовпчики J і K) і провести в них розрахунок швидкості і координати для другого значення прискорення;
  - для кожної з діаграм провести додавання нового ряду даних і задати ім'я кожному з двох рядів даної діаграми. В якості імені ряду зручно вибрати відповідне йому значення прискорення, ввівши в поле "Ім'я

ряду" посилання на діапазон B14 : D14 (перший ряд) і B13 : D13 (другий ряд даних);

- вивести легенду ;
- відформатувати дані в таблиці (діапазони H3 : H24 , I3 : I24 , J3 : J24 і K3 : K24 ) , призначивши колір шрифту в цих діапазонах відповідно до кольору графіків .

4. На окремому листі розв'яжіть графічно **задачу**: з пункту А виїхав вантажний автомобіль з сталою швидкістю 72 км/год. Одночасно з ним з пункту В, розміщеного на відстані 1,5 км, почав рухатись мотоцикліст. Вважаючи рух мотоцикліста рівноприскореним з прискоренням  $2 \text{ м/с}^2$ , визначити з допомогою відповідних графіків час, за який мотоцикліст дожене вантажівку, та шлях, пройдений кожним з них до моменту зустрічі (для точнішого визначення величини за графіком рекомендується включити основні та проміжні лінії сітки).

5. Зберегти дані.

## Лабораторна робота № 4

### Моделювання коливального руху на прикладі математичного маятника

#### Теоретичні відомості

Якщо тіло здійснює вільні незатухаючі коливання, то його координата, швидкість і прискорення з часом будуть змінюватися за законом косинуса або синуса, відповідно:

$$x = x_{max} \sin(\omega t + \varphi);$$
$$v_x = x_{max} \omega_0 \sin(\omega_0 t + \varphi + \frac{\pi}{2});$$
$$a_x = x_{max} \omega_0^2 \sin(\omega_0 t + \varphi + \pi).$$

Однією з найпростіших і поширених моделей коливальних систем є математичний маятник: матеріальна точка масою  $m$ , підвішена на нерозтяжній нитці довжиною  $L$  і здійснює коливання у вертикальній площині. Кутова частота коливань у цьому випадку дорівнює:

$$\omega_0 = \sqrt{\frac{g}{L}}.$$

Період коливань

$$T = \frac{2\pi}{\omega_0} = 2\pi \sqrt{\frac{L}{g}}.$$

Метою даної роботи є побудова графіків залежності  $x(t)$ ,  $a(t)$ ,  $v(t)$  та прослідкувати за їх змінами при різних параметрах системи.

#### Завантаження файлу таблиці

##### 1. Запустити Excel.

The screenshot shows a Microsoft Excel spreadsheet with the following content:

- Row 2:** Розрахункові формули:
- Row 3:**  $T = \frac{2\pi}{\omega_0} = 2\pi \sqrt{\frac{L}{g}}$        $v_x = x_{max} \omega_0 \sin(\omega_0 t + \varphi + \frac{\pi}{2})$
- Row 4:**  $x = x_{max} \sin(\omega t + \varphi)$        $a_x = x_{max} \omega_0^2 \sin(\omega_0 t + \varphi + \pi)$
- Row 5:** вихідні дані
- Row 6:** Амплітуда коливань  $x_{max}$  м
- Row 7:** Довжина маятника  $L$  м
- Row 8:** період коливань  $T$  с
- Row 9:** Часовий інтервал  $t_{min}$  с
- Row 10:**  $t_{max}$  с
- Row 11:** Крок  $\Delta t$  с
- Row 12:** Кутова частота  $\omega_0$  рад/с
- Row 13:** Початкова фаза  $\varphi$  град. =

On the right side of the spreadsheet, there is a table titled "Таблиця даних" with the following structure:

№	Час, с	Швидкість, м/с	Координата, м
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			

2. Створити *модель 3* (рис.), в якій вихідними даними будуть:

- амплітуда (максимальне зміщення)  $x_{max}$ ;
- початкова фаза  $\varphi$ ;
- довжина нитки  $L$ ;
- період коливань  $T$ ;
- часовий інтервал  $t_{min} \div t_{max}$  протягом якого розглядається рух.

*Змінні* величини: зміщення тіла в кожний момент часу (координата  $x$ ), швидкість і час; *постійні*: амплітуда, початкова, фаза, довжина нитки, період.

3. Зберегти файл під назвою *Лабораторна робота 3.xls*.

### **Заповнення таблиці**

1. Таблиця розрахована на 101 точку, тобто кількість підінтервалів, на які розбивається часовий інтервал, дорівнює 100.
2. Занести вихідні числові дані в комірки C10, C11, C14, C15 і C18. Початковий момент часу (комірка C14) зручно прийняти рівним нулю.
3. Ввести формули, які дозволяють розрахувати період, частоту коливань, а також інтервал часу  $\Delta t$  у комірки C12, C17 і C16, відповідно.
4. Провести перерахунок початкової фази з градусної міри в радіани (тригонометричні функції, які використовуються при розрахунках координати, швидкості і прискорення, обчислюють значення кута, заданого в радіанах) за допомогою вбудованої функції *РАДІАНИ* ():
  - виділити комірку E18;
  - викликати Майстер функцій;
  - у полі "Категорія" вибрати Математичні, в полі "Функція:" – РАДІАНИ. Дана функція перетворює градуси в радіани;
  - у полі "Кут" панелі формул ввести адресу комірки, що містить значення кута у градусах – C18.
5. Заповнити Таблицю даних (Час-Координата-Швидкість-Прискорення).
  - Стовпчик G повинен містити значення часу, що відрізняються на величину  $\Delta t$ , при цьому комірку G8 необхідно пов'язати з C14, щоб відстежувати зміни значення початкового моменту часу. У наступну комірку стовпчика вводиться формула, яка потім копіюється в діапазон G10:G108.
  - У стовпчиках H, I, J обчислюються значення координати, швидкості і прискорення у відповідні моменти часу з стовпчика G.

*Примітка:* При складанні формул не забувайте використовувати абсолютні посилання для постійних величин.
6. Змінити ім'я аркуша, що містить таблицю (Модель 3), наприклад, на *Коливання маятника*.

### ***Побудова графіків***

1. Побудувати графіки залежності  $x(t)$ ,  $v(t)$ ,  $a(t)$ . Оскільки у всіх випадках вздовж осі  $Ox$  відкладається одна і та ж величина (час), всі три залежності можна представити в одній системі координат. Діаграма повинна містити: назву, підписи до осей, легенду.
2. Зберегти дані.

### ***Завдання***

1. Змінюючи початкові дані (початковий зсув і фазу, довжину маятника), простежити за зміною вигляду графіків.
2. Розібрати питання про співвідношення величин  $x_{\max}$  і  $L$ .
3. Задати для діапазону G8:J108 відповідний формат числових даних
4. Змінюючи момент часу  $t_{\max}$  (і, тим самим, – інтервал  $\Delta t$ ), визначити межі застосування даної моделі.
5. Провести побудову кожного графіка окремо.
6. Зберегти дані.



## Лабораторна робота № 5

### Моделювання руху тіла, кинутого під кутом до горизонту

#### Теоретичні відомості

Розглянемо випадок руху тіла, кинутого під кутом до горизонту, що відбувається тільки під дією сили тяжіння (тертям нехтуємо). У цьому випадку однієї координати для опису руху недостатньо. Необхідно ввести систему координат  $xOy$ , при цьому вісь  $Ox$  спрямовують горизонтально, а вісь  $Oy$  – вертикально вгору або вниз. Тепер положення тіла задається двома координатами  $(x, y)$ , кожна з яких з часом буде змінюватися. Закон зміни координат можна встановити з наступних міркувань.

Оскільки ми вважаємо, що ніякі сили, крім сили тяжіння на тіло не діють, рух уздовж осі  $Ox$  буде рівномірним, і абсциса тіла змінюється за законом  $x = v_x t$ , де  $v_x = v_{0x} = \text{const}$  – проекція швидкості на вісь  $Ox$ .

Сила тяжіння, яка діє на тіло, надає йому прискорення  $\vec{g}$ , напрямленого, як і сама сила, вертикально вниз. Тому проекція швидкості на вісь  $Oy$  буде змінюватися за законом  $v_y = v_{0y} + g_y t$ , де  $v_{0y}$ ,  $g_y$  – проекція початкової швидкості і прискорення вільного падіння на цю вісь, а ордината тіла з часом змінюється як

$$y(t) = y_0 + v_{0y}t + \frac{g_y t^2}{2}.$$

Рівняння траєкторії, тобто залежність  $y(x)$ , можна знайти, виключивши час з останнього виразу. Виразимо час через абсцису:  $t = x/v_{0x}$  і підставимо в рівняння ординати:

$$y = y_0 + \frac{v_{0y}}{v_{0x}}x + \frac{g_y}{2} \cdot \frac{x^2}{v_{0x}^2},$$

де знаки проекцій  $v_{0x}$ ,  $v_{0y}$ ,  $g_y$  залежать від напрямку осей координат. У кожній точці траєкторії швидкість тіла спрямована по дотичній до неї і може бути розкладена на дві складові  $\vec{v} = \vec{v}_x + \vec{v}_y$ , де  $v_x = v_{0x}$ . Модуль швидкості при

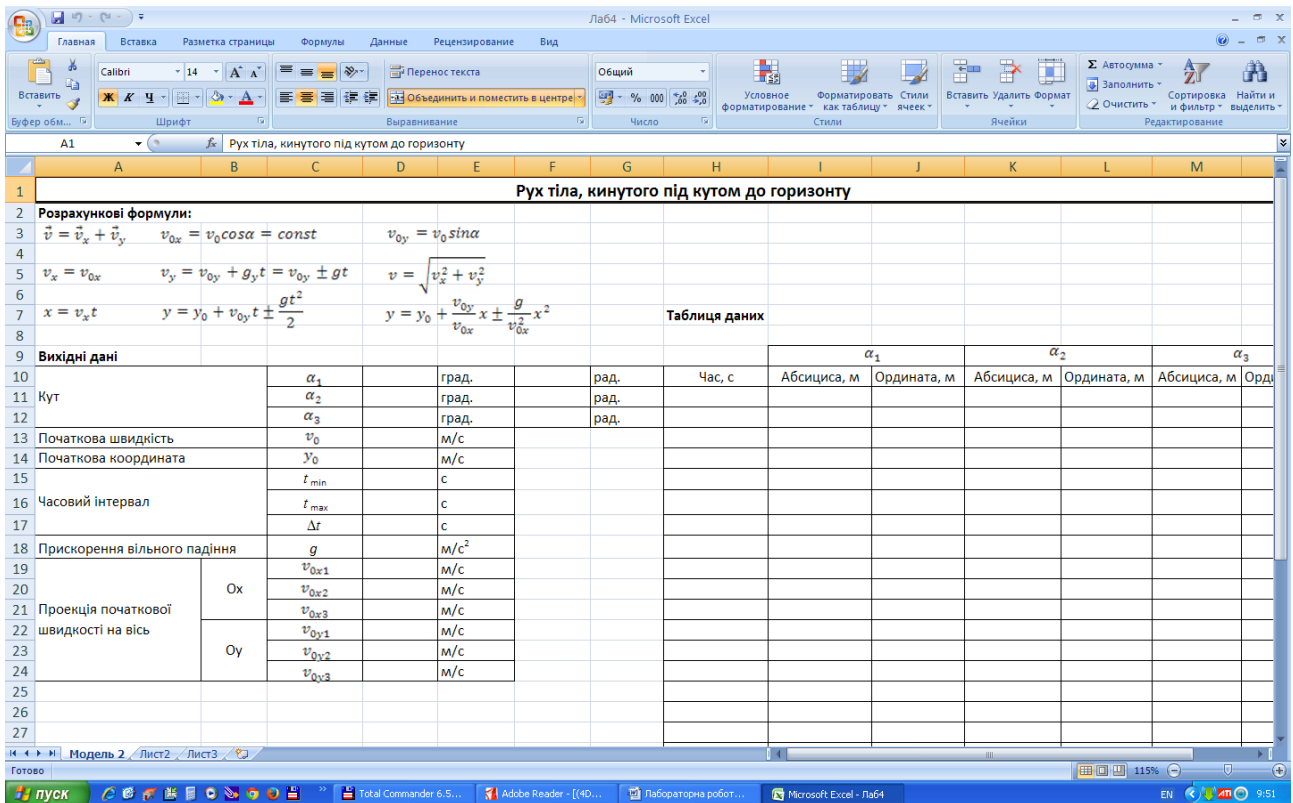
цьому  $v = \sqrt{v_{0x}^2 + v_y^2}$ .

У даній роботі можна простежити за взаємозв'язками наступних величин:  $x$  і  $t$ ,  $y$  і  $t$ ,  $y$  і  $x$ , а крім того, встановити, як залежить дальність польоту тіла від кута, під яким його кидають, і від величини його початкової швидкості.

#### Завантаження файлу таблиці

1. Запустити Excel.
2. Створити модель 4 (рис.).

Для побудови траєкторії руху тіла потрібні формули для розрахунку координати точки в різні моменти часу:



$$\begin{cases} x = v_x t = v_{0x} t \\ y = y_0 + v_{0y} t - \frac{g t^2}{2} \end{cases}$$

а також додаткові формули для обчислення проекцій початкової швидкості на осі координат  $Ox$  і  $Oy$ , переведення градусної міри кута в радіани, інтервалу часу  $\Delta t$ .

Проаналізуйте всі розглянуті вище формули і виділіть вихідні дані, змінні і постійні величини.

### Заповнення таблиці

1. Занести вихідні числові дані

(наприклад,  $\alpha_1 = 36^\circ$ ,  $\alpha_2 = 45^\circ$ ,  $\alpha_3 = 60^\circ$ ,  $v_0 = 100$  м/с,  $y_0 = 0$  м,  $t_0 = t_{min} = 0$  с,  $t_{max} = 12$  с).

в наступні комірки:

- кути (у градусах) між початковою швидкістю  $\vec{v}_0$  та віссю  $Ox$  – у комірки D15, D16, D17. Це дозволить простежити за залежністю траєкторії руху тіла від кута  $\alpha$ . Якщо значення кутів задані в радіанах, вони повинні бути занесені в комірки F15, F16, F17;
- модуль початкової швидкості – в D18;
- початкову ординату (висоту, з якої зроблений кидок) – у D19;
- кінцевий момент часу – в D21;
- прискорення вільного падіння  $g$  – у D23;

Початковий момент часу (комірка D20) повинен бути рівний нулю!

2. Якщо кути задані в градусах, необхідно виконати перерахунок їх в радіани. Формулу достатньо записати тільки в комірку F15, а потім скопіювати її в комірки F16 : F17.
3. У комірці D22 розрахувати крок зміни часу  $\Delta t$  для  $n = 50$  підінтервалів, на які розбивається обраний проміжок часу.
4. Відповідно з розрахунковими формулами, наведеними в робочому листі таблиці, визначити проекції початкової швидкості на осі  $Ox$  і  $Oy$  (комірки D24:D26; D27:D29).
5. Таблиця даних крім колонки *Час* містить три пари колонок *Абсциса* – *Ордината* для трьох заданих значень кута  $\alpha$ .
6. Побудувати графіки залежності  $y(x)$  для різних заданих кутів падіння в одній системі координат, що дозволить простежити вплив кута на вигляд траєкторії (висоту підняття і дальність польоту). Однак на відміну від *Лабораторної роботи № 3* вони не мають загального діапазону, залежно від якого і ведеться побудова всіх графіків. Тому побудову графіка необхідно починати для одного тільки ряду даних (наприклад, для I11 : J61), а потім на відповідному кроці провести додавання інших двох рядів. Кожному ряду необхідно присвоїти ім'я.
7. Зберегти дані у файлі Лабораторна робота 4.xls.

#### **Завдання**

1. Змінюючи початкові дані ( $y_0, v_0, t_{max}$ ), простежити за зміною траєкторії. Визначити, при якому значенні кута спостерігається найбільша дальність польоту.
2. Провести побудову графіків залежності  $y(t)$  для різних значень кута  $\alpha$  на одній діаграмі (всі три залежності мають загальний діапазон даних у стовпчику H). Дані для графіка беруться з створеної раніше таблиці.
3. Зберегти дані.

## Лабораторна робота № 6

### Розв'язування диференціального рівняння другого порядку чисельним методом. Коливання пружинного маятника

#### Теоретичні відомості

Розглядаючи коливання математичного маятника (див. Лабораторна робота № 3), ми користувалися готовими формулами, що дозволяють розрахувати зміщення, швидкість і координату тіла в будь-який момент часу. Опишемо кількісно процес коливань тіла під дією сили пружності (пружинний маятник). Аналогічно можна також розглянути і питання про коливання кульки, підвішеної на нитці (математичний маятник). Нехай вантаж масою  $m$ , підвішений на пружині жорсткістю  $k$ , здійснює коливання в горизонтальній площині під дією тільки сили пружності. Згасанням коливань нехтуємо. Згідно з другим законом Ньютона:

$$m\vec{a} = \vec{F}, \quad (5.1)$$

де  $m$  – маса тіла,  $\vec{F}$  – рівнодійна всіх сил, прикладених до тіла,  $\vec{a}$  – прискорення, надане тілу цією силою. Запишемо це рівняння для кульки, підвішеної на пружині і яка рухається прямолінійно вздовж горизонтальної осі. Вісь  $Ox$  спрямуємо вправо, і початок відліку координат сумістимо з положенням рівноваги кульки. Оскільки ми вирішили знехтувати згасанням коливань, то можна вважати, що на тіло діє тільки сила пружності, яка і надає тілу прискорення. У проекції на обраний напрям рівняння руху перепишеться у вигляді:

$$ma_x = F_x, \quad (5.2)$$

де  $a_x$  – проекція прискорення на вісь  $Ox$ ,  $F_x$  – проекція сили пружності на ту ж вісь. Вона прямо пропорційна зсуву тіла з положення рівноваги, причому проекція сили і координата мають протилежні знаки (тому сила пружності завжди протилежна до зміщення тіла з положення рівноваги):  $F_x = -kx$ , де  $x$  – координата тіла (зміщення). Отже, рівняння (5.2) набере вигляду:

$$ma_x = -kx. \quad (5.3)$$

Це і є рівняння руху кульки під дією сили пружності. Перепишемо його в іншому вигляді. Виразимо проекцію прискорення:

$$a_x = -\frac{k}{m}x$$

і врахуємо, що проекція прискорення на вісь  $Ox$  є не що інше, як друга похідна координати  $x$  за часом.

