

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"**

Рекомендовано до видання Вченою радою ФЕА

Протокол №11 від 23.06. 2014 р.

**ОБЧИСЛЮВАЛЬНА ТЕХНІКА  
ТА АЛГОРИТМІЧНІ МОВИ**

**Методичні вказівки до практичних занять  
для студентів за спеціальністю**

**7/8.05070107 "Нетрадиційні та відновлювані джерела енергії"**

**ЧАСТИНА 1. Мови програмування C++ та MATLAB**

Київ  
2015

«Обчислювальна техніка та алгоритмічні мови. Частина 1. Програмування математичних задач». Методичні вказівки до практичних занять для студентів за спеціальністю 7/8.05070107 "Нетрадиційні та відновлювані джерела енергії" /укл. О.Ю. Гаєвський – К.: ФЕА НТУУ «КПІ», 2015. – 34с.

Затверджено  
на засіданні кафедри відновлюваних джерел енергії  
Протокол № 13/2 від 29.05.2014 р.

**Укладач:** Гаєвський Олександр Юлійович, д.ф.-м.н, проф. каф.  
відновлюваних джерел енергії

**Рецензент:** Островерхов Микола Якович, д.т.н., проф., зав. каф.  
теоретичних основ електротехніки

**Відповідальний редактор:** Кудря Степан Олександрович, д.т.н., проф.  
зав. каф. відновлюваних джерел енергії

Оригінал-макет в авторській редакції

# Зміст

Практична робота 1. «Основні конструкції мови C++. Арифметичні операції».....	6
Теоретичні відомості.....	6
Завдання.....	7
Методичні вказівки .....	7
Практична робота 2. «Основні конструкції мови C++. Розгалуження і цикли» .....	8
Теоретичні відомості.....	8
Керуючі логічні вирази.....	8
Логіка умови в операторах циклів.....	9
Завдання.....	9
Методичні вказівки .....	10
Практична робота 3. «Функції в C/C++» .....	11
Теоретичні відомості.....	11
Завдання.....	11
Методичні вказівки .....	11
Практична робота 4. «Робота з рядками» .....	12
Теоретичні відомості.....	12
Створення заголовного файлу та файлу реалізації.....	12
Завдання.....	13
Методичні вказівки .....	13
Практична робота 5. «Основні операції з матрицями в MATLAB» .....	14
Теоретичні відомості.....	14
Пакет MATLAB для математичних обчислень.....	14
Конструювання матриць в MATLAB.....	14
Арифметичні операції в MATLAB.....	15
Завдання.....	16
Методичні вказівки .....	16
Практична робота 6. «Функції у MATLAB» .....	17
Теоретичні відомості.....	17
Елементарні математичні функції MATLAB.....	17
Файли-сценарії і файли-функції в MATLAB.....	18
Функції інтерактивного обміну даними в MATLAB.....	19
Завдання.....	20
Практична робота 7. «Алгоритми розгалуження і цикли в MATLAB» .....	21
Теоретичні відомості.....	21
Умовні оператори в MATLAB.....	21
Оператор вибору в MATLAB.....	21

Оператори циклу в MATLAB.....	22
Постановка задачі.....	22
Варіанти.....	23
Методичні вказівки.....	24
Практична робота 8. «Двовимірна графіка в MATLAB».....	26
Теоретичні відомості.....	26
Завдання.....	27
Методичні вказівки.....	28
Практична робота 9. «Тривимірна графіка в MATLAB».....	29
Теоретичні відомості.....	29
Завдання.....	29
Практична робота 10. «Ітераційні процеси для розв'язання нелінійних рівнянь».....	31
Теоретичні відомості.....	31
Метод Ньютона розв'язання нелінійних рівнянь.....	31
Функції MATLAB для розв'язання нелінійних рівнянь.....	31
Постановка задачі.....	32
Варіанти.....	32
Методичні вказівки.....	33
Література.....	34

## Вступ

Діяльність сучасного інженера неможлива без кваліфікованого використання обчислювальної техніки для збору й обробки даних, для завдань проектування й моделювання, для автоматизації технологічних процесів. Особливе значення в підготовці майбутніх інженерів має оволодіння студентами знаннями й навичками з алгоритмізації і програмування. Таку мету ставить навчальний курс «Обчислювальна техніка та алгоритмічні мови», якій складається з лекцій, практичних занять, лабораторних робіт і самостійної роботи студентів (СРС).