Отже

$$x'' = \frac{d^2x}{dt^2} = -\frac{k}{m}x \quad (5.4)$$

або

$$x'' = \frac{d^2x}{dt^2} = -\omega_0^2 x, \quad (5.4 \text{ а})$$

де величина  $\omega_0 = \sqrt{k/m}$  носить назву власної циклічної або кутової частоти коливальної системи. Таким чином, рівняння коливань маятника має вигляд:

$$x'' + \omega_0^2 x = 0$$

Точний розв'язок цього рівняння має вигляд:

$$\begin{cases} x = x_{max} \sin(\omega_0 t + \varphi_0) \\ v_x = \omega_0 x_{max} \sin(\omega_0 t + \varphi_0 + \frac{\pi}{2}) \\ a_x = \omega_0^2 x_{max} \sin(\omega_0 t + \varphi_0 + \pi) \end{cases}$$

Ці формули дозволяють знайти величини  $x$ ,  $v_x$ ,  $a_x$  у будь-який момент часу. Оскільки рух тіла відбувається під дією змінної сили (а, отже, не є рівноприскореним), завдання це вкрай складне.

Для розв'язання таких завдань використовують *чисельний метод*, який є наближенням, але за певних умов дозволяє отримати достатньо хороші результати.

Розглянемо суть так званого *методу половинного інтервалу*.

1. Вибирається проміжок часу  $t_{min} \div t_{max}$  (як правило,  $t_{min} = 0$ ), який розбивають на  $n$  однакових підінтервалів  $\Delta t = \frac{t_{max} - t_{min}}{n}$ . Кількість таких підінтервалів вибирається довільно. Наближення на цьому кроці полягає в тому, що рух на кожному з них вважається рівноприскореним. Незавжди бачити, що в цьому випадку саме величина  $\Delta t$  буде визначати точність наших обчислень: чим менший цей інтервал, тим менше будуть відрізнятися шукані значення від істинних. Крім того, інтервал  $\Delta t$  (крок програми), можна вибрати довільно, не зазначаючи проміжок часу  $t_{min} \div t_{max}$ .

2. Перейдемо тепер від точного рівняння (5.4) до наближеної рівності, скориставшись визначенням похідної функції  $x(t)$  у точці:

$$x' = \frac{dx}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t} \approx \frac{\Delta x}{\Delta t}. \quad (5.5)$$

У (5.5)  $\Delta t = t_n - t_{n-1}$ . У загальному випадку можна записати  $\Delta t = t_{i+1} - t_i$ , де індекс  $i$  пробігає всі значення від 0 до  $n$  ( $i = 0, 1, 2, \dots, n$ ). Аналогічно:  $\Delta x = x_{i+1} - x_i$ . Враховуючи, що  $x' = v_x$ ,  $x'' = v'_x$ , розпишемо за тим же принципом і другу похідну координати за часом:

$$x'' = \frac{d^2x}{dt^2} = \frac{dv_x}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta v_x}{\Delta t} \approx \frac{\Delta v_x}{\Delta t}. \quad (5.6)$$

З (5.5) і (5.6) видно, що чим менша величина інтервалів, на які ми розбиваємо обраний проміжок часу, тим точніше наближення.

3. Отже, суть методу половинного інтервалу полягає в тому, що

1) значення швидкості розраховуються в середині кожного інтервалу  $\Delta t$ , тобто в моменти часу

$$t_{1/2} = \frac{t_1 - t_0}{2} = \Delta t/2,$$

$$t_{3/2} = \frac{t_2 - t_1}{2} = 3 \cdot \Delta t/2,$$

...

$$t_{i+1/2} = \frac{t_{i+1} - t_i}{2} = (2i + 1) \cdot \Delta t/2,$$

...

$$t_{n-1/2} = \frac{t_n - t_{n-1}}{2} = (2n - 1) \cdot \Delta t/2,$$

за формулою, яку можна отримати, підставивши (5.6) в (5.4):  $\frac{\Delta v_x}{\Delta t} \approx -\frac{k}{m}x$ ;

2) значення координати обчислюються в кінці кожного інтервалу часу, тобто в точках розбиття:  $t_1, t_2, \dots, t_{i-1}, t_i, \dots, t_n$  осі часу за формулою (5.5). Таким чином, ми отримуємо, що точки, в яких обчислюються значення координати і швидкості, зсунуті одна відносно одної на половину інтервалу, тобто на величину  $\Delta t/2$ . Самі ж значення  $x$  і  $v_x$  обчислюють через інтервал  $\Delta t$ .

Тоді значення проекції швидкості в середині першого інтервалу знайдемо з умови:

$$\frac{(v_x)_{1/2} - (v_x)_0}{(\Delta t/2)} \approx -\frac{k}{m}x_0,$$

де індекси біля координати і швидкості позначають момент часу, в який вони обчислюються. Таким чином

$$(v_x)_{1/2} \approx (v_x)_0 - \frac{k}{m}x_0 \cdot \frac{\Delta t}{2}.$$

У наступні моменти часу:

$$(v_x)_{3/2} \approx (v_x)_{1/2} - \frac{k}{m}x_1 \cdot \Delta t;$$

$$(v_x)_{5/2} \approx (v_x)_{3/2} - \frac{k}{m}x_2 \cdot \Delta t;$$

...

$$(v_x)_{i+1/2} \approx (v_x)_{i-1/2} - \frac{k}{m}x_i \cdot \Delta t; \dots$$

Таким чином, швидкість тіла в деякий момент часу визначається через відомі значення координати і швидкості у попередні моменти часу. Координату тіла в момент часу  $t_1$  знайдемо з умови:

$$x' = v_x \approx \frac{\Delta x}{\Delta t}.$$

А

$$(v_x)_{1/2} \approx \frac{x_1 - x_0}{\Delta t},$$

звідки

$$x_1 \approx x_0 + (v_x)_{1/2} \cdot \Delta t.$$

У наступні моменти часу

$$x_2 \approx x_1 + (v_x)_{3/2} \cdot \Delta t;$$

$$x_3 \approx x_2 + (v_x)_{5/2} \cdot \Delta t;$$

...

$$x_{i+1} \approx x_i + (v_x)_{2i+1/2} \cdot \Delta t,$$

де  $i = 0, 1, 2, \dots, n$ .

Узагальнюючи все вище сказане, можна записати основні рівняння, які являють собою розв'язок диференціального рівняння коливань:

$$\begin{cases} (v_x)_{1/2} \approx (v_x)_0 - \frac{k}{m} x_0 \cdot \frac{\Delta t}{2} \\ (v_x)_{i+1/2} \approx (v_x)_{i-1/2} - \frac{k}{m} x_i \cdot \Delta t \\ x_{i+1} \approx x_i + (v_x)_{2i+1/2} \cdot \Delta t \end{cases}$$

Де  $i = 0, 1, 2, \dots, n$ . Ці формули дозволять скласти таблицю даних для побудови графіків залежностей  $x(t)$ ,  $v(t)$ .

Зверніть увагу, що координата тіла в будь-який момент часу  $t_i$  знаходиться за загальною формулою, тоді як швидкість в середині першого інтервалу визначається за окремою формулою. У подальших розрахунках ця формула не використовується.

### Завантаження файлу таблиці

1. Запустити Excel.
2. Створити модель 5 (рис.).

The screenshot shows an Excel spreadsheet titled "Ла65 - Microsoft Excel". The main content is a table for calculating the motion of a spring oscillator. The table is structured as follows:

Розв'язування диференціального рівняння другого порядку чисельним методом. Коливання пружинного маятника					
Розрахункові формули:					
$\begin{cases} (v_x)_{1/2} \approx (v_x)_0 - \frac{k}{m} x_0 \cdot \frac{\Delta t}{2} \\ (v_x)_{i+1/2} \approx (v_x)_{i-1/2} - \frac{k}{m} x_i \cdot \Delta t \\ x_{i+1} \approx x_i + (v_x)_{2i+1/2} \cdot \Delta t \end{cases}$			Таблиця даних		
№	Час, с	Швидкість, м/с	Час, с	Координата, м	Координата, м
1					
2					
3					
4					
5					
6					
7					
8					
9					
Вихідні дані					
11	Маса тіла	$m$	кг		
12	Жорсткість пружини	$k$	Н/м		
13		$t_{\min}$	с		
14	Часовий інтервал	$t_{\max}$	с		
15		$\Delta t$	с		
16	Початкове зміщення	$x_0$	м		
17	Початкова швидкість	$v_{0x}$	м/с		
18	Кутлова швидкість	$\omega_0$	рад/с		
19	Період коливань	$T$	с		
20	Початкова фаза	$\varphi_0$	рад		
21					
22					
23	Амплітуда коливань	$X_{\max}$	м		

3. Проаналізувати розрахункові формули, виділити вихідні дані, змінні і постійні величини.

### **Заповнення таблиці**

1. Занести вихідні числові дані ( $m = 1$  кг,  $k = 5$  Н/м,  $t_{\max} = 10$  с,  $v_0 = 3$  м/с  $x_0 = 0$  м) у наступні комірки:

- масу тіла – в D11;
- жорсткість пружини – в D12;
- кінцевий момент часу – в D14;
- початковий зсув – в D16;
- початкову швидкість – у D17.

Початковий момент часу при розв'язуванні рівнянь за методом половинного інтервалу вибирають рівним нулю:  $t_0 = 0$  (комірка D13).

2. Ввести формули, які дозволяють розрахувати інтервал часу  $\Delta t$  (крок програми), власну циклічну частоту і період коливань в комірки D15, D18 і D19.

3. Заповнити *Таблицю даних* (Час – Швидкість; Час – Координата).

- Столпчики G і I містять значення часу. Введення двох колонок часу необхідне, оскільки обчислення швидкості і координати проводиться в різні моменти часу. Так, в комірки G6 і I6 копіюється початковий момент часу (комірка D13), після чого в клітинку I7 заноситься формула для обчислення моменту часу  $t_1$ , яка відразу копіюється в нижчий діапазон комірок; а в комірку G7 вводиться проміжна формула для обчислення моменту  $t_{1/2}$ , віддаленого від початкового на величину  $\Delta t/2$ . Подальша зміна часу у колонці G також має відбуватися через інтервал  $\Delta t$ .
- У столпчиках H і J містяться значення швидкості і координати. Заповнення цих столпчиків проводиться аналогічно. Це означає, що при заповненні колонки Швидкість у комірку H7 вводиться проміжна формула, що обчислює швидкість у середині першого інтервалу, тобто через інтервал  $\Delta t/2$  після початку відліку часу. Подальше обчислення швидкості ведеться за загальною формулою через крок  $\Delta t$ . У комірку J7 вводиться загальна формула для обчислення координати у відповідні моменти часу, яка потім копіюється в нижчі комірки. Зверніть увагу на те, що в обчисленнях швидкості і координати не використовуються значення часу з колонок Час! Вони потрібні тільки для побудови графіків.

*Примітка:* При складанні формул не забувайте використовувати абсолютні посилання для постійних величин.



4. Змінити ім'я аркуша, що містить таблицю (Модель 5), наприклад, на *Коливання пружинного маятника*.

### ***Побудова графіків***

1. Побудувати графіки залежності  $x(t)$ ,  $v(t)$ ,  $a(t)$ . Оскільки у всіх випадках вздовж осі Ох відкладається одна і та ж величина (час), всі три залежності можна представити в одній системі координат. Діаграма повинна містити: назву, підписи до осей, легенду.
2. Зберегти дані.

### ***Завдання***

1. Отримати самостійно формулу для обчислення прискорення (користуючись при цьому тільки визначенням прискорення і введеним спрощенням, що рух між двома сусідніми моментами часу можна вважати рівноприскореним), провести розрахунок в стовпчику М і побудувати графік залежності прискорення від часу на тій же діаграмі.
2. Змінюючи масу тіла і жорсткість пружини, простежте за зміною величин, що характеризують коливальний процес: кутовою частотою, періодом, амплітудою координати і швидкістю.

## Лабораторна робота № 7

### Моделювання падіння тіла з урахуванням опору середовища

#### Теоретичні відомості

У реальному світі на тіла, які падають в атмосфері діє сила тертя. Для невеликих тіл, які падають з малою швидкістю, сила тертя, як показує досвід, пропорційна швидкості ( $kv$ ). Для великих тіл, що рухаються з великою швидкістю, сила тертя пропорційна квадрату швидкості ( $f v^2$ ), де  $k$  і  $f$  – коефіцієнти пропорційності. Згідно з другим законом Ньютона

$$m\vec{a} = \vec{F}_{\text{тяж}} + \vec{F}_{\text{тер}}.$$

Складемо диференціальне рівняння руху тіла з урахуванням сили тертя. Другий закон Ньютона можна записати як

$$m \cdot y'' = m \frac{d^2 y}{dt^2} = F_y \quad \text{або} \quad y'' = \frac{d^2 y}{dt^2} = \frac{F_y}{m},$$

де  $F_y$  – проекція рівнодійної сил, що діють на тіло, на обраний напрям. Нехай позитивний напрям осі  $Oy$  збігається з напрямом сили тяжіння, а початок відліку – з початковим положенням тіла. Виразимо результуючу силу  $F_y$  як різницю між силою тяжіння і силою тертя ( $f(y')^2 = f v$ ). Отримаємо диференціальне рівняння

$$y'' = \frac{d^2 y}{dt^2} = \frac{mg_y - f \left(\frac{dy}{dt}\right)^2}{m}$$

де  $f$  – коефіцієнт пропорційності, який залежить від об'єму та форми тіла, а також від в'язкості середовища, в якій відбувається падіння. Це рівняння зручніше переписати у вигляді

$$\frac{d^2 y}{dt^2} = g - \alpha \left(\frac{dy}{dt}\right)^2, \quad \text{де } \alpha = \frac{f}{m}. \quad (6.1)$$

Оскільки швидкість тіла змінюється, рух відбувається під дією змінної сили, тому для розв'язання цього рівняння використовується чисельний метод половинного інтервалу. Скористаємося алгоритмом, викладеним у *Лабораторній роботі № 5*, і складемо систему рівнянь для визначення швидкості і координати тіла. З його допомогою можна отримати наступну систему:

$$\begin{cases} \left(\frac{dy}{dt}\right)_{1/2} \approx \left(\frac{dy}{dt}\right)_0 + \left(g - \alpha \left(\frac{dy}{dt}\right)_0^2\right) \cdot \frac{\Delta t}{2} \\ \left(\frac{dy}{dt}\right)_{i+1/2} \approx \left(\frac{dy}{dt}\right)_{i-1/2} + \left(g - \alpha \left(\frac{dy}{dt}\right)_{i-1/2}^2\right) \cdot \Delta t \\ y_{i+1} \approx y_1 + \left(\frac{dy}{dt}\right)_{i+1/2} \cdot \Delta t \end{cases} \quad (6.2)$$

Проаналізуємо задачу. Рух під дією зазначених сил, очевидно, буде відбуватися зі зростаючою швидкістю і прискоренням, яке зменшується, так що з часом, коли прискорення стане рівним нулю, швидкість падіння тіла перестане

змінюватися. Швидкість руху, що встановиться легко оцінити з таких міркувань: прискорення обертається в нуль, коли сила опору врівноважує силу тяжіння, тобто

$$fv_y^2 = mg.$$

Звідси легко отримати:

$$v_{\text{теор}}^{\text{вст}} = \sqrt{\frac{mg}{f}} = \sqrt{\frac{g}{\alpha}} \quad (6.3)$$

Умова (6.3) може служити критерієм тривалості розрахунків: ми не можемо вказати, починаючи з якого моменту часу рух тіла стане рівномірним (тобто задати часовий інтервал  $t_{\min} \div t_{\max}$ ), отже, обчислення необхідно проводити доти, поки значення швидкості, отримане згідно формули (6.2) не буде відповідати теоретичному значенню (6.3) з певною точністю, що задається самим користувачем.

### Завантаження файлу таблиці

1. Запустити Excel.
2. Створити *модель 6* (рис.. Тут наведено дві таблиці для різних  $\alpha$ , кожна з

The screenshot shows an Excel spreadsheet with the following structure:

- Row 3:** "Вихідні дані" (Input data) and "Розрахункові формули:" (Calculation formulas).
- Row 5-12:** Input data table with columns for parameter name and units.
- Row 5-12:** Calculation formulas for velocity and position at various time steps.
- Row 16-17:** Summary table for "Таблиця 1" and "Таблиця 2" showing velocity and coefficient alpha.
- Row 19-23:** Simulation results tables for "Таблиця 1" and "Таблиця 2" with columns for time, velocity, and position.

яких, у свою чергу, виконана для двох різних інтервалів  $\Delta t$ . Така побудова дозволяє відразу простежити залежності, по-перше, характеру руху тіла від коефіцієнта  $\alpha$ ; і, по-друге, точності розв'язування від кроку програми.

### Заповнення таблиці

3. Занести вихідні числові дані

$t_0 = 0$  с,  $y_0 = 0$  м,  $(v_y)_0 = 0$  м/с,  $\alpha_1 = 0,03$  м<sup>-1</sup>,  $\alpha_2 = 0,1$  м<sup>-1</sup>,  $\Delta t_1 = 0,1$  с,  $\Delta t_2 = 1$  с.

у наступні комірки:

- у E5: початковий момент часу (він завжди дорівнює нулю);
- у E6: початкову координату (дорівнює 0);
- у E7: початкову швидкість;
- у E8 і E9: коефіцієнти  $\alpha$ , що відрізняються один від одного на порядок або на два (для повітряного середовища  $\alpha < 1$ );
- у E10: прискорення вільного падіння;
- у E12 і E13: крок програми. Оскільки в даному випадку неможливо вказати момент часу  $t_{\max}$ , починаючи з якого тіло буде рухатися рівномірно, таблиця виконана для 200 значень, проте при необхідності її можна продовжити.

2. Розрахувати швидкість руху, що встановився  $v_{\text{теор}}^{\text{вст}}$  в комірках H16 і R16 – вона буде служити критерієм припинення обчислень.

3. Заповнити *Таблицю № 1* у такій послідовності:

- скопіювати
  - 1) у комірку H17 – вміст комірки E8 (коефіцієнт  $\alpha_1$ );
  - 2) у комірку R17 – вміст комірки E9 (коефіцієнт  $\alpha_2$ );
  - 3) у комірку D19 – крок програми  $\Delta t_1$  з комірки E12;
  - 4) у комірку H19 – крок програми  $\Delta t_2$  з комірки E13;
  - 5) у комірки B21, D21, F21, H21 – початковий момент часу з E5;
  - 6) у комірки C21 і G21 – початкову швидкість з E7;
  - 7) у комірки E21 і I21 – початкову координату з E6 .

Всі копіювання виконуються з встановленням зв'язку між відповідними комірками;

- провести обчислення швидкості тіла у колонці *Швидкість*, м/с, яка відповідає колонці C таблиці, в різні моменти часу у відповідності з розрахунковими формулами, тобто :
  - 1) у клітинці C22 розрахувати швидкість в середині першого інтервалу часу;
  - 2) у нижчих комірках даної колонки таблиці розрахувати швидкість в інші моменти часу (тобто через інтервал  $\Delta t$ ). Оскільки обчислення повинні проводитися тільки до моменту встановлення постійної швидкості (див. критерій припинення обчислень), задамося деякою точністю, наприклад, в 0,01 (це значення попередньо вводиться в комірку E11). Це означає, що обчислення проводяться до тих пір, поки одержувані значення не будуть відрізнятися від теоретичного менш ніж

на одну соту, тобто значення швидкості, які обчислюють за основною розрахунковою формулою, повинні постійно порівнюватися із значенням, отриманим аналітично. Такий спосіб побудови формули називається обчисленням за умовою і організовується за допомогою вбудованих логічних функцій (ЯКЩО, І, АБО, НЕ).

Остаточний вигляд формули в комірці C23:

= ЯКЩО(АБО(\$H\$16-C22<\$E\$11; C22=0); 0; C22+(\$E\$10-\$H\$17\*C22^2)\*\$D\$19).

У результаті, починаючи з деякого моменту часу, у всіх комірках стовпчика С буде виводитися значення 0, яке свідчить про те, що швидкість тіла практично перестала змінюватися.

Скопіювати створену формулу в діапазон комірок C24:C220;

- аналогічним чином заповнити колонку *Швидкість, м/с* у стовпчику G таблиці № 1 для іншого кроку програми;
- обчислення координати можна організувати за допомогою логічної функції ЯКЩО, тому що при цьому використовується тільки одна логічна умова, а саме – перевірка на 0 значення швидкості у відповідній комірці:  
= ЯКЩО (C22=0, 0, E21+C22\*\$E\$12). З формули видно, що при виконанні умови C22=0, комірці E22 буде присвоєно значення 0, у протилежному випадку обчислюється координата у відповідний момент часу. Скопіювати формулу в діапазон комірок E23:E220;
- аналогічним чином заповнити колонку *Координата, м* у стовпчику I таблиці для іншого кроку програми;
- заповнити колонки *Час, с* для швидкості (стовпчики В і F таблиці), враховуючи, що перше значення швидкості обчислюється в середині першого інтервалу часу. Надалі час змінюється через інтервал  $\Delta t$ . Скопіювати формули в діапазони клітинок B24:B220 і F24:F220;
- заповнити колонки *Час, с* для координати (у стовпчиках D і H). У цьому випадку час змінюється через інтервал  $\Delta t$ , тому що обчислення координати проводиться в точках розбиття осі часу (наприкінці кожного інтервалу).

4. Самостійно заповнити *Таблицю 2*.

### ***Побудова графіків***

1. За результатами розрахунків побудувати дві діаграми:  $v(t)$  і  $y(t)$ , кожна з яких містить 4 графіки (для двох різних  $\alpha$  і двох інтервалів  $t$ ). Діаграми будуються на окремих аркушах. Тип і вид діаграми необхідно вибрати Точкова (без відрізків, що з'єднують експериментальні точки).
2. Зберегти дані.

### *Завдання*

1. Змінюючи коефіцієнти  $\alpha$ , простежити за зміною швидкості рівномірного руху.
2. Простежити за впливом на точність розрахунків кроку програми. Визначити, за яких умов модель перестане працювати.

**Лабораторна робота №8**  
**Основні поняття і принципи роботи в Autocad.**  
**Підготовка робочого середовища**

***Завдання***

1. Запустіть програму Autocad.
2. Розгляньте вкладки Files, Display, Open and Save, Plotting, System, User Preferences, Drafting, Selection, Profiles, Customize, Toolbars, Commands, Properties, Keyboard.
3. Розгляньте засоби організації креслення: фіксовану світову систему координат і рухому систему координат користувача; одиниці вимірювання, шари; графічні примітиви; введення команд та даних.
4. Використайте команди системи Autocad для створення робочого середовища: Units, Limits, Layer, Style, Dimstyle, Setvar.
5. Скористайтесь командою встановлення режимів креслення Dsettings; командою перегляду параметрів робочого середовища Status; командами керування зображенням на екрані: Pan, redraw, Regen, Regenauto, Zoom, Ucs.
6. Виконайте наступні тестові завдання:
  1. AutoCAD – це програма векторної графіки, потужна система конструювання за допомогою комп'ютера:
    - а) правильно;
    - б) не правильно.
  2. Програму AutoCAD розроблено фірмою:
    - а) AutoNesk;
    - б) AutoLuks;
    - в) AutoDesign;
    - г) AutoDesk;
    - д) AutoCAD.
  3. На якому пристрої зберігається створене креслення?
    - а) Plotter;
    - б) Printer;
    - в) Keyboard;
    - г) Hard drive;
    - д) Display monitor.
  4. Яке розширення мають файли креслення?
    - а) .dwg;
    - б) .tmp;
    - в) .doc;
    - г) .dwt;
    - д) .bak.

5. В якому меню знаходиться команда, що дозволяє розпочати роботу з новим кресленням?

- а) Tools;
- б) Edit;
- в) File;
- г) Format;
- д) Options.

6. Яка максимальна кількість символів може бути в імені фала?

- а) 127;
- б) 255;
- в) 32;
- г) 8;
- д) 64.

7. Яка команда дозволяє встановити формат і точність подання лінійних і кутових одиниць?

- а) Format;
- б) Units;
- в) Options;
- г) Drawing ;
- д) Length.

8. Яка команда визначає розмір аркуша паперу?

- а) Grid;
- б) Paper;
- в) Drawing Limits;
- г) Settings;
- д) Utility.

9. Яка команда забезпечує перегляд частини креслення, яка не вмістилася на екрані монітора, без зміни масштабу?

- а) Scale;
- б) Pann;
- в) View;
- г) Realtime;
- д) Pan.

10. Яка команда збільшує або зменшує зображення об'єкта на екрані монітора без зміни фактичних розмірів?

- а) Zoo;
- б) View;
- в) Layer;
- г) Zoom;



д) Window.

11. Яка команда дозволяє визначити на кресленні власні системи координат користувача?

а) Top;

б) Botton;

в) Grid;

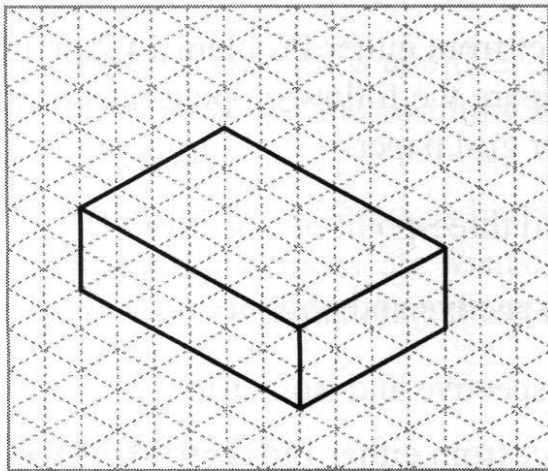
г) Ucs;

д) Face.

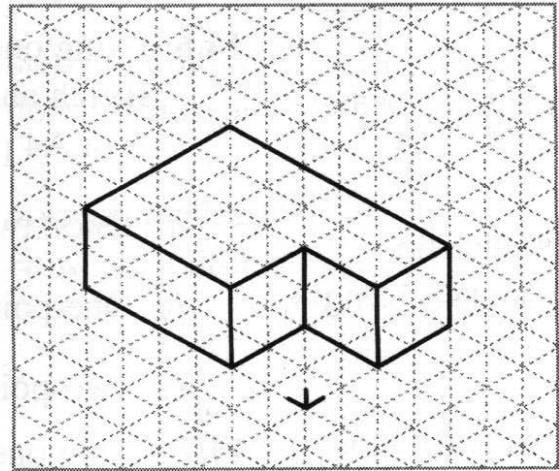
**Лабораторна робота №9**  
**Засоби креслення. Засоби редагування креслень**

***Завдання***

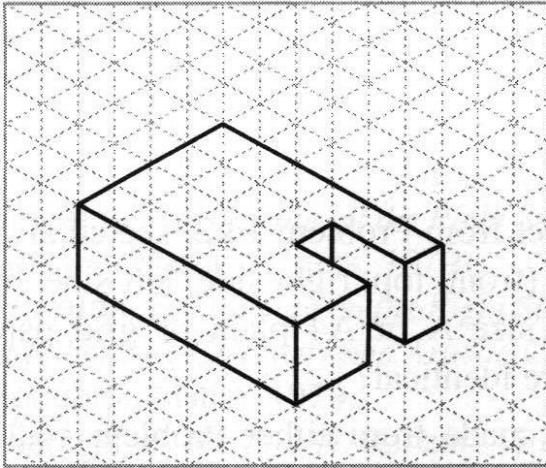
1. Розгляньте команди побудови ліній: Line, Pline, Spline, Ray, Xline.
2. Розгляньте команди побудови багатокутників, кіл, еліпсів: Polygon, Rectang, Arc, Circle, Ellipse.
3. Розгляньте команди генерації тексту: Text, Mtext.
4. Розгляньте команди нанесення штрихувань: Hatch, Bhatch, Booundary.
5. Розгляньте команди редагування креслень: Properties, Align, Array, Break, Chamfer, Copy, Erase, Explode, Fillent, Trim, Undo.
6. Розгляньте команди: редагування поліліній Pedit; штрихувань Hatchedit; тексту Ddedit, Scaletext, Spacetrans.
7. Використовуючи команди графічних примітивів середовища AutoCAD, побудуйте три основні вигляди вибраної деталі.



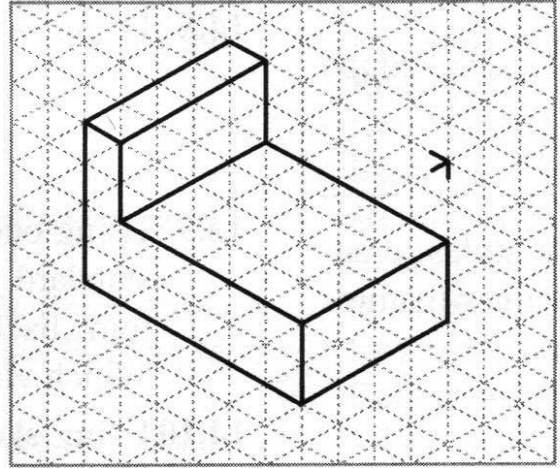
**1**



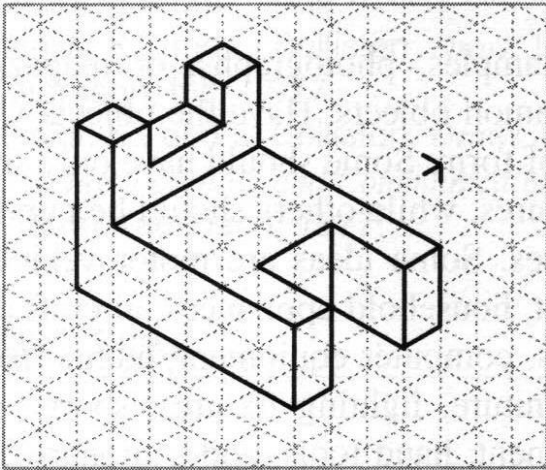
**2**



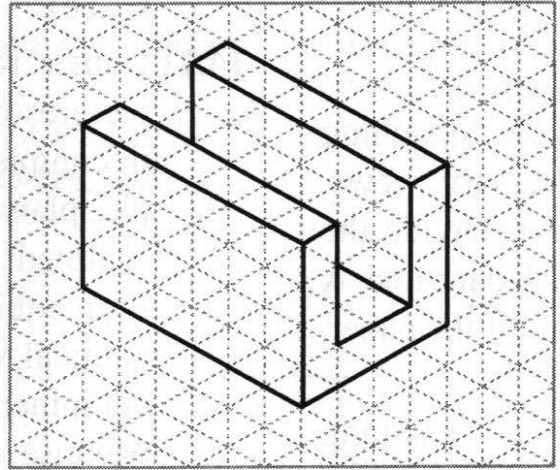
3



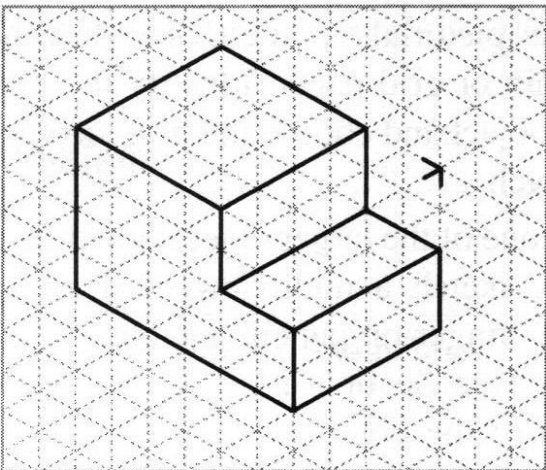
4



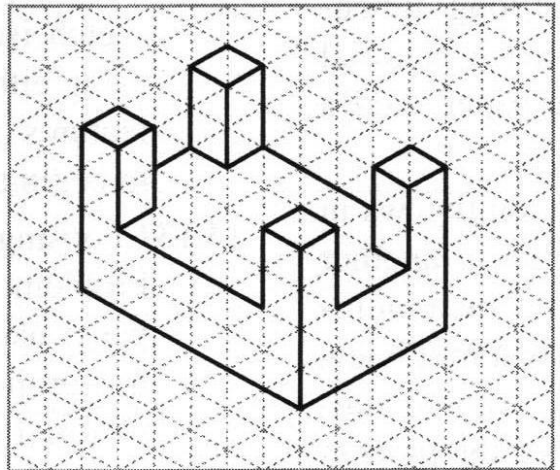
5



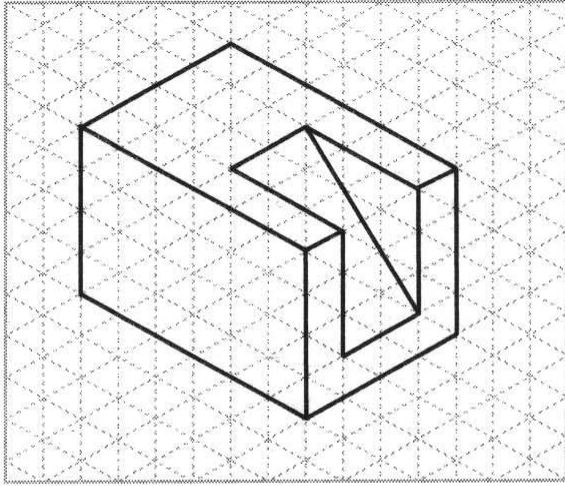
6



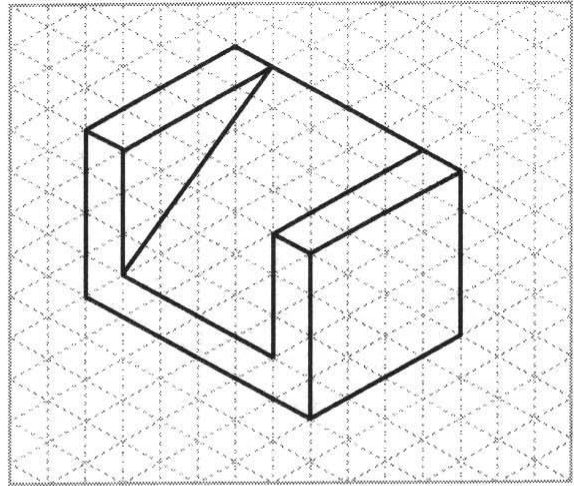
7



8



**9**



**10**

8. Виконайте тести:

1. Яка команда розділяє складені об'єкти?

- а) Break;
- б) Load;
- в) Explode;
- г) Divide;
- д) Modify.

2. Яка команда подовжує вказаний графічний об'єкт?

- а) Save;
- б) Trim;
- в) Extend;
- г) Box;
- д) Layer.

3. Яка команда здійснює спряження дугою заданого радіуса?

- а) Ray;
- б) Chamfer;
- в) Extrude;
- г) Fillet;
- д) Exit.

4. Яка команда дозволяє змінити довжину відкритих об'єктів?

- а) Select;
- б) Copy;
- в) Oops;
- г) Offset;
- д) Lengthen.

5. Яка команда дозволяє створювати подібні об'єкти?

- а) Select;

- б) Copy;
- в) Oops;
- г) Offset;
- д) Lengthen.

## Лабораторна робота №10

### Нанесення розмірів на креслення. Додаткові засоби формування креслень

#### Завдання

1. Розгляньте команди нанесення розмірів: Dimlinear, Dimaligned, Dimbaseline, Dimcontinue, Dimangular, Dimdiameter, Dimradius, Qleader, Tolerance, Qdim.
2. Розгляньте команди редагування розмірного блоку: Dimedit, Dimtedit, Dimcenter, Dimoverride.
3. Розгляньте команду редагування розмірних стилів Properties.
4. Використайте додаткові засоби формування креслення, зокрема команди: Block, Wblock, Base, Attdef, Eattedit, Battman.
5. Виконайте тести:
  1. Яка команда використовується для визначення розмірів відрізків прямих ліній і прямокутних сегментів поліліній або блоків?
    - a) Aligned;
    - б) Enter;
    - в) Linear;
    - г) Origin;
    - д) Orto.
  2. Яка команда дозволяє будувати лінійний розмір, розмірна лінія якого паралельна відріжку?
    - a) Aligned;
    - б) Enter;
    - в) Linear;
    - г) Origin;
    - д) Orto.
  3. Яка команда дозволяє будувати розмірну лінію для позначення кута між двома непаралельними лініями?
    - a) Aligned;
    - б) Enter;
    - в) Linear;
    - г) Origin;
    - д) Angular.
  4. Яка команда створює блок?
    - a) Save Block;
    - б) Make Block;
    - в) Insert Block;
    - г) Definition Block;
    - д) Open Block.

5. Яку команду треба використати для поділу відрізка на п'ять рівних частин?
- а) Measure;
  - б) Divide;
  - в) Break;
  - г) Change;
  - д) Polyline.
6. Яку команду треба використати для поділу відрізка на частини довжиною по 5 см?
- а) Measure;
  - б) Divide;
  - в) Break;
  - г) Change;
  - д) Polyline.
7. Для створення множини подібних ліній можна застосовувати команду Offset до поліліній:
- а) правильно;
  - б) не правильно.
8. Виберіть команду, яка дозволяє стиснути чи розтягнути об'єкт в одному напрямку:
- а) Extend;
  - б) Trim;
  - в) Stretch;
  - г) Move;
  - д) Edit.
9. За допомогою якої команди можна швидко побудувати 16 отворів відносно єдиного центру?
- а) Rectangular array;
  - б) Multiple copy;
  - в) Copy circular;
  - г) Repeat;
  - д) Polar array.