Виклад основ програмування в курсі «Обчислювальна техніка та алгоритмічні мови» доцільно проводити з використанням мови C++ та мови пакету MATLAB. Це сучасні та потужні мови програмування, які застосовуються для математичних та інженерних розрахунків, для створення прикладних програм. Крім того, знання цих мов набуває все більшої популярності, підвищує затребуваність фахівців і сприяє ранній професійній орієнтації.

Для успішного застосування навичок програмування в інженерній діяльності необхідне узгодження курсу програмування із загальнотехнічними й спеціальними дисциплінами. Тому в даній частині методичних вказівок велика увага приділена практиці складання програм для розв'язування різних математичних задач: обчислення арифметичних виразів, обчислення і застосування функцій, використання розкладів у ступеневі ряди, операції з векторами й матрицями.

Виконуючи завдання відповідно даного посібника студенти повинні навчитися використовувати вбудовані типи даних, операції й оператори, застосовувати функції введення й виведення, створювати користувальницькі функції, організовувати ітераційні обчислювальні процеси. При цьому особлива увага приділяється роботі з матрицями й векторами.

# Практична робота 1. «Основні конструкції мови C++. Арифметичні операції»

**Мета роботи:** Вивчення основних конструкцій мови C і написання простих програм на мові C. Використання арифметичних операцій, введення і виведення даних.

## Теоретичні відомості

У C++ визначено великий набір операцій над даними, які служать для побудови виразів. Вираз являє собою послідовність операндів і знаків операцій і підлягає обчисленню. В якості операндів у виразі виступають ідентифікатори змінних, константи, і рядкові літерали. Залежно від кількості операндів, що беруть участь у вираженні, розрізняють:

- унарні операції (1 операнд);
- бінарні (2 операнда);
- тернарні (3 операнда).

Дана робота присвячена вивченню арифметичних операцій і їх значенню в залежності від типу операндів:

- \* множення
- / ділення і цілочисельне ділення без остачі
- % залишок від ділення
- + додавання
- віднімання.

Знак /, який означає ділення, дає результат, що залежить від типу операндів, що беруть участь в операції. Якщо з обох сторін від цього знака стоять цілі величини (константи, змінні або їх комбінації), то він означає цілочисельне ділення, залишок відкидається.

Знак% означає виділення залишку при целочисленном діленні. Ця операція вимагає, щоб з обох сторін від її знака стояли цілі величини. Якщо операнди арифметичної операції мають різні типи, то попередньо виконуються стандартні арифметичні перетворення і тип результату операції визначається загальним типом операндів після стандартних перетворень. Отже, вираз  $7/2$  буде мати значення 3 типу `int`, так як обидва операнда мають тип `int`, а вираз  $7.0/2$  дасть результат 3.5 типу `double`, оскільки цей тип має перший операнд. Приклади:

```
cout << 10/2 <<"\n"; // 5 (залишку немає)
cout << 10/3 <<"\n"; // 3 (залишок відкинутий)
cout << 300/165 <<"\n"; // 1 (залишок відкинутий)
cout << 10% 3 <<"\n"; // 1 (залишок)
cout << 20/4 <<"\n"; // 0 (залишку немає)
```