**Лабораторна робота №11**  
**Знайомство з режимом "прямих обчислень"**  
**пакета прикладних програм MATLAB**

*Теоретичні відомості*

**1. Основні відомості про правила роботи в режимі прямих обчислень.**

- 1) Для вказівки точки введення початкових даних в MATLAB використовується символ >>.
- 2) Дані вводяться за допомогою найпростішого рядкового редактора.
- 3) Для блокування виведення результату обчислень деякого виразу після нього треба вставити знак « ; » (крапка з комою).
- 4) Якщо не вказана змінна для значення результату обчислень, то MATLAB призначає таку змінну з ім'ям «ans».
- 5) знаком привласнення є звичний математикам знак рівності «=».
- 6) результат обчислень виводиться в рядках виведення (без знака >>);
- 7) діалог відбувається в стилі «задав питання – отримав відповідь».

Таблиця 10.1

**Операції рядкового редагування**

<b>Комбінація клавiш</b>	<b>Призначення</b>
→ або Ctrl+b	Переміщення курсору вправо на один символ
← або Ctrl+f	Переміщення курсору вліво на один символ
Ctrl+→ або Ctrl+r	Переміщення курсору вправо на одне слово
Ctrl+← або Ctrl+l	Переміщення курсору вліво на одне слово
Ctrl+← або Ctrl+l	Переміщення курсору вліво на одне слово
Home або Ctrl+a	Переміщення курсору на початок рядка
End або Ctrl+e	Переміщення курсору в кінець рядка
↑ або Ctrl+p	Перегортання раніше введених команд «вгору» і підстановка їх в рядок введення
↓ або Ctrl+n	Перегортання раніше введених команд «вниз» і підстановка їх в рядок введення
Del або Ctrl+d	Стирання символу праворуч від курсору
Del або Ctrl+d	Стирання символу праворуч від курсору
⇐ або Ctrl+h	Стирання символу зліва від курсору
Ctrl+k	Стирання до кінця рядка
Esc	Стирання рядка введення
Ins	Включення/виключення режиму вставки
PgUp	Перегортання сторінки сесії вгору



Комбінація клавiш	Призначення
PgDn	Перегортання сторінки сесії вниз

Зверніть особливу увагу на застосування клавiш вгору і вниз. Вони використовуються для розміщення після маркера рядка введення >> раніше введених рядків, наприклад для їх виправлення, дублювання або доповнення. При цьому вказані клавiші забезпечують перегортання раніше введених рядків зверху вниз або знизу вверх. Така можливість існує завдяки організації спеціального стека, що зберігає рядки з виконаними раніше командами.

## 2. Поняття про математичний вираз. Типи і формати даних

**Математичні вирази** будуються на основі чисел, констант, змінних, операторів, функцій і різних спецзнаків. Нижче дамо короткі пояснення суті цих понять.

Оператори MATLAB мають форму **змінна = вираз** або просто **вираз**. Вираз, як правило, формується з операторів, функцій і імен змінних. Після виконання виразу генерується матриця, яка виводиться на екран і привласнюється відповідній змінній для подальшого використання. Якщо ім'я змінної в лівій частині і знак = відсутні, автоматично генерується змінна **ans**, якій привласнюється результат обчислень.

Зазвичай, оператор завершується клавiшею **Enter**. Проте у разі необхідності оператор може бути продовжений на наступному рядку, для чого його необхідно завершити трьома або більше крапками, після яких натиснути **Enter**. З іншою сторони, в одному рядку може бути декілька операторів, розділених комою або крапкою з комою.

Якщо останнім символом в рядку є крапка з комою, то на екран не виводиться результат дії, але привласнення виконується. Це допомагає позбутись непотрібних проміжних результатів. MATLAB розрізняє великі і маленькі букви в іменах команд, функцій і змінних.

Змінна може бути видалена з робочого простору командою **clear** <ім'я\_змінної>. Команда **clear** очищає всі непостійні змінні в робочому просторі.

**Число** – найпростіший об'єкт мови MATLAB, що представляє кількісні дані. Числа можна вважати константами, імена яких співпадають з їх значеннями. Числа використовуються в загальноприйнятому уявленні про них. Вони можуть бути цілими, дробовими, з фіксованою і плаваючою крапкою. Можливе представлення чисел в добре відомому науковому форматі з вказівкою мантиси і порядку числа.

### Правила запису чисел в MATLAB:

- 1) в мантисі чисел ціла частина відділяється від дробової не комою, а крапкою, як прийнято в більшості мов програмування;
- 2) для відділення порядку числа від мантиси використовується символ **e**;
- 3) знак «плюс» у чисел не проставляється, а знак «мінус» у числа називають **унарним мінусом**;
- 4) пропуски між символами в числах не допускаються.

**Числа можуть бути комплексними:**  $z = \text{Re}(x) + \text{Im}(x) * i$ . Такі числа містять дійсну  $\text{Re}(z)$  і уявну  $\text{Im}(z)$  частини. Уявна частина має множник  $i$  або  $j$ , що дорівнює кореню квадратному з  $-1$ :

3i  
 2j  
 2+3i  
 -3.141i  
 -123.456+2.7e-3i

### **Константи і системні змінні**

Константа — це наперед визначене числове або символічне значення, представлене унікальним ім'ям. Числа (наприклад 1, -2 і 1.23) є безіменними числовими константами.

Інші види констант в MATLAB прийнято назвати **системними змінними**, оскільки, з одного боку, вони задаються системою при її завантаженні, а з іншого – можуть перевизначатися. Основні системні змінні, вживані в системі MATLAB, вказані нижче:

- 1) **i** або **j** – уявна одиниця (корінь квадратний з  $-1$ );
- 2) **pi** - число  $\pi$ – 3.1415926...;
- 3) **eps** – похибка операцій над числами з плаваючою крапкою ( $2^{-52}$ );
- 4) **realmin** – найменше число з плаваючою крапкою ( $2^{-1022}$ );
- 5) **realmax** – найбільше число з плаваючою крапкою ( $2^{1023}$ );
- 6) **inf** – значення машинної нескінченності;
- 7) **ans** – змінна, що береже результат останньої операції і звичайно що викликає його відображення на екрані дисплея;
- 8) **NaN** - вказівка на нечисловий характер даних (Not-a-Number).

### **Текстові коментарі**

Оскільки MATLAB використовується для достатньо складних обчислень, велике значення має наочність їх опису. Вона досягається, зокрема, за допомогою текстових коментарів. Текстові коментарі вводяться за допомогою символу **%**, наприклад, так:

*% Функція обчислення сили, що намагнічує*

## Змінні і привласнення їм значень

**Змінні** – це об'єкти, що мають імена, здатні берегти деякі, звичайно різні за значенням, дані. Залежно від цих даних змінні можуть бути числовими або символічними, векторними або матричними.

В системі MATLAB можна задавати змінним певні значення. Для цього використовується **операція привласнення**, що вводиться знаком рівності = .

**Типи змінних наперед не декларуються.** Вони визначаються виразом, значення якого привласнюється змінній. Так, якщо цей вираз – вектор або матриця, то змінна буде векторною або матричною.

**Ім'я змінної** (її ідентифікатор) може містити скільки завгодно символів, але запам'ятовується і ідентифікується тільки 31 початковий символ. Ім'я будь-якої змінної не повинне співпадати з іменами інших змінних, функцій і процедур системи, тобто воно повинне бути унікальним. Ім'я повинне починатися з букви, може містити букви, цифри і символ підкреслення «\_». Неприпустимо включати в імена змінних пропуски і спеціальні знаки, наприклад «+.- \*/» і т. д., оскільки в цьому випадку правильна інтерпретація виразів стає неможливою.

Багато використовувати змістовні імена для позначень змінних, наприклад `speed_1` для змінної, що позначає швидкість першого об'єкту. Змінні можуть бути звичайними і індексованими, тобто елементами векторів або матриць. Можуть використовуватися і символічні змінні, причому символічні значення ставлять в апострофи, наприклад `s='Demo'`.

## 3. Оператори і функції

**Оператор** – це спеціальне позначення для певної операції над даними – операндами.

Наприклад, найпростішими арифметичними операторами є знаки суми «+», віднімання «-», множення «\*» і ділення «/». Оператори використовуються спільно з операндами. Наприклад, у виразі «2+3» знак «+» є оператором складання, а числа 2 і 3 – операндами.

Слід зазначити, що більшість операторів відноситься до матричних операцій, що може служити причиною серйозних непорозумінь. Є ряд спеціальних операторів, наприклад, оператор «\» означає ділення справа наліво, а оператори «.\*» і «./» означають відповідно поелементне множення і поелементне ділення масивів.

Повний список операторів можна отримати, використовуючи команду «**help ops**». Нижче приведена тільки частина повного списку операторів, що містить арифметичні оператори:

У виразах використовують звичайні арифметичні операції і правила старшинства:

+ додавання

- віднімання
- \* множення
- .\* поелементне множення
- / ділення зліва направо
- \ ділення справа наліво
- ./ поелементне зліва направо
- .\ поелементне ділення справа наліво
- ^ степінь
- .^ поелементна степінь
- ' комплексно спряжене транспонування
- () визначення порядку обчислення.

**Функції** – це об'єкти, що мають унікальні імена, вони виконуючі певні перетворення своїх аргументів і при цьому повертають результати цих перетворень.

Функції в загальному випадку мають список аргументів (параметрів), взятих в круглі дужки. Якщо функція повертає декілька значень, то вона записується у вигляді  $[Y_1, Y_2, \dots] = \text{func}(X_1, X_2, \dots)$ , де  $Y_1, Y_2, \dots$  – список вихідних параметрів і  $X_1, X_2, \dots$  – список вхідних аргументів (параметрів).

Із списком елементарних функцій можна ознайомитися, виконавши команду **help elfun**, а із списком спеціальних функцій – за допомогою команди **help specfun**. Функції можуть бути вбудованими (внутрішніми) і зовнішніми, або m-функціями. Так, вбудованими є найпоширеніші елементарні функції наприклад,  $\sin(x)$  і  $\exp(y)$ , тоді як функція  $\sinh(x)$  є зовнішньою функцією. Зовнішні функції містять свої визначення в m-файлах. Вбудовані функції зберігаються в ядрі системи MATLAB, що відкомпілювалося, через що вони виконуються дуже швидко.

Імена функцій не є зарезервованими, тому їх значення можна змінювати на нові. Початкове значення можна відновити командою **clear**.

#### **Застосування оператора: (двокрапка)**

Дуже часто необхідно провести формування впорядкованих числових послідовностей. Такі послідовності потрібні для створення векторів або значень абсциси при побудові графіків. Для цього в MATLAB використовується оператор «:» (двокрапка):

*«Початкове значення» : «Крок» : «Кінцеве значення»*

Дана конструкція породжує зростаючу послідовність чисел, яка починається з початкового значення, йде із заданим кроком і завершується кінцевим значенням. Якщо Крок не заданий, то він приймає значення 1.

**4. Формування векторів і матриць. Особливості задання векторів і матриць. Дії над матрицями**

**Щоб ввести матрицю** за допомогою першого способу (повний список елементів) треба дотримуватись **наступних основних умов**:

- відділяти елементи рядка **пропусками чи комами ( чи ,)**;
- використовувати **крапку з комою (;)** для позначення кінця кожного рядка;
- заключати весь список елементів у **квадратні дужки ([])**.

Якщо треба задати **вектор з трьох елементів**, то їх значення слід перерахувати в квадратних дужках, розділяючи пропусками або комами. Так, наприклад, привласнення

```
>> V=[1 2 3]
```

```
V=
```

```
1 2 3
```

задає вектор V, має три елементи із значеннями 1, 2 і 3. Після введення вектора система виводить його на екран дисплея.

Слід відмітити, що в системі MATLAB, таким зразком вводяться вектор-рядки. Щоб задати вектор-стовпчик необхідно використовувати оператор транспонування «'», наприклад:

```
v=[1 2 3]'
```

```
v=
```

```
1
```

```
2
```

```
3
```

Задання матриці потребує вказівки декількох рядків. Для розмежування рядків застосовують **знак «;» (крапка з комою)**. Так, введення

```
>> M=[1 2 3; 4 5 6; 7 8 9];
```

задає квадратну матрицю:

```
M =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

Можливе введення елементів матриць і векторів у вигляді арифметичних виразів, що містять будь-які доступні системі функції, наприклад:

```
>> V = [2+2/(3+4) exp(5) sqrt(10)]
```

```
V =
```

```
2.2857 148.4132 3.1623
```

Для вказівки окремого елемента вектора чи матриці використовується вираз вигляду **V(n)** або **M(n,k)**. Наприклад, якщо задати

```
>> m_32=M(3,2)
```

```
m_32= 8
```

Можлива вибірка діапазону елементів матриці. Для цього використовують оператор «:». Наприклад, задавши

```
>> R=M(2:3, 1:2)
```

отримаємо

```
R= 4 5  
    7 8
```

Нижче показано, як вибирати за допомогою оператора «:» рядки, стовпчики і елементи із векторів, матриць і багатомірних масивів:

- $A(:,j)$  – це  $j$ -й стовпчик з  $A$ ;
- $A(i,:)$  – це  $i$ -й рядок з  $A$ ;
- $A(:,:)$  – еквівалент двомірного масиву. Для матриць це аналогічно  $A$ ;
- $A(j:k)$  – це  $A(j), A(j+1), \dots, A(k)$ . Для матриць – вектор-рядок з елементів з  $j$  по  $k$  при послідовній нумерації елементів матриці  $A$  зверху-вниз-зліва-направо;
- $A(:,j:k)$  – це матриця з стовпчиків  $A(:,j), A(:,j+1), \dots, A(:,k)$ ;
- $A(k, :)$  – це матриця – «зріз» трьохмірного масиву  $A$  по  $k$ -ому рядку;
- $A(i, j, :)$  – вектор, виділений з трьохмірного масиву  $A$ . Вектор включає елементи  $A(i, j, 1), A(i, j, 2), A(i, j, 3)$  і т. д.;
- $A(:)$  – записує всі елементи масиву  $A$  у вигляді стовпчика.

### Видалення стовпчиків і рядків матриць

Для формування матриць і виконання ряду матричних операцій виникає необхідність видалення окремих стовпців і рядків матриці. Для цього використовуються порожні квадратні дужки «[ ]» і оператор двокрапка. Виконаємо це з матрицею  $M$ :

```
>> M=[1 2 3; 4 5 6; 7 8 9];
```

Видалимо другий стовпець використовуючи оператор : (двокрапка):

```
>> M(:,2)=[ ]
```

$M =$

```
1 3  
4 6  
7 9
```

А зараз, використовуючи оператор : (двокрапка), видалимо другий рядок:

```
>> M(2,:)=[ ]
```

$M =$

```
1 3  
7 9
```

### Об'єднання малих матриць у велику

Описаний спосіб задання матриць дозволяє виконати операцію конкатенації – об'єднання малих матриць у велику.

Так, якщо є  $N$  матриць  $A_1, A_2, \dots, A_N$  з однаковою кількістю рядків, то об'єднати їх в одну матрицю, розміщуючи блоки горизонтально можна командою

$$A=[A_1, A_2, \dots, A_N].$$

Ця операція називається горизонтальною конкатенацією.

Вертикальна конкатенація (при умові, що складові блоки-матриці мають однакову кількість стовпчиків) реалізується аналогічно, шляхом відокремлення блоків оператором « ; » замість « , »:

$$A=[A_1; A_2; \dots; A_N]$$

Наприклад

$$A_1 = [1 \ 2 \ 3; 7 \ 8 \ 9]$$

$$A_2 = [4 \ 5; 6 \ 7]$$

$$A_3 = [-5 \ -1 \ -3]$$

$$A=[A_1, A_2]$$

$$A =$$

$$1 \ 2 \ 3 \ 4 \ 5$$

$$7 \ 8 \ 9 \ 6 \ 7$$

$$A=[A_1; A_3]$$

$$A =$$

$$1 \ 2 \ 3$$

$$7 \ 8 \ 9$$

$$-5 \ -1 \ -3$$

### Операції над матрицями

У мові MATLAB є дві функції ділення матриць, які не визначені в математиці. При цьому вводяться поняття ділення матриць зліва направо «/» і ділення матриць справа наліво «\». Операція  $\mathbf{B}/\mathbf{A}$  еквівалентна операції  $\mathbf{B} \cdot \text{inv}(\mathbf{A})$ . Цю операцію зручно використовувати для розв'язування матричного рівняння  $\mathbf{X} \cdot \mathbf{A} = \mathbf{B}$ . Аналогічно операція  $\mathbf{A} \backslash \mathbf{B}$  рівносильна операції  $\text{inv}(\mathbf{A}) \cdot \mathbf{B}$ , яка являє собою розв'язок матричного рівняння  $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$ . Однак для розв'язування цього матричного рівняння кращим з точки зору мінімізації часу розрахунку і підвищення точності обчислень є використання оператора матричного ділення  $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ . Ця операція використовує метод виключення Гаусса без явного формування оберненої матриці.

**diag(A)** – повертає вектор з елементами головної діагоналі матриці  $\mathbf{A}$ .

**trace(A)** – повертає слід (суму елементів головної діагоналі) матриці  $\mathbf{A}$ .

**rot90(A)** – здійснює поворот матриці  $\mathbf{A}$  на  $90^\circ$  проти годинникової стрілки.

**rot90(A, k)** – здійснює поворот матриці  $\mathbf{A}$  на величину  $90 \cdot k$  градусів, де  $k$  – ціле число.

Створити квадратну одиничну діагональну матрицю розміром  $n \times n$  можна функцією **eye(n)**.

Функція **squeeze(A)** повертає масив, у якому видалені всі одиничні розмірності. Для визначення розмірів матриць і довжин векторів застосовуються 2 команди **length** і **size**:

**size(A)** – розмір матриці. Для  $m \times n$  матриці **A** повертає двохелементний вектор-рядок, у якому перша складова – число рядків  $m$ , а друга складова – число стовпчиків  $n$ .

**length(A)** – повертає максимальну розмірність матриці **A**.

Роботу цих функцій можна зрозуміти з наступного прикладу

Команда	Результат	Команда	Результат
<code>a=[1 2 3;7 8 9]</code>	<pre>a =      1     2     3      7     8     9</pre>	<code>b=[5 6 4 7]</code>	<pre>b =      5     6     4     7</pre>
<code>size(a)</code>	<pre>ans =      2     3</pre>	<code>size(b)</code>	<pre>ans =      1     4</pre>
<code>length(a)</code>	<pre>ans =      3</pre>	<code>length(b)</code>	<pre>ans =      4</pre>

## 5. Збереження робочої області сесії

Змінні і визначення нових функцій в системі MATLAB зберігаються в особливій області пам'яті, іменованою робочою областю. MATLAB дозволяє зберігати значення змінних у вигляді бінарних файлів з розширенням **.mat**.

Для цього служить команда **save**, яка може використовуватися у ряді форм:

**save fname** — записується робоча область всіх змінних у файлі бінарного формату з ім'ям `fname.mat`;

**save fname X** — записує тільки значення змінної **X**;

**save fname X Y Z** — записує значення змінних **X**, **Y** і **Z**.

Слід зазначити, що можливості збереження всього тексту сесії, сформованої в командному режимі, команда `save` не дає. І не випадково! Річ у тому, що сесія є результатом проб і помилок, і її текст разом з правильними визначеннями містить повідомлення про помилки, перевизначення функцій і змінних і багато іншого «лушпиння». Необхідності зберігати таку «творчість» звичайно немає. А якщо є – для цього служить команда **diary**, описана трохи нижче.

## 6. Ведення щоденника

Ми відзначали, що сесії не записуються на диск стандартною командою **save**. Проте якщо така необхідність є, можна скористатися спеціальною командою для ведення так званого щоденника сесії:

**diary file name** — веде запис на диск всіх команд в рядках введення і отриманих результатів у вигляді текстового файлу з вказаним ім'ям;

**diary off** — припиняє запис у файл;

**diary on** — знов починає запис у файл.



Таким чином, чергуючи команди **diary off** і **diary on**, можна зберігати потрібні фрагменти сесії в їх формальному вигляді. Команду **diary** можна задати і у вигляді функції **diary('file')**, де рядок 'file' задає ім'я файлу.

### 7. Завантаження робочої області сесії

Для завантаження робочої області раніше проведеної сесії (якщо вона була збережена) можна використовувати команду **load**:

**load fname ...** — завантаження раніше збережених у файлі **fname.mat** визначень із специфікаціями на місці багатокрапки, подібними описаним для команди **save**;

**load('fname'....)** — завантаження **файлу fname.mat** у формі функції.

Якщо команда (або функція) **load** використовується в ході проведення сесії, то відбудеться заміна поточних значень змінних тими значеннями, які були збережені в прочитуваному Мат-файлі.

### 8. Завершення роботи з системою

Для завершення роботи з системою можна використовувати команди **exit**, **quit**. Якщо необхідно зберегти значення всіх змінних (векторів, матриць) системи, то перед введенням команди **exit** слід дати команду **save** потрібної форми. Команда **load** після завантаження системи прочитує значення цих змінних і дозволяє почати роботу з системою з того моменту, коли вона була перервана.

#### *Порядок виконання роботи:*

1.1 Відкрийте файл для ведення **щоденника сесії** (для даної лабораторної роботи - основний документ звітності про її виконання).

Для цього виконайте наступні дії:

1) створіть директорію **D:\gr\_№групи\**, а в ній файл **Прізвище.txt** (це необхідно зробити поза програмою *MatLab* – в будь-якому файловому менеджері);

2) запустіть пакет прикладних програм *MatLab*;

3) після появи у вікні *Command* символа «>>» наберіть **diary D:\gr\_№групи\Прізвище.txt** (тим самим, ви відкриєте **щоденник сесії** – «запис всіх дій під час вашої роботи»);

1.2. Ознайомтесь з константами *MatLab* (системні змінні), для чого введіть по чергово наступні символи (ввід кожного символу завершується натисканням клавіші <Enter>) **i, j, pi, inf**, і т.і. (див. вище).

1.3. Розрахуйте значення функції  $y(x)$  при значеннях  $x$  і  $\omega$  вказаних в таблиці 10.1.

1.4. Створіть вектор  $x$ , що містить послідовність чисел від -1 до 1 с кроком 0.1. Розрахуйте значення функції  $y(x)$  з таблиці 10.2, для всіх значень даного вектора.

1.5. Освойте роботу з матрицями, виконавши завдання наведені в таблиці 10.3.

Таблиця 10.2

**Функції до завдання з пункту 1.3**

№ КОМП.	Функція
1	$y(x) = \frac{5}{\pi} x^2 \cdot \sin \frac{\omega}{x+2} + \frac{5}{7\pi} \cos(3\omega \cdot x), \quad \omega = 2, \quad x = 3$
2	$y(x) = x \cdot \cos \frac{4\omega}{\pi} + \frac{4}{9\pi} \cos(0.5 + \omega \cdot x^3) \quad \omega = 0.5, \quad x = 1/3$
3	$y(x) = \frac{4}{\pi} \sin^2(\omega \cdot x) - (x + \frac{7}{12\pi}) \cdot \cos 7\omega \quad \omega = 1/2, \quad x = 0.7$
4	$y(x) = \frac{4}{3\pi} \sin(\omega \cdot x^3) - \frac{5x}{\pi} \cos^2 3\omega \quad \omega = 2/7, \quad x = 3$
5	$y(x) = \frac{3}{\pi} \cdot x^2 \cdot \cos 5\omega + \frac{4}{7} \cos(2\omega + \frac{x}{\pi}) \quad \omega = 7/27, \quad x = 4$
6	$y(x) = \pi \cos^3(\omega \cdot x) + \frac{4}{\pi} (x^2 - 1) \cdot \cos 7\omega \quad \omega = 1/5, \quad x = 33$
7	$y(x) = \pi x \cdot \sin \left( \frac{\omega - x}{x^2 + 2} \right) + \frac{4}{\pi} \cos^2(7\omega \cdot x) \quad \omega = 0.8, \quad x = 3.65$
8	$y(x) = \pi \sin(\omega^2 - x^2) - \frac{3x^3}{2\pi} \cdot \cos(3\omega \cdot x) \quad \omega = 2, \quad x = 3/45$
9	$y(x) = 2\pi \sin^3(3\omega \cdot x) + \left( \frac{\omega + x}{\omega^2 + 1} \right) \cos x \quad \omega = 0.23, \quad x = 3/65$
10	$y(x) = \frac{x}{x^2 + 1} \cdot \cos(2\omega - 1) + \frac{\pi}{3} \sin^2(3\omega \cdot x) \quad \omega = 2.2, \quad x = 0.3$
11	$y(x) = \frac{x + \omega}{\pi} \cdot \sin^2 \omega + \frac{5}{3\pi} \cos(3\omega + x), \quad \omega = 2/15, \quad x = 4.2$
12	$y(x) = (\omega - 2) \cdot \cos \frac{x}{\pi} + \frac{4}{9\pi} \sin^2(\omega \cdot x) \quad \omega = 0.54, \quad x = 1/4$
13	$y(x) = 4\pi x \cos \frac{\omega^2}{\pi} - \frac{7}{12\pi} \cos(7\omega \cdot x + 2) \quad \omega = 1/6, \quad x = 2.7$
14	$y(x) = x \cdot \sin(\omega + \frac{4\pi}{3}) - \frac{5}{\pi} \cos^2(\omega \cdot x^3) \quad \omega = 1/7, \quad x = 4.25$

## Варіанти завдань до пункту 1.5

№ комп.	1	2	3	4	5
Скаляр	$x=33+44$ $z=0.3\pi$	$c=35-77$ $d=0.4\pi$	$d=2.2/55$ $k=j2$	$y=77/4.4$ $d=2.2/\pi$	$z=4/3+3$ $s=i5$
Вектор	$C=2/x, 4,$ $7$	$X=4, 6\cdot c,$ $7$	$Y=7, 1,$ $7-d$	$A=5, 6,$ $y\cdot 8.8$	$Q=4.4, 5,$ $z/44$
Матриця $B=$	$1, 2, 3$ $z\cdot C$ $4, 5, x-5$	$5, 8+d,$ $9$ $\pi\cdot X$ $5+c, 7,$ $9\cdot d$	$d, 5,$ $77+k$ $88, 6-k, 9$ $Y\cdot d$	$y\cdot 2, 55,$ $7/8$ $4, y+5,$ $6-d$ $A\cdot y$	$z, 55,$ $88+s$ $z-1, 66,$ $88\cdot s$ $Q\cdot z$
Виділіть з отриманої матриці B підматрицю G розмірністю 2x2, в якій вирізані:					
-<>-	останній рядок і стовпчик	перші рядок і стовпчик	перший рядок і другий стовпчик	третій рядок і другий стовпчик	перший рядок і третій стовпчик
Збережіть набрані в ході сесії змінні (див. вище “збереження робочої області”), у файл <b>D:\gr_№групи\Прізвище</b>					
№ комп.	6	7	8	9	10
Скаляр	$n=2-46$ $z=2.3\pi$	$c=5+45$ $d=1.4i$	$d=2.2/65$ $k=2\pi$	$y=73/4.4$ $d=5.2/\pi$	$z=7/3+8$ $s=j5$
Вектор	$C=3/n, 5,$ $\pi$	$X=2, 3\cdot c,$ $1$	$Y=2, 3,$ $4-d$	$A=3, 3,$ $y\cdot 5.1$	$Q=4.5, 6,$ $z/7$
Матриця $B=$	$4, 5, 3$ $z\cdot C$ $1, 2, x-3$	$5, 9+d,$ $3$ $\pi\cdot X$ $4+c, 6,$ $8\cdot d$	$d, 5,$ $7+2k$ $88, 6-k, 9$ $Y\cdot k$	$y\cdot 2, 35,$ $7/6$ $4, y+5,$ $6-d$ $A\cdot d$	$z, 54,$ $38+s$ $z-1, 14,$ $32\cdot s$ $Q\cdot z$
Виділіть з отриманої матриці B підматрицю G розмірністю 2x2, в якій вирізані:					
-<>-	перші рядок і стовпчик	останній рядок і стовпчик	перший рядок і третій стовпчик	третій рядок і перший стовпчик	перший рядок і другий стовпчик
Збережіть набрані в ході сесії змінні (див. вище “збереження робочої області”)					
Визначте різницю між операторами «.*» і «*», використовуючи введені змінні. Для чого виконайте операції множення між двома скалярами, скаляром і вектором, двома векторами, вектором і матрицею, двома матрицями, використовуючи обидва оператори по чергово.					



## Лабораторна робота №12

### Знайомство з графічними можливостями паketу прикладних програм MATLAB

#### *Теоретичні відомості*

##### **1. Основні відомості**

Графіка є могутнім інтерактивним засобом аналізу і синтезу систем управління. Повне рішення задач аналізу і синтезу часто вимагає детального розгляду безлічі даних в найрізноманітніших форматах і уявленнях. MATLAB надає користувачу широкі можливості для графічної обробки аналізованих даних.

На відміну від текстової інформації, системою MATLAB, що виводиться, у відповідь на запити користувача безпосередньо в командне вікно сесії, для представлення графічних зображень MATLAB використовує спеціальне **графічне вікно (FIGURE)**. В одному сеансі роботи з MATLAB може бути створено декілька графічних вікон (figure 1, figure 2, figure n) і, крім того, кожне з цих вікон може бути розбите на кілька графічних областей з своєю окремою системою координат. Користувачу надаються можливості виведення графіків в будь-яке з відкритих графічних вікон або в будь-яку підобласть вибраного вікна, з очищенням попередньої графічної інформації або накладенням декількох графіків один на інший. При виведенні графіків користувач може задавати колір лінії, тип лінії і маркерних точок, створювати підписи координатних осей, заголовки графіків, розміщувати пояснюючі написи на поточному графіку, виводити або прибирати сітку і т.п.

Найпростішою формою представлення інформації в графічному вигляді є побудова двовимірного графіка. Існують три основні групи графічних функцій, що надаються системою MATLAB для цієї мети. Перша група призначена для активації і управління графічними вікнами, друга визначає формат графіка і викликає процедуру його побудови і третя – оформлення графічної інформації.

##### **2. Функції активації і управління графічними вікнами**

Для створення графічного вікна використовується функція **figure**. Якщо одне або декілька графічних вікон вже створено, то команда **figure** створить нове вікно з наступним по порядку зростання номером.

Враховуючи, що система MATLAB дозволяє працювати з декількома графічними вікнами, важливим є поняття **активного** або **поточного графічного вікна**, тобто вікна готового у даний момент до виведення графічної інформації. Для задання поточного вікна виведення серед відкритих графічних вікон слід виконати команду **figure (n)**, де n – номер вікна, що активується. У випадку якщо

серед відкритих вікон немає графічного вікна з даним номером, то воно буде створено, а потім активовано.

Слід помітити, що якщо немає відкритих графічних вікон, то перше графічне вікно автоматично буде створено і активізовано при виконанні будь-якої функції, результатом якої повинен бути деякий графік (наприклад, функції **plot**, детально описаної далі).

В початковому стані при виведенні нового графіка в активне вікно MATLAB очищає раніше виведену в це вікно графічну інформацію. Для побудови нової залежності без знищення попередніх необхідно виконати команду **hold on**, а потім вже задати команду побудови графіка. При цьому, якщо необхідно, MATLAB автоматично підстроїть масштаби по осях координат. Повернутися до первинного порядку побудови можна командою **hold off**. Не виходячи з режиму **hold on** графічне зображення можна видалити з вікна за допомогою функції **cla**.

Якщо поточне графічне вікно знаходиться в згорнутому стані, то побудова деякої залежності відбувається без його розвертання. Виконання функції **shg** в командному рядку дозволяє розвернути і вивести активне графічне вікно на перший план.

Графічне вікно можна розділити на декілька підобластей меншого розміру. Функція **subplot (m, n, p)** ділить поточне вікно на **m x n** підобластей (**m** — число «рядків», **n** — число «стовпців»), а цілий параметр **p** в цій функції визначає номер активної підобласті. Нумерація йде зліва направо і зверху вниз.

### Приклад 1

```
x = -1:1:1;  
y1 = sin(x);  
subplot(2, 1, 1), plot(x, y1);  
y2 = log(abs(y1));  
subplot(2, 1, 2), plot(x, y2)
```

### 3. Інтерфейс графічних вікон

Починаючи з версії MATLAB 5.3.1 в графічних вікнах появилось нове меню Tools (Інструменти). Це меню містить ряд команд:

- Show Toolbar – показати чи сховати панель інструментів;
- Enable Plot Editing – включення чи виключення редагування графіку;
- Axis Properties – вивести вікно властивостей осей графіка;
- Line Properties – вивести вікно властивостей ліній графіка;
- Text Properties – задати виведення текстових коментарів;
- Lock/Unlock Axes Position – заблокувати або розблокувати позицію осей;
- Show Legend – показати легенду;
- Add – додати в графік новий об'єкт з підменю;

- Zoom In – збільшити розміри графіка;
- Zoom Out – зменшити розміри графіка;
- Rotate 3D – почати повертання графіка мишею.

Можливість повертання графіка мишею – спосіб, введений в цілий ряд систем комп’ютерної математики (MathCad, Maple V). При введенні цієї команди в дію навколо фігури появляється окреслюючий її паралелепіпед, який можна повертати мишею в будь-якому напрямі, що дає можливість спостерігати фігуру під різними кутами. Відпустивши мишу, можна спостерігати графік в просторі. Цікаво, що ця можливість діє і по відношенню до двовимірних графіків. При цьому повертається площина, в якій розміщено графік.

#### 4. Функції побудови двовимірних графіків

Для побудови двовимірних графіків в середовищі MATLAB служать функції, які приведені в таблиці 14.1. Синтаксис, приведених в таблиці 11.1 функцій, подібний синтаксису функції **plot**, яка і буде розглянута більш детально.

Таблиця 11.1

**Функції побудови двовимірних графіків**

Функція	Призначення
plot	побудова графіка в декартовій системі координат
loglog	побудова графіка з логарифмічними осями координат
semilogx	побудова графіка з логарифмічною віссю абсцис
semilogy	побудова графіка з логарифмічною віссю ординат
polar	побудова графіка в полярній системі координат
plotyy	побудова графіка в декартовій системі координат з висновком осей справа і зліва від вікна графіка

Функція **plot** має різні форми, пов’язані з вхідними параметрами, наприклад **plot(y)** створить лінійний графік залежності елементів  $y$  від їх індексів. Якщо задати два вектори в якості аргументів, **plot(x,y)** створить графік залежності  $y$  від  $x$ .

#### Приклад 2

Побудувати графік значень функції **sin** від 0 до  $2\pi$ .

```
clear
t = 0:pi/100:2*pi;
y = sin(t);
plot(t,y)
```

Виклик функції **plot** з багатьма парами  $x$ - $y$  створить багато графіків. MATLAB автоматично присвоює кожному графіку свій колір (у випадку, якщо користувач сам цього не зробив).

**Приклад 3** (наступні три рядка відображають графік близьких функцій):

```
y2 = sin(t-.25);
```

```
y3 = sin(t-.5);
```

```
plot(t, y, t, y2, t, y3)
```

Змінити колір, стиль ліній і маркерів можна таким чином:

```
plot(x, y, 'колір_стиль_маркер'),
```

де 'колір\_стиль\_маркер' – символний рядок взятий в одинарні лапки, який складається з типу кольору, стилю ліній і маркерів

Символи, що відносяться до кольору: 'c', 'm', 'y', 'r', 'g', 'b', 'w', 'k' позначають, відповідно, голубий, малиновий, жовтий, червоний, зелений, синій, білий і чорний кольори.

Символи, що відносяться до типу ліній: '\_', '--', ':', '-.' 'none' позначають, відповідно, неперервну, розривну, пунктирну, штрих-пунктирну, відсутню лінію.

Маркери, що найбільше зустрічаються: '+', 'o', '\*', 'x'.

#### Приклад 4

```
clear
```

```
x=0:.01:2*pi;
```

```
y1=sin(x);
```

```
y2=sin(2*x);
```

```
y3=sin(4*x);
```

```
plot(x,y1,'--',x,y2,':',x,y3,'+')
```

### 5. Функції керування оформленням графіків

Призначення функцій, що відповідають за оформлення графіків і графічних вікон, розшифровується в таблиці 11.2.

Таблиця 11.2

#### Функції управління оформленням графіків

Функція	Призначення
title ('text')	поміщає над графіком заголовок 'text'
legend (string1, string2...)	поміщає на поточному графіку пояснення у вигляді заданих текстових рядків
xlabel ('text')	позначає вісь $x$ написом 'text'
ylabel ('text')	позначає вісь $y$ написом 'text'
text (p1, p2, 'text')	додає до поточного графіка напис 'text', починаючи з позицією (p1, p2), де (p1, p2) - координати крапки по осях $x$ и $y$
grid on	наносить лінії сітки на поточний графік
grid off	видаляє лінії сітки з поточного графіка
axis	управління виведенням координатних осей



Функція **axis** має декілька можливостей для настройки масштабу, орієнтації і коефіцієнта стиснення координатних осей.

Звичайно MATLAB автоматично знаходить максимальне і мінімальне значення змінних і вибирає відповідний масштаб і маркіровку осей. Функція **axis([xmin xmax ymin ymax])** замінює значення за замовчуванням граничними значеннями, що вводяться користувачем.

У функції **axis** можна також використовувати ключові слова для керування зовнішнім виглядом осей. Наприклад, **axis square** створює осі абсцис і ординат рівної довжини, а **axis equal** створює для цих осей відмітки приростів однакової довжини. Команда **axis auto** повертає значення за замовчуванням і переводить побудову осей в автоматичний режим. Функція **axis on** включає позначення осей і мітки проміжних розподілів, а **axis off**, відповідно, вимикає позначення осей і мітки проміжних розподілів.

### Приклад 5