## Завдання

1. Ввести з клавіатури три числа, вивести їх середнє арифметичне.
2. Ввести з клавіатури два числа. Знайти і вивести на екран суму, різницю, добуток і частку цих чисел.
3. Ввести з клавіатури два цілих числа. Знайти і вивести на екран цілу частину і залишок від ділення цих чисел, а також точну частку.
4. Ввести з клавіатури два цілих числа  $m$  і  $n$ . Перевірити за допомогою операції %, чи ділиться без остачі  $m$  на  $n$ .
5. Ввести з клавіатури числа  $A$  і  $B$ . Знайти рівняння  $A * X = B$ .
6. Ввести з клавіатури  $A$ ,  $B$  і  $C$ . Знайти розв'язок рівняння  $A * X + B = C$ .
7. Ввести 3 числа, вивести на екран їх куби і квадрати.
8. Обчисліть і виведіть на екран значення  $y = 3x^2 + 10x^4$ . Значення  $x$  ввести з клавіатури.
9. Введіть 3 числа і знайдіть суму і різницю всіх можливих пар чисел.
10. Введіть 3 числа і знайдіть частку всіх можливих пар чисел.
11. Вирішіть систему рівнянь:

$$A_1x + B_1y = C_1$$

$$A_2x + B_2y = C_2.$$

Параметри  $A_1, A_2, B_1, B_2, C_1, C_2$  ввести з клавіатури.

## Методичні вказівки

- Введення і виведення виконувати за допомогою операцій `cin` і `cout`.
- Перед введенням даних з клавіатури на екрані повинен з'являтися текст запиту на введення.

## Практична робота 2. «Основні конструкції мови C++. Розгалуження і цикли»

**Мета роботи:** Вивчення логічних виразів, складених операторів мови C. Написання простих програм з умовними операторами та операторами циклу.

### Теоретичні відомості

У більшості більш-менш складних алгоритмів використовуються умовні конструкції (або розгалуження). Загальні відомості про відповідуючі умовних операторах в мові C++. в розд. 1.2. «Керуючі оператори C++» в [1].

### Керуючі логічні вирази

Коли в алгоритмі зустрічається точка розгалуження, відбуваються обчислення керуючого логічного виразу. Це стосується як конструкції `if-else` `if` (управляючий вираз) `оператори-1` `else` `оператори-2`

так і конструкції `switch`

```
switch (управляючий вираз) {  
  case константа-1: оператори; break;  
  case константа-2: оператори; break;  
  .....  
  default : оператори  
}
```

Залежно від обчисленого значення керуючого виразу вибирається та чи інша гілка для подальшого виконання програми. При цьому необхідно пам'ятати, що в мові C логічні вирази представляються за допомогою цілих чисел: логічному значенню `false` відповідає 0, логічному `true` – будь-яке число, відмінне від нуля.

В керуючому виразі використовуються *операції відношення*:

```
== дорівнює;  
!= не;  
< менше;  
> більше;  
<= менше або дорівнює;  
>= більше або дорівнює
```

і *логічні операції*:

```
|| логічне АБО (диз'юнкція);  
&& логічне І (кон'юнкція);  
! логічне НІ (інверсія).
```

Операції відношення мають пріоритет нижче арифметичних операцій, так що вираз виду `k > n * i` обчислюється як `k > (n * i)`. Пріоритет логічного І вище, ніж



логічного АБО, але обидві логічні операції виконуються після операцій відношення і арифметичних (див. розд.1.1.3 у [1]). У сумнівних випадках краще розставляти дужки.

На основі трьох логічних операцій можна формувати складні вирази. Наприклад, якщо є три змінні `exp1`, `exp2` і `exp3`, то логічна конструкція

```
if (exp1 > exp2 && exp2 < exp3 )
```

істинна, якщо значення змінної `exp1` більше значення `exp2` і значення `exp2` менше значення `exp3`. Якщо при обчисленні логічного виразу слід виконати операції з нижчим пріоритетом, то ці операції потрібно помістити в круглі дужки.

### Логіка умови в операторах циклів

Вивчаючи цикли, потрібно добре розуміти концепцію істинних і хибних значень в мовах C / C ++, оскільки в циклах, так як і в операторах розгалуження, присутні керуючі логічні вирази, або іншими словами, умови циклу. У заголовку циклу з параметром

```
for (E1; E2; E3)
```

умовою циклу є `E2`, яке може бути як простим, так і складним виразом. У циклах `while` і `do-while` умова циклу слідує за словом `while`:

```
while (E)
```

Будь-який з циклів виконується доти, поки умова циклу має значення `true` (не нуль). Якщо ж потрібно перервати виконання циклу