```
Побудувати графік функції  $y=\sin(\alpha*t)$   
alpha=1.8  
t=[0:0.01:1];  
y=sin(alpha*t);  
plot(t,y)  
xlabel('t (c)')  
ylabel('y(t)=sin(\alphan)')  
grid on
```

### Порядок виконання роботи

1. Відкрийте файл для ведення щоденника сесії **D:\GR\_№ групи\Прізвище.txt**. Виконайте 5 прикладів.
2. Створіть нове графічне вікно і розбийте його на області у відповідності з завданням (рис. 11.1). Пропорції розбивки вказані вздовж сторін областей графічного вікна, умовна нумерація вказана в центрі даних областей.
3. В областях виконайте побудову у відповідності з таблицею 11.3.
4. При необхідності виконайте масштабування графіків за допомогою команди **axis**. Підпишіть отримані графіки за допомогою функцій **title**, **xlabel**, **ylabel**
5. Перенесіть отримані фігури за допомогою команди з меню графічного вікна **Edit**→**Copy Figure** в буфер обміну, а потім вставте в якості рисунка в документ Word.

2	3
3 №1	№3 2
	№4 2
3 №2	№5 2

Варіанти 1, 7

3	2
3 №1	№3 2
	№4 2
3 №2	№5 2

Варіанти 2, 8

3	4
3 №1	№3 2
	№4 2
3 №2	№5 2

Варіанти 3, 9

2	3
4 №1	№3 3
	№4 2
4 №2	№5 2

Варіанти 4, 10

4	3
4 №1	№3 2
	№4 3
3 №2	№5 3

Варіанти 5, 11

3	5
5 №1	№3 2
	№4 3
3 №2	№5 3

Варіанти 6, 12

Рис. 11.1. Варіанти розбивки вікна фігури на області

## Варіанти завдань

№ варіанту	област	Функція
1, 6	1	$y_1(t) = 5 \cdot e^{a \cdot t} \cdot \sin(0.1t)$ , $a = -10$ , $0 \leq t \leq 1$ , $\Delta t = 0.01$ неперервна синя
	2	$y_2(t) = 2 \cdot e^{a \cdot t} \cdot \sin(0.3t)$ , $a = 10$ , $0 \leq t \leq 1$ , $\Delta t = 0.01$ неперервна червона, маркери типу "+"
	3	$y_3(x) = 2 \sin(x) \cdot \sin(2x)$ , $y_4(x) = 2 \sin(x) \cdot \sin(2x + \pi/6)$ , $-2\pi \leq x \leq 2\pi$ , $\Delta x = 1^\circ$ неперервна чорна, пунктирна чорна; легенда
	4	$y_5(x) = 2 \sin(x) + 0.3 \sin(5x + \pi/3)$ , $-2\pi \leq x \leq 2\pi$ , $\Delta x = 1^\circ$ штрих-пунктирна синя
	5	$y_6(x) = 2 \sin(x) / \cos(2x + \pi/6)$ , $-\pi \leq x \leq \pi$ , $\Delta x = 1^\circ$ розривна жовта
2, 7	1	$y_1(t) = 50 \cdot e^{a \cdot t} \cdot \sin(0.2t)$ , $a = -1$ , $0 \leq t \leq 2$ , $\Delta t = 0.01$ неперервна чорна
	2	$y_2(t) = 50 \cdot e^{a \cdot t} \cdot \sin(0.2t)$ , $a = 1$ , $0 \leq t \leq 2$ , $\Delta t = 0.01$ неперервна червона
	3	$y_3(x) = \sin(x) \cdot \cos(5x)$ , $-2\pi \leq x \leq 2\pi$ , $\Delta x = 1^\circ$ неперервна зелена маркери типу 'o'
	4	$y_{41}(x) = 2 \sin(x) \cdot \cos(2x + \pi/3)$ , $y_{42}(x) = \sin(x) \cdot \cos(2x + \pi/3)$ , $-2\pi \leq x \leq 2\pi$ , $\Delta x = 1^\circ$ синя неперервна, штрих-пунктирна; легенда
	5	$y_5(x) = 2 \sin(x) + 2 / \cos(3x + \pi/4)$ , $-\pi \leq x \leq \pi$ , $\Delta x = 1^\circ$ розривна синя
3, 8	1	$y_1(t) = 2 \cdot e^{a \cdot t} \cdot \sin(0.03t)$ , $a = -1$ , $0 \leq t \leq 2$ , $\Delta t = 0.01$ неперервна синя
	2	$y_2(t) = 2 \cdot e^{a \cdot t} \cdot \sin(0.03t)$ , $a = -3$ , $0 \leq t \leq 2$ , $\Delta t = 0.01$ неперервна червона

	3	$y_3(x) = \sin(x) \cdot \cos(5x), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ неперервна жовта, маркери типу '*'
	4	$y_4(x) = \sin(x),$ $y_5(x) = \sin(x) + 0.4 \cos(5x + \pi/7), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ синя неперервна, штрих-пунктирна; легенда
	5	$y_6(x) = 2 \sin(x) - 10 / \cos(5x + \pi/4), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ пунктирна червона
4, 9	1	$z_1(t) = 5 \cdot e^{at} \cdot \sin(0.05t), \quad a = -5, \quad 0 \leq t \leq 5, \quad \Delta t = 0.01$ неперервна зелена
	2	$z_2(t) = 5 \cdot e^{at} \cdot \sin(0.05t), \quad a = -3, \quad 0 \leq t \leq 1, \quad \Delta t = 0.01$ пунктирна синя
	3	$z_3(x) = 10 \cos(x) \cdot \sin(2x), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ неперервна чорна, маркери типу 'x'
	4	$v_4(x) = 4 \sin(x),$ $v_5(x) = 4 \sin(x) + \sin(3x + \pi/3), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ штрих-пунктирна червона; легенда
	5	$z_6(x) = 2 \cos(x) / \sin(4x + \pi/10), \quad -4\pi \leq x \leq 4\pi, \quad \Delta x = 1^\circ$ пунктирна жовта
5, 10	1	$u_1(t) = 3 \cdot e^{at} \cdot \sin(0.02t), \quad a = -1, \quad 0 \leq t \leq 4, \quad \Delta t = 0.02$ неперервна синя
	2	$u_2(t) = 3 \cdot e^{at} \cdot \sin(0.02t), \quad a = 1, \quad 0 \leq t \leq 4, \quad \Delta t = 0.02$ неперервна червона
	3	$u_3(x) = 4 \sin(x),$ $u_4(x) = 4 \sin(x) \cdot \cos^2(2x), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ неперервна зелена; легенда
	4	$u_5(x) = \sin^2(x) + \cos(2x + \pi/3), \quad -2\pi \leq x \leq 2\pi, \quad \Delta x = 1^\circ$ штрих-пунктирна жовта, маркери типу 's'
	5	$u_6(x) = 2 \sin(x) / \cos(x + \pi/4), \quad -\pi \leq x \leq \pi, \quad \Delta x = 1^\circ$ пунктирна синя

# Лабораторна робота №13

## Засоби програмування системи MATLAB

### *Теоретичні відомості*

#### **1. Основні відомості**

MATLAB – це потужна мова програмування. Файли, які містять код на мові MATLAB, називаються m-файлами. Файл з виразами зберігають під іменем, що закінчується на **.m**.

Для створення m-файлів може використовуватися як вбудований редактор, так і будь-який текстовий редактор, підтримуючий формат ASCII. Підготовлений і записаний на диск m-файл стає частиною системи, і його можна викликати як з командного рядка, так і з іншого m-файла.

Є два типи m-файлів: файли-сценарії і файли-функції. Важливо, що в процесі свого створення вони проходять синтаксичний контроль за допомогою вбудованого в систему MATLAB редактора/відладчика m-файлів.

#### **2. Структура і властивості файлів сценаріїв**

Файл-сценарій, іменованій також Script-файлом, є просто записом серії команд без вхідних і вихідних параметрів. Він має наступну структуру:

%Основний коментар

%Додатковий коментар

Тіло файлу з будь-якими виразами.

**Важливі наступні властивості файлів-сценаріїв:**

- вони не мають вхідних і вихідних аргументів;
- працюють з даними з робочої області;
- в процесі виконання не компілюються;
- є фіксованою у вигляді файлу послідовністю операцій, повністю аналогічною тій, що використовується в сесії.
- -основним коментарем є перший рядок текстових коментарів;
- додатковим коментарем є подальші рядки.

Основний коментар виводиться при виконанні команд `lookfor` і `help ім'я_каталога`. Повний коментар виводиться при виконанні команди `help ім'я_файла`.

Розглянемо наступний файл-сценарій `plotdata.m` (пояснення дано в коментарях самого файлу):

% Побудова графіка функції  $y=\sin(\alpha*t)$ .

% Значення  $\alpha$  повинно бути задане в робочій області до виклику

% скрипта на виконання

`t=[0:0.01:1];`

```

y=sin(alpha*t);
plot(t,y)
xlabel('t (c)')
ylabel('y(t)=sin(\alphat)')
grid on

```

### 3. Керуючі структури

Поруч із програмами з *лінійною структурою*, інструкції яких виконуються точно по порядку, існує багато програм, структура яких *нелінійна*. При цьому гілки програм можуть виконуватися в залежності від умов, інколи з кінцевим числом повторів – циклів, інколи у вигляді циклів, що закінчуються виконанням заданої умови. Для створення таких програм необхідні спеціальні керуючі структури. Вони є в будь-якій мові програмування, в тому числі і в MATLAB.

#### Умовний оператор **if**

Оператор **if** обчислює логічний вираз і виконує групу операторів, якщо вираз істинний. Необов'язкові ключові слова **elseif** і **else** служать для виконання альтернативних груп операторів. Ключове слово **end**, яке узгоджується з **if**, завершує останню групу операторів. Таким чином, всі групи операторів містяться між чотирма ключовими словами, без використання яких-небудь дужок.

Варіанти використання умовного оператора **if** у загальному вигляді описуються однією з наступних структур:

<pre> if Умова     Інструкція end </pre>	<pre> if Умова     Інструкція_1 else     Інструкція_2 end </pre>	<pre> if Умова_1     Інструкція_1 elseif Умова_2     Інструкція_2 else     Інструкція_3 end </pre>
--	--	--

Робота цих конструкцій очевидна – при виконанні умови виконуються відповідні інструкції.

Умова записується у вигляді:

Вираз\_1 Оператор\_відношення Вираз\_2,

причому в якості Операторів\_відношення в MATLAB використовуються наступні **оператори**:

<	менше ніж
>	більш ніж
<=	менше або рівно
>=	більше або рівно
==	рівно

~= не рівно

Відзначимо, що знак = використовується в операторах присвоєння, тоді як знак == використовується в операторах відношення. Оператори відношення (або, іншими словами, логічні змінні, які вони створюють) можуть об'єднуватися за допомогою наступних логічних операторів:

&	І
	АБО
~	НЕ

Коли ці оператори застосовуються до скалярів, то результатом є також скаляр **1** або **0** залежно від того, чи є результат **істиною** або **брехнею**.

### Цикли типу **for...end**

**Цикл for** повторює групу операторів фіксоване число раз. Ключове слово **end** закінчує цикл. Синтаксис конструкцій такого циклу має наступний вигляд:

```
for var=Вираз Інструкція1; ... Інструкція 2, end
або
for var=Вираз
    Інструкції
end
```

Вираз найчастіше записується у вигляді s:d:e, де s – початкове значення змінної циклу var, d – приріст цієї змінної і e – кінцеве значення керуючої змінної, при досягненні якого цикл завершується.

Наступні приклади пояснюють застосування циклу для отримання квадратів значень змінної циклу:

```
» for i=1:5 i^2, end;
» for X=0:.25:1 X^2, end
```

Оператор **continue** передає керування в наступну ітерацію циклу, пропускаючи оператори, які записані за ним, причому у вкладеному циклі він передає керування на наступну ітерацію основного циклу. Оператор **break** може використовуватись для дострокового переривання виконання циклу. Як тільки він зустрічається в програмі, цикл переривається.

Можливі вкладені цикли, наприклад:

```
for i=1:3
    for j=1:3
        A(i,j)=i+j;
    end
end
```

Слід відмітити, що робота з матрицями з допомогою вкладених циклів займає більше часу. Крім того, МАТАВ уже містить практично всі функції

обробки даних в матрицях. Однак, застосування циклів нерідко є більш наглядним і зрозумілим.

### Цикли типу **while...end**

У загальному вигляді цикл **while** записується у вигляді

**while** <умова>

<оператори>

**end**

<Оператори> повторюватимуться до тих пір, поки <умова> залишатиметься істинною. Наприклад, для заданого числа **a** наведена далі послідовність операторів обчислить і виведе на дисплей найменше негативне число **n**, таке що  $2n < a$ :

```
n = 0;
```

```
while 2*n < a
```

```
n = n + 1;
```

```
end
```

### 4. Структура М-файла-функції

М-файл-функція є типовим об'єктом мови програмування системи MATLAB. Одночасно він є повноцінним модулем з погляду структурного програмування, оскільки містить вхідні і вихідні параметри і використовує апарат локальних змінних. Структура такого модуля з одним вихідним параметром виглядає таким чином:

```
function var=f_name(Список_параметрів)
```

```
%Основний коментар
```

```
%Додатковий коментар
```

```
Тіло файлу з будь-якими виразами
```

```
var=вираз
```

#### **М-файл-функція має наступні властивості:**

- він починається з оголошення **function**, після якого указується ім'я змінної **var** — вихідного параметра, ім'я самої функції і список її вхідних параметрів;
- функція повертає своє значення і може використовуватися у вигляді **name** (**Список\_параметрів**) в математичних виразах;
- всі змінні, що є в тілі файлу-функції, є локальними, тобто діють тільки в межах тіла функції;
- файл-функція є самостійним програмним модулем, який спілкується з іншими модулями через свої вхідні і вихідні параметри;
- правила висновку коментарів ті ж, що у файлів-сценаріїв;
- файл-функція служить засобом розширення системи MATLAB;



- при виявленні файлу-функції він компілюється і потім виконується, а створені машинні коди зберігаються в робочій області системи MATLAB;
- остання конструкція `var=вираз` вводиться, якщо вимагається, щоб функція повертала результат обчислень.

Наведена форма файлу-функції характерна для функції з одним вихідним параметром. Якщо вихідних параметрів більше, то вони вказуються у квадратних дужках після слова `function`. При цьому структура модуля має наступний вигляд:

```
function [var1,var2,...]=f_name(Список_параметрів)
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами
var1=вираз
var2=вираз
```

### Приклад 1

Створення файлу-функції `kolo.m`, який як вхідний параметр приймає значення радіусу кола, а як вихідні параметри видає довжину кола і площу круга, ним обмеженого:

```
function [dlina,ploschad]=kolo(r)
% Розрахунок довжини кола і площі круга
% Вхідним параметром є радіус
p=2*pi*r;
s=(r^2)*pi/2;
dlina=p;
ploschad=s;
```

Далі наведений приклад використання функції `kolo`:

```
>> x=5
x =
    5
>> [dl,pl]=kolo(x)
dl =
   31.4159
pl =
   39.2699
```

Якщо функція використовує велику кількість вхідних параметрів, то передати їх у функцію можна не як аргументи, а назвавши їх як **глобальні змінні**. Значення глобальних змінних будуть визначені у зовнішньому описі і сприйняті функцією. Для визначення змінної як глобальної використовується команда **global**.

### Порядок виконання роботи

1. Створіть файл-функцію, що реалізує розрахунок складної функції. Вхідним аргументом є значення  $x$ , а вихідним – вектор, перший елемент якого це значення функції ( $y$ ), а другий – номер умови, за якою відбувався вибір конкретної підфункції. Складові функції вибрати з табл. 12.1, відповідно до варіанту.
2. Створіть вектор значень аргументу  $x$  (10-13 значень), щоб він містив значення, які потрапляють в кожний діапазон, заданий умовами аргументу складової функції.
3. Створіть скрипт (m-файл сценарій), в якому використовується створена функція. Використовуючи оператори циклу отримаєте вектор значень функції  $y$ , що відповідають елементам вектора  $x$ , і вектор  $nf$  номерів діапазону, що відповідають заданим аргументам.
4. Виведіть у командному вікні вектори  $x$ ,  $y$ ,  $nf$ , а також виведіть їх у вигляді таблиці з шапкою:

Argument                     $y$                     Nom fcn

Виведення рядків у таблиці можна організувати, використовуючи в циклі команду **disp**.

Таблиця 12.1

### Складові функції

№ варіанту	Функція
1	$y = \begin{cases} 1,2 \cdot \sin(\arccos(1/x)) \cdot \frac{ch(\sqrt{3-x})}{e^{-0.3x^2+1}}, & \text{при } -\infty < x < -4 \\ \ln( 3 \cdot \cos(0.6x^5) ) + x \cdot \log_2 x, & \text{при } -4 \leq x < 2 \\ \frac{\sqrt[3]{3 \cdot x^4 - 1.9 \cdot x}}{x-1}, & \text{при } 2 \leq x < \infty \end{cases}$
2	$y = \begin{cases} sh^4(x/2) - \frac{\sin(0.5 \cdot \pi x)}{\cos(\pi \cdot x) + 1.2}, & \text{при } -\infty < x \leq -\pi/2 \\ \sqrt{ 1.5 \cdot x^3 + 0.4 } - 2.3 \cdot arctg(x), & \text{при } -\pi/2 < x < \pi/2 \\ 4 \lg \frac{x^2}{x+1} + e^{-0.17 \cdot x}, & \text{при } \pi/2 \leq x < \infty \end{cases}$

3	$y = \begin{cases} x^2 \cdot \sin(0.3 \cdot x^3 + 2.1 \cdot x) + \cos(2/x), & \text{npu } -\infty < x < 0 \\ 2 \cdot \sqrt{x} + ch \frac{e^{4x}}{x+0.5}, & \text{npu } 0 \leq x < 3.5 \\ \sin(2.8 \cdot \ln(3^{\cos(-0.15 \cdot x)})), & \text{npu } 3.5 \leq x < \infty \end{cases}$
4	$y = \begin{cases} \frac{\sqrt{3-x}}{e^{\sin(x)-2}} + th \frac{x}{x^2+1}, & \text{npu } -\infty < x < -1 \\ \ln^3(\cos(x^4) + 2), & \text{npu } -1 \leq x < 2 \\ \frac{ctg(x) -  \sin(2 \cdot x - 1) }{(x-1) \cdot \sqrt{3x}}, & \text{npu } 2 \leq x < \infty \end{cases}$
5	$y = \begin{cases} \exp(0.2 \cdot x^3) - \cos^2(\sqrt{ \sin(0.3 \cdot x)  - x}), & \text{npu } -\infty < x \leq -1 \\ \frac{2 \cdot (x+1) \cdot x^3 + \sin(0.7 \cdot \pi \cdot x + 2)}{x^2 + 1}, & \text{npu } -1 < x \leq 0 \\ ch(x^{0.5} + 3x) \cdot \ln(\cos(x/\pi) + 1.2), & \text{npu } 0 < x < \infty \end{cases}$
6	$y = \begin{cases} \frac{(x+3) \cdot e^{2.8 \cdot x}}{\cos(3\pi \cdot x + \pi/4) + 1.1}, & \text{npu } -\infty < x < -2 \\ 5.1 \cdot x^2 \cdot \lg( \cos(th(\sqrt{x+3}))  + 0.8), & \text{npu } -2 \leq x \leq 2 \\ \frac{1.7^{(3 \cdot x - 0.2)}}{(3.5 \cdot x^2 + 2)^3 \sqrt{\sin(x-2) + 1}}, & \text{npu } 2 < x < \infty \end{cases}$
7	$y = \begin{cases} 2 \cdot x^3 \cdot \cos(x \cdot e^{(0.24 \cdot x + 1)}) \cdot \sqrt{1-x}, & \text{npu } -\infty < x < 0 \\ \frac{\sin(\pi \cdot x/3) \cdot sh^2(\sqrt{x+1})}{ x^2 - 2.6 \cdot x + 3  \cdot \ln(2 \cdot x + 0.3)}, & \text{npu } 0 \leq x < 3 \\ 1.6 \cdot \sin(2 \cdot \lg(x^{-3} + e^{-3.4 \cdot x})), & \text{npu } 3 \leq x < \infty \end{cases}$
8	$y = \begin{cases} \frac{x^3 -  ch^2(1/x) - sh(3/x) }{0.8 \cdot \cos(2.3 \cdot x + \pi) + 1}, & \text{npu } -\infty < x < 0 \\ 5 \cdot \cos(\lg(x + \sqrt{x+1})) \cdot arctg(e^{-3 \cdot x}), & \text{npu } 0 \leq x < 2.5 \\ \frac{x^2 + 4}{(x+2) \cdot \ln(x+1)} + \frac{5.1 \cdot \sin^3(\pi \cdot x)}{e^{(0.3 \cdot x^2 - 2)}}, & \text{npu } 2.5 \leq x < \infty \end{cases}$

9	$y = \begin{cases} 0.7 \cos(x^2 + 2) + e^{0.8x} \cdot \lg^2(2.1 - x), & \text{при } -\infty < x < -2 \\ \operatorname{sh} \frac{\cos(2x) \cdot \sqrt{x^2 + 0.5}}{ 3 \cdot x^{-2} + 0.7 \cdot x + 2 }, & \text{при } -2 \leq x < 1.5 \\ 0.3 \cdot \ln(\cos(\arcsin(1/x^2) + 1)), & \text{при } 1.5 \leq x < \infty \end{cases}$
10	$y = \begin{cases} \exp\left(\frac{x}{\sqrt{x^2 + 1}}\right) \cdot \sin^2(\pi \cdot x / 2 + 0.7), & \text{при } -\infty < x < -1 \\ \cos( \arcsin(0.2x)  + \lg(3 - x)), & \text{при } -1 \leq x < 2 \\ \frac{\sqrt{x^2 + 7} \cdot (3x - 1)}{\sqrt[3]{\sin(x^2 - 1) + 2 \cos(x)}}, & \text{при } 2 \leq x < \infty \end{cases}$

5. Створіть файл-функцію, яка дозволяє визначити вектор вихідних значень функції  $y(x) = 10 + 5e^{-x} \cos(\omega \cdot x + 0.5)$ . Вхідними параметрами даної функції повинні бути значення частоти  $\omega$ , початкове і кінцеве значення діапазону зміни  $x$ . Крок зміни повинен дорівнювати одній сотій діапазону її зміни.
6. Розробіть скрипт, який би використовував створену функцію, і дозволяв побудувати в одних осях графіки  $y(x)$  для трьох значень  $\omega$  (табл. 12.2).

Таблиця 12.2

№ варіанту	$\omega$ рад/с	№ варіанту	$\omega$ рад/с
1	1, 5, 9	6	5, 15, 20
2	6, 8, 10	7	12, 5, 3
3	10, 3, 6	8	3, 10, 15
4	15, 5, 2	9	5, 10, 15
5	1, 3, 7	10	4, 8, 16

Діапазон зміни вхідної змінної  $0 \leq x \leq 5$  с. Графічне зображення повинно містити наступні атрибути:

- заголовок\*  $y(x) = 10 + 5 \exp(-x) \cos(\omega \cdot x + 0.5)$
- позначення осі  $x$  time (с)
- позначення осі  $y$   $y(x)$
- типи ліній неперервна, штрихова, пунктирна
- сітка нанесена.

Зберегти отримані графіки.

\*Вивести символ  $\omega$  можна з допомогою наступної послідовності: 'omega'.

### Лабораторна робота №14

# Побудова гістограм, трьохмірних графіків та застосування анімації за допомогою системи MATLAB

## Теоретичні відомості

### 1. Побудова гістограми

У прикладних розрахунках часто зустрічаються графіки, що називаються гістограмами. Найчастіше використовуються стовпчикові гістограми (або діаграми), що відображають вміст деякого вектора  $V$ . При цьому кожен елемент вектора являється стовпчиком, висота якого пропорційна значенню елемента. Стовпчики нумеруються і масштабуються по відношенню до максимального значення найбільш високого стовпчика. Виконує побудову команда (оператор) **bar(V)**.

### 2. Побудова трьохвимірних графіків

MATLAB має широкі можливості побудови трьохвимірних графіків. Розглянемо основні команди – **mesh**, **surf**, **plot3**, **contour3**. Наведемо їх короткі характеристики:

*mesh* – дає трьохвимірне зображення у вигляді сітки;

*surf* – сегменти сітки замальовуються в різні кольори;

*plot3* – через точки з координатами  $x$ ,  $y$ ,  $z$  проводяться лінії;

*contour3* – через точки з координатами  $x$  і  $y$  проводяться лінії з сталим значенням  $z$  або ізолінії.

Кожна з наведених команд має свої модифікації, з якими можна познайомитися у меню Help системи MATLAB.

Наприклад, для побудови графіка трьохвимірної поверхні і її проекції у вигляді контурного графіка на площині під поверхнею достатньо використати наступні команди:

```
[X,Y]=meshgrid([-5:0.1:5]);
```

```
Z=X*sin(X+Y);
```

```
meshc(X,Y,Z)
```

### 3. Рухомі графіки (анімація)

Для виведення на екран об'єктів, що рухаються, в MATLAB є ряд можливостей. Одна з них розрахунок послідовних станів об'єкту і збереження відповідного зображення в спеціальному форматі з подальшою прокруткою цього мультфільму. Цим займається функція **movie**.

Цікавою є можливість отримання на екрані рухомих об'єктів "на льоту", тобто в процесі обчислення. Частково цю проблему вирішують такі функції як **comet** і **comet3**. Ці функції дозволяють отримати квазі-анімацію, тобто вивести і показати в русі траєкторію однієї або декількох частинок (**comet** – на площині, а **comet3** – у тривимірному просторі), але після того, як ці траєкторії будуть

пораховані. Це буває досить корисним, але в той самий час позбавляє нас можливості оперативно втручатися в описуваний процес (змінювати які-небудь параметри розрахункової моделі).

**Приклад 1** (виведення пакету біжучих хвиль).

```
% Задання значень параметрів
t=0:0.1:100;
x=0:0.3:30;
k=1.3; w=0.9;
n=length(t);
% Розрахунок форми хвилі при t(1)=0
y=cos(k*x-w*t(1))+cos(x-t(1));
% Обчислення меж рисунка
axlim = [min(x) max(x) min(y) max(y)];
figure
% Збереження дескриптора об'єкту, що виводиться
lh = line(x,y);
% Задання кольору кривої і вибір меж
set(lh,'color','r');
axis(axlim);
% Вибір режиму стирання
set(lh,'erasemode','xor');
% Початок циклу анімації
155
for i = 2:n
% Перерахунок координат кривої
y=cos(k*x-w*t(i))+cos(x-t(i));
% Найголовніший момент – заміна координат
set(lh,'XData',x,'YData',y);
drawnow;
end
```

У наведеній вище програмі є три вузлові моменти, що забезпечують якісну анімацію.

1. По-перше, це підготовка системи координат і створення дескриптора виведеного об'єкта з присвоєнням властивості **'EraseMode'** (режим стирання) значення **'xor'**. Зверніть увагу на цей параметр. Річ у тому, що значення цієї властивості за замовчуванням – **'none'** і при цьому виведений раніше об'єкт не стирається, а залишається на екрані. Можливе здійснення анімації і за допомогою режиму стирання **'background'**, причому анімація відбуватиметься швидше. Відмінності між цими двома режимами найкраще побачити, якщо

ввести додатково перед циклом анімації команду **grid on** (намалювати координатну сітку). При режимі **xor** сітка залишається після «проходження» хвилі по екрану, а при використуванні режиму стирання **background** сітка витирається там, де пройшла хвиля.

2. Фактично перемальовування об'єкту відбувається при зміні значень властивостей об'єкту 'XData' і 'YData'.

3. Важливим чинником безперервного оновлення картинки є команда **drawnow** яка забезпечує постійне виведення оновлених значень на екран. Такий примусовий вивід забезпечують також функції **pause** і **getframe**.

### *Порядок виконання роботи*

А. Створити m-файл.

1. Побудувати гістограми до завдання:

*Припустимо, що охоронець фіксує кількість автомобілів, що в'їжджають у двір університету кожні 10 хвилин. У результаті, за 2,5 години він набрав статистику по 15-ти десятихвилинних інтервалах.*

Представимо дані у вигляді рядка матриці [1 1 2 1 2 3 1 2 3 4 1 2 3 4 5] (через пропуск). Побудуємо розподіл за числом автомобілів у вигляді гістограми. Для цього скористаємося командою **hist**. Текст програми має вигляд:

```
cars=[1 1 2 1 2 3 1 2 3 4 1 2 3 4 5];  
hist(cars,5), grid;  
xlabel('Number of cars');  
ylabel('Number of intervals');  
title('Cars in yard', 'FontSize',14);
```

2. Замінити команду **hist** на **bar** і змінити підписи осей на український шрифт:

```
x=1:5; y=[5 4 3 2 1];  
bar(x,y), grid  
set(gca,'FontName','Arial Cyr');  
xlabel('Число автомобілів ');  
ylabel('Число інтервалів');  
title('Розподіл за числом автомобілів ', 'FontSize',14);
```

3. Припустимо, що є дані про автомобілі, що в'їжджали у двір університету протягом трьох днів. Їх можна представити у вигляді матриці з трьох рядків:

```
cars=[1 1 2 1 2 3 1 2 3 4 1 2 3 4 5; 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1; 2 1 2 3 1  
2 3 4 1 2 3 4 5 1 2];  
bar(cars), grid
```

```
set(gca,'FontName','Arial Cyr');
xlabel('Число автомобілів '); ylabel('Число інтервалів');
```

4. Представити результати у вигляді трьохмірного графіка:

```
cars=[1 1 2 1 2 3 1 2 3 4 1 2 3 4 5; 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1; 2 1 2 3 1
2 3 4 1 2 3 4 5 1 2];
bar3(cars), grid
set(gca,'FontName','Arial Cyr');
xlabel('Число автомобілів'); ylabel('Дні');
zlabel('Число інтервалів');
```

5. Скомпонувати два останні графіки в один рисунок за допомогою команди **subplot**:

```
cars=[1 1 2 1 2 3 1 2 3 4 1 2 3 4 5; 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1; 2 1 2 3 1
2 3 4 1 2 3 4 5 1 2];
subplot(1,2,1); bar(cars);
set(gca,'FontName','Arial Cyr');
title('Рис.1');
subplot(1,2,2); bar3(cars);
set(gca,'FontName','Arial Cyr');
title('Рис.2');
```

6. Обчислити суму елементів кожного рядка (суму автомобілів кожного дня протягом трьох днів) і побудувати кругову діаграму (вона будується за допомогою команди **pie**):

```
cars=[1 1 2 1 2 3 1 2 3 4 1 2 3 4 5
1 2 1 2 3 1 2 3 4 1 2 3 4 5 1
2 1 2 3 1 2 3 4 1 2 3 4 5 1 2];
sm=sum(cars');
pie(sm)
```

## Б. Створити m-файл.

### Трьохмірні графіки.

*Побудувати графік функції  $z = x^2 + y^2$  для  $0 \leq x \leq 10$  і  $0 \leq y \leq 10$ .*

Для цього площина покривається координатною сіткою і значення функції визначається в її вузлах. Густина сітки визначається кроком розбиття координатних осей (візьмемо рівним 2). Сама сітка задається командою **meshgrid**.

```
[X, Y] = meshgrid(0:2:10, 0:2:10);
Z=3*X.^2+Y.^2;
subplot(2,2,1); mesh(X, Y, Z);
title ('mesh'); xlabel('x'); ylabel('y')
```



```

subplot(2,2,2); plot3(X, Y, Z);
title ('plot3'); xlabel('x'); ylabel('y')
subplot(2,2,3); surf(X, Y, Z);
title ('surf'); xlabel('x'); ylabel('y')
subplot(2,2,4); contour3(X, Y, Z);
title ('contour'); xlabel('x'); ylabel('y')

```

**В. Створити m-файл.**

**Рухомі графіки (анімація).**

*Розглянемо рух пакету хвиль.*

```

% Задання значень параметрів
t=0:0.1:100;
x=0:0.3:30;
k=1.3; w=0.9;
n=length(t);
% Розрахунок форми хвилі при t(1)=0
y=cos(k*x-w*t(1))+cos(x-t(1));
% Встановлення границь рисунка
axlim = [min(x) max(x) min(y) max(y)];
figure
% Збереження дескриптора об'єкта, що виводиться
lh = line(x,y);
% Задання кольору кривої і вибору меж
set(lh,'color','r');
axis(axlim);
% Вибір режиму стирання
set(lh,'erasemode','xor');
% Початок циклу анімації
for i = 2:n
% Перерахунок координат кривої
y=cos(k*x-w*t(i))+cos(x-t(i));
% Найголовніший момент – заміна координат
set(lh,'XData',x,'YData',y);
drawnow;
end

```

## Моделювання фігур Ліссажу за допомогою системи MATLAB

### Завдання

Побудувати на екрані графік функції, заданої параметрично:

$$\begin{cases} x = a_1 \cdot \cos(\omega_1 t) \\ y = a_2 \cdot \cos(\omega_2 t) \end{cases}$$

Якщо відношення  $\omega_1/\omega_2$  – число раціональне, то така крива називається фігурою Ліссажу. Такі криві ви неодноразово бачили на екрані осцилографа.

Наведений далі текст є протоколом роботи з системою MATLAB, отриманий при розв'язуванні поставленої задачі в режимі безпосереднього обчислення. Для отримання протоколу і збереження його у вигляді файлу використовується команда **diary** <ім'я\_файлу>. Такий протокол може бути корисний для подальшого аналізу рішення, а також при невеликих доробках може бути використаний як основа для написання програми-сценарію для подальшого розв'язування подібних задач.

### Порядок виконання роботи

1. Створити М-файл.
2. Побудувати графік даної залежності  $y(x)$  при зміні  $t$  від 0 до 10 з кроком 0.1:

```
t=0:0.1:10;  
x=a1*cos(w1*t);  
y=a2*cos(w2*t);  
plot(x,y);
```

3. Використовуючи можливість повторювати виконання операторів без їх повторного набору (клавіша  $\uparrow$ ), виконати обчислення і побудову фігур Ліссажу для різних відношень  $\omega_1/\omega_2$  і для різних значень амплітуд  $a_1$  і  $a_2$ :

```
%Інший варіант розрахунку фігур Ліссажу  
% Задання коефіцієнтів  
a1=1.2; a2=1.0; w2=1.0;  
% Задання значень вектора t  
t=0:0.1:15; x=a1*cos(w2*t);  
% Задання різних частот w1  
w1=1.25:0.25:2.0;  
%Цикл виводу різних графіків на одному аркуші  
for k=1:4,  
y=a2*cos(w1(k)*t);  
% Задання вектора-рядка виведення підписів  
s=['w1/w2=' num2str(w1(k))];
```

```
% Виведення графіків в різних ділянках екрану  
subplot(2,2,k); plot(x,y); title(s);  
end;
```

4. Включити початкові фази  $\varphi_1$  і  $\varphi_2$  в аргументи косинусів, тобто  $\cos(\omega_1 t)$  замінити на  $\cos(\omega_1 t + \varphi_1)$ , а  $\cos(\omega_2 t)$  – на  $\cos(\omega_2 t + \varphi_2)$  (самостійно внести зміни у програму).
5. У програмі замінити команду `plot` на `comet`, що дозволить зобразити ділянку кривої, що "біжить" і проаналізувати скільки разів точка пробігає криву при зміні часу в заданому інтервалі.

**Лабораторна робота №16**  
**Моделювання коливань математичного маятника**  
**за допомогою системи MATLAB**

***Теоретичні відомості***

Задачі про коливання зустрічаються у всіх областях фізики. Багато в чому коливання абсолютно різних фізичних об'єктів схожі між собою. Найпростіші приклади – малі коливання маятника і електричні коливання в електромагнітному контурі, який складається з конденсатора і котушки.

Такий рух маятника добре відомий – це гармонічні коливання. Закон руху можна записати у вигляді  $x = a \cos(\omega_0 t + \varphi_0)$ , де  $\omega_0$  – частота коливань,  $a$  – амплітуда,  $\varphi_0$  – початкова фаза. (Кут відхилення маятника ми позначили тут  $x$ . Далі вважатимемо масу важка  $m$ , довжину маятника  $l$  та  $\omega_0$  рівними одиниці). Малі коливання описуються рівнянням

$$\ddot{x} + x = 0,$$

лінійним щодо функції  $x$ , тому їх зазвичай називають лінійними.

У цій роботі ми досліджуватимемо рух математичного маятника при великих кутах відхилення, спеціально звертаючи увагу на відмінність від законів малих коливань. Подібні відхилення називають нелінійними ефектами.

**Вільні коливання**

Результати дослідження руху маятника зручно представити у вигляді набору кривих на площині  $(x, p)$ , де  $p = \dot{x}$  – швидкість зміни кута. Площина  $(x, p)$  називається фазовою площиною, змінна  $p$  – імпульсом, а криві, які визначаються параметрично законом руху як  $x = x(t)$ ,  $p = p(t)$ , – фазовими траєкторіями.

Фазова траєкторія визначається, наприклад, початковими значеннями координати  $x(0)$  і імпульсу  $p(0)$ .

Фазові траєкторії лінійного осцилятора є еліпсами, заданими законом збереження енергії. Для математичного маятника це справедливо при малих кутах відхилення, у загальному випадку, при великих значеннях кутів відхилення рух математичного маятника буде складнішим.

Кут відхилення маятника достатньо задавати в деяких кінцевих межах, наприклад  $-\pi \leq x < \pi$ . При цьому слід уявляти, що точки фазової площини  $(-\pi, p)$  і  $(\pi, p)$  ототожені, іншими словами, прямі  $x = \pi$  і  $x = -\pi$  співпадають одна з одною так, що із смуги

$$\begin{aligned} -\pi \leq x < \pi, \\ -\infty < p < \infty \end{aligned}$$

отримався циліндр. Якщо маятник робить один або декілька обертів, то точка, що зображає його стан, рухається по кривій, що обвиває цей циліндр.

Відмінність коливань з великою амплітудою від малих коливань зводиться до того, що закон зміни кута з часом відрізняється від гармонічного, а частота їх залежить від амплітуди.

При деякій енергії коливання змінюються обертянням.

Вільний рух математичного маятника без тертя описується диференціальним рівнянням

$$\ddot{x} + \sin x = 0. \quad (15.1)$$

Є аналітичний розв'язок цього рівняння (досить складний), проте ми досліджуватимемо рух маятника чисельно.

Запишемо це диференціальне рівняння у вигляді системи рівнянь першого порядку

$$\begin{aligned} \dot{x} &= p, \\ \dot{p} &= -\sin x. \end{aligned} \quad (15.2)$$

Основна ідея чисельного розрахунку надзвичайно проста: знаючи значення координати і швидкості у момент часу  $t$ , можна приблизно знайти їх значення через малий проміжок  $\Delta t$

$$\begin{aligned} x(t + \Delta t) &= x(t) + \dot{x}(t)\Delta t, \\ p(t + \Delta t) &= p(t) + \dot{p}(t)\Delta t. \end{aligned}$$

Багаторазово повторюючи такі обчислення ми знайдемо залежність  $x(t)$  і  $p(t)$ . Більшої точності можна досягти, практично не ускладнюючи розрахунки, якщо використовувати так звану обчислювальну схему з переступанням. У цій схемі обчислюються значення координат в моменти часу

$$t - \frac{1}{2}\Delta t, \quad t + \frac{1}{2}\Delta t, \quad t + \frac{3}{2}\Delta t, \quad t + \frac{5}{2}\Delta t, \quad \dots$$

а значення швидкостей в моменти часу

$$t, \quad t + \Delta t, \quad t + 2\Delta t, \quad t + 3\Delta t \dots$$

Дана обчислювальна схема і буде використана нами в програмі.

Щоб відхилення від точного розв'язку рівняння (15.1) було невеликим, крок за часом  $\Delta t$  повинен бути достатньо малим.

### ***Порядок виконання роботи***

1. Створити М-файл.
2. Ввести початкову програму для розв'язування нашої задачі:

```
% Фазова траєкторія математичного маятника
% Задання початкових значень
x1=2.2; % Початкова координата
p1=0.0; % Початковий імпульс
dt=0.025; % Крок за часом
axis([-pi pi -pi pi]); % Задання діапазону осей
```

```

hl=line(x1,p1); % Задання дескриптора лінії
% Задання параметрів лінії, що виводиться
set(hl,'EraseMode','none','LineStyle',':','Color','r');
grid on; % Задання координатної сітки
pause;
while 1 % Нескінченний цикл
% Основний алгоритм розрахунку
x2=x1+p1*dt;
p2=p1-sin(x2)*dt;
% Склеювання граничних умов
if x2> pi
x2=x2-2*pi;
end;
if x2< -pi
x2=x2+2*pi;
end;
% Вивід чергової ділянки фазової траєкторії
set(hl,'XData',x2,'YData',p2);
% Переприсвоєння початкових значень
x1=x2;p1=p2;
end;

```

(Тут крок за часом  $\Delta t$  позначений  $dt$ ,  $\pi = \pi$ .)

Після запуску програма відкриє графічне вікно і намалює в ньому одне вікно з системою координат, осями і координатною сіткою, після чого перейде в режим очікування (команда `pause`). Для продовження розрахунку необхідно натиснути будь-яку клавішу, і програма почне вивід фазової траєкторії. Оскільки використовується нескінченний цикл, виконання програми можна перервати клавішею `Ctrl+C`. Для повторного прогону з іншими параметрами необхідно ввести відповідні зміни в текст програми і перезапустити її.

- Зменшуючи крок  $dt$ , переконайтеся, що фазова траєкторія відтворюється при цьому без змін протягом декількох періодів (зрозуміло, з тією точністю, яку допускає графічне зображення). Збільшуючи крок досягніть такої його величини, щоб з'явилося явно видиме спотворення форми фазової траєкторії. Зобразіть декілька різних фазових траєкторій.

**Лабораторна робота №17**  
**Моделювання руху частинок у центральному полі**  
**за допомогою системи MATLAB**

**Теоретичні відомості**

Закон руху частинки  $r(t)$  у заданому полі  $U(r)$  повністю визначається рівнянням руху

$$m\ddot{\vec{r}} = -\frac{\partial U}{\partial \vec{r}} = \vec{F}(\vec{r}) \quad (16.1)$$

і початковими умовами  $\vec{r}(0) = \vec{r}_0$ ,  $\vec{v}(0) = \vec{v}_0$ . У деяких випадках диференціальне рівняння (16.1) можна розв'язати аналітично, в інших – можливе якісне дослідження. Проте в достатньо складних полях  $U(\vec{r})$  доводиться застосовувати чисельні розрахунки.

**Траєкторія фінітного руху**

У даній роботі вивчається, в першу чергу, рух у центральних полях. Щоб орієнтуватися в очікуваних результатах, необхідно пригадати якісні способи дослідження руху в центральному полі. В цьому випадку зберігається момент імпульсу  $\vec{M} = m[\vec{r}\vec{v}]$ , а траєкторія лежить в площині перпендикулярній вектору  $\vec{M}$ . Ми обмежимося випадком, коли частинка не віддаляється від центру необмежено (*фінітний рух*) і не падає в центр поля. Тоді траєкторія виявляється розташованою між двома колами  $r_{\min} \leq r \leq r_{\max}$  і заповнює це кільце, проходячи як завгодно близько до будь-якої його точки.

Фінітний рух у кулонівському полі  $U_0 = -\alpha/r$ , на відміну від цього загального випадку, є рух по замкнутій траєкторії – по еліпсу з напівосями: великою  $a = \alpha/2|E|$  і малою  $b = M/\sqrt{2m|E|}$ .

Найпростіший метод чисельного розв'язування рівняння (1) (метод Ейлера) полягає в тому, щоб, вибравши малий "крок"  $\Delta t$ , безпосередньо визначити прирости координат і швидкості рівностями

$$\begin{aligned} \Delta \vec{r} &= \vec{v} \Delta t, \\ \Delta \vec{v} &= \frac{1}{m} \vec{F}(\vec{r}(t)) \Delta t. \end{aligned} \quad (16.2)$$

Багаторазово повторюючи таке обчислення, отримаємо ряд послідовних значень координат і швидкості.

Набагато більшої точності можна добитися, якщо замість (16.2) використовувати наступну схему

$$\Delta \vec{v} = \frac{1}{m} \vec{F}(\vec{r}(t + \Delta t)), \quad (16.3)$$

тобто при обчисленні сили використовувати нові значення координат у момент часу  $t + \Delta t$ .

### ***Порядок виконання роботи***

1. Створити М-файл.
2. Ввести програму для обчислення і зображення на екрані траєкторії частинки в кулонівському полі (в програмі  $\alpha$  означає  $\alpha/m$ , а  $dt = \Delta t$ ,  $v = (v_x, v_y)$ ,  $F/m = a = (a_x, a_y)$ ):

% Задання початкових значень

Alpha=1; dt=0.025;

x1=2.5; y1=0;

vx1=0; vy1=0.25;

E<sub>max</sub>=-1e18;

E<sub>min</sub>=1e18;

r=sqrt(x1<sup>2</sup>+y1<sup>2</sup>); % Відстань до центру

E1=(vx1<sup>2</sup>+vy1<sup>2</sup>)/4 ...

- Alpha/r; % Повна енергія

hl=line(x1,y1); % Задання дескриптора лінії

a=Alpha/(2\*abs(E1)); % Велика напіввісь

b=r\*vy1/sqrt(2\*abs(E1)); % Мала напіввісь

axis([-2\*a 2\*a -1.2\*b 1.2\*b]); % Задання масштабу

set(hl,'EraseMode','none','Color','g');

ha=gca;

% Задання розмітки осей

set(ha,'XTick',[-a 0 a],'YTick',[-b 0 b]);

grid on; % Введення сітки

pause % Пауза для обов'язкового виводу

while 1 % Нескінченний цикл

x2=x1+vx1\*dt; % Нові координати

y2=y1+vy1\*dt; %

r=sqrt(x2<sup>2</sup>+y2<sup>2</sup>); % Відстань до центру

A = Alpha/r<sup>2</sup>; % Модуль прискорення

ax=-A\*x2/r; % Компоненти вектора

ay=-A\*y2/r; % прискорення

vx2=vx1+ax\*dt; % Нові швидкості

vy2=vy1+ay\*dt; %

E2=((vx2+vx1)<sup>2</sup>+(vy2+vy1)<sup>2</sup>)/8 ...

-Alpha/r; % Повна енергія

if E2 > E<sub>max</sub> % Межі зміни

E<sub>max</sub>=E2; % повної енергії



```

end;
if E2 < Emin
Emin = E2;
end;
% Оновлення даних на графіку
set(hl, 'XData', [x1 x2], 'YData', [y1 y2]);
% Переприсвоєння
x1 = x2;
y1 = y2;
vx1 = vx2;
vy1 = vy2;
end; % Кінець циклу 'while 1'

```

Оскільки використовується нескінченний цикл виконання програми можна перервати клавішею **Ctrl+C**. Слід пам'ятати, що в **MATLAB** всі дані зберігаються в робочій області до тих пір, поки Ви не вийдете з командного вікна. Тому після першого прогону програми можна зупинити вивід командою **Ctrl+C**, поміняти необхідні параметри (наприклад, крок по часу), задати режим **hold on** (малювати поверх), і знову запустити програму. Графік буде виведений на той же рисунок.

3. Дослідити вплив величини кроку  $\Delta t$  на точність розрахунку траєкторії. При дуже великому значенні  $\Delta t$  виникає прецесія орбіти, пов'язана з неточністю розрахунку. Для оцінки точності обчислень виведіть на екран значення повної енергії залежно від координати  $x$ . Це краще зробити, розбивши область графічного виведення на дві підобласті – для побудови орбіти і для побудови енергії. Це можна зробити за допомогою функції **subplot**. Відмітити на траєкторії іншим кольором точки найбільшого і найменшого віддалення від центру.

**Лабораторна робота №18**  
**Моделювання розподілу Максвелла молекул ідеального газу за швидкостями у системі MATLAB**

***Теоретичні відомості***

Функція густини вірогідності цього розподілу Максвелла молекул ідеального газу за швидкостями має вигляд

$$f(v) = \frac{4}{\sqrt{\pi}} \left( \frac{m}{2kT} \right)^{\frac{3}{2}} v^2 \cdot \exp\left(-\frac{mv^2}{2kT}\right),$$

де  $v$  – швидкість частинки газу,  $m$  – її маса,  $T$  – температура газу,  $k = 1,38 \cdot 10^{-23} \frac{\text{Дж}}{\text{К}}$  – стала Больцмана. Функція  $f(v)$  є нормованою:

$$\int_0^{\infty} f(v) dv = 1.$$

***Порядок виконання роботи***

1. Створити М-файл.
2. Ввести програму та побудувати графік функції  $f(v)$  для газу, що складається з молекул водню  $H_2$  з  $m = 3,32 \cdot 10^{-27} \text{ кг}$  при  $T = 350 \text{ К}$ :

```
% Розподіл Максвелла молекул за швидкостями
m=3.32e-27; k=1.38e-23; T=250;
% Задаються кількість кроків і крок по осі ординат
n=2500; step=2;
% Початкові значення швидкості і функції розподілу
v(1)=0; f(1)=0;
a=m/(2*k*T);
b=4/sqrt(pi);
% В циклі створюються масиви швидкості і функції розподілу
for i=2:n
v(i)=v(i-1)+step;
f(i)=b*a^1.5*v(i)^2*exp(-a*v(i)^2);
end
plot(v,f), grid
set(gca,'FontName','Arial Cyr');
title('Розподіл Максвелла','FontSize',14)
xlabel('Швидкість');
ylabel('Густина ймовірності');
v1=sqrt(8*k*T/pi/m)
v2=sum(v.*f)/sum(f)
```

```
sm=sum(f*step)
```

$v_1$  – середня швидкість частинок газу, обчислена за формулою  $\langle v \rangle = \sqrt{\frac{8kT}{\pi m}}$ ;  $v_2$  – та ж величина, але визначена шляхом усереднення швидкості по розрахунковій кривій; sm – площа під кривою розподілу, яка близька до одиниці.

Як видно з обчислень існує незначна відмінність швидкостей  $v_1$  і  $v_2$ . При форматі Format long величина sm буде відмінна від одиниці, що пов'язано з похибкою чисельних розрахунків. Для збільшення точності розрахунків потрібно збільшити число кроків n і зменшити їх довжину step. Виконайте цю операцію.

**Примітка:** В цій програмі використовується операція циклу  
*for i=2:n*  
.  
*end*

*Тобто командні рядки, що містяться між вказаними рядками, будуть виконані циклічно для змінної i, що змінюється у вказаних межах з кроком 1 за замовчуванням.*

3. Ввести програму і побудувати на одному графіку розподіл Максвелла для молекул водню і двох значень температур (300 і 500 K).

```
% Розподіл Максвелла молекул за швидкостями  
m=3.32e-27; k=1.38e-23; T1=300; T2=500;  
% Задаються кількість кроків і крок по осі ординат  
n=3000; step=2;  
% Початкові значення швидкості і функції розподілу  
v(1)=0; f1(1)=0; f2(1)=0;  
% В циклі створюються масиви швидкості і функції розподілу  
for i=2:n  
v(i)=v(i-1)+step;  
f1(i)=4/sqrt(pi)*(m/(2*k*T1))^1.5*v(i)^2*exp(-m*v(i)^2/(2*k*T1));  
f2(i)=4/sqrt(pi)*(m/(2*k*T2))^1.5*v(i)^2*exp(-m*v(i)^2/(2*k*T2));  
end  
plot(v,f1,v,f2), grid  
set(gca,'FontName','Arial Cyr');  
title('Розподіл Максвелла','FontSize',14)  
xlabel('Швидкість');  
ylabel('Густина ймовірності');  
hgt=gtext('T=300K'); hgt=gtext('T=500K');
```

*Примітка: Написи на двовимірних графіках робляться за допомогою команди gtext. Потрібно вказати курсором на місце в основному малюнку. Результат передається на графік в тексті m-книги.*

Ці розподіли, по суті, є лініями постійної температури тривимірного графіка залежності  $f(v,T)$ .

4. Ввести програму і побудувати залежність для молекул водню в широкому інтервалі швидкостей і температур:

```
m=3.32e-27; k=1.38e-23;
p=sqrt(pi); a=4/p; b=m/2/k;
[v, t] = meshgrid(30:30:3000, 203:3:500);
z3=v.^2; z4=t.^(-1);
z1=log(a*b^1.5*z4.^1.5); z2=log(z3);
z5=b*z3.*z4; z6=z1+z2-z5;
% Початкова функція логарифмується, а потім експонується
z=exp(z6);
mesh(v, t, z)
set(gca,'FontName','Arial Cyr');
title('Розподіл Максвелла','FontSize',14)
xlabel('Швидкість (м/с)'); ylabel('T(K)')
```

5. Ввести програму і побудувати на одному графіку тривимірні розподіли для водню і кисню:

```
clear; k=1.38e-23; p=sqrt(pi); a=4/p;
% Задається координатна сітка, у вузлах якої визначається значення
функції
[v, t] = meshgrid(30:30:3000, 203:3:500);
for i=1:2
    if i==1; m=3.32e-27; end ;
    if i==2; m=8*3.32e-27; end ;
b=m/2/k; z3=v.^2; z4=t.^(-1); z1=log(a*b^1.5*z4.^1.5);
z2=log(z3); z5=b*z3.*z4; z6=z1+z2-z5; z=exp(z6);
mesh(v, t, z); hold on; mesh(v, t, z)
end
set(gca,'FontName','Arial Cyr');
title('Розподіл Максвелла','FontSize',14)
xlabel('Швидкість (м/с)'); ylabel('T(K)')
% Написи до поверхонь на малюнку
text(400,350,1.55e-3,'O_2')
```

```
text(300,100,1.3e-3,'H_2')
```

*Примітка: Команда text дозволяє підписувати як двовимірні, так і тривимірні графіки. Числа відповідають координатам. Команда hold on "утримує" першу залежність при побудові другої.*

## ЗМІСТ

Вступ . . . . .	3
Програма навчальної дисципліни . . . . .	4
Перелік тем лабораторних робіт . . . . .	7
Лабораторна робота № 1. Віртуальні лабораторії на PhET, що демонструють різні явища в галузі фізики . . . . .	8
Лабораторна робота № 2. Моделювання рівномірного руху тіла . . . . .	8
Лабораторна робота № 3. Моделювання рівноприскореного руху тіла . . . . .	11
Лабораторна робота № 4. Моделювання коливального руху на прикладі математичного маятника . . . . .	14
Лабораторна робота № 5. Моделювання руху тіла, кинутого під кутом до горизонту . . . . .	17
Лабораторна робота № 6. Розв'язування диференціального рівняння другого порядку чисельним методом. Коливання пружинного маятника . . . . .	29
Лабораторна робота № 7. Моделювання падіння тіла з урахуванням опору середовища . . . . .	26
Лабораторна робота № 8. Основні поняття і принципи роботи в Autocad. Підготовка робочого середовища . . . . .	31
Лабораторна робота № 9. Засоби креслення. Засоби редагування креслень . . . . .	34
Лабораторна робота № 10. Нанесення розмірів на креслення. Додаткові засоби формування креслень . . . . .	38
Лабораторна робота № 11. Знайомство з режимом "прямих обчислень" пакета прикладних програм MATLAB . . . . .	40
Лабораторна робота № 12. Знайомство з графічними можливостями пакету прикладних програм MATLAB . . . . .	53
Лабораторна робота № 13. Засоби програмування системи MATLAB . . . . .	61
Лабораторна робота № 14. Побудова гістограм, трьохмірних графіків та застосування анімації за допомогою системи MATLAB . . . . .	69
Лабораторна робота № 15. Моделювання фігур Ліссажу за допомогою системи MATLAB . . . . .	74
Лабораторна робота № 16. Моделювання коливань математичного маятника за допомогою системи MATLAB . . . . .	76
Лабораторна робота № 16. Моделювання руху частинок у центральному полі за допомогою системи MATLAB . . . . .	79
Лабораторна робота № 18. Моделювання розподілу Максвелла молекул ідеального газу за швидкостями у системі MATLAB. . . . .	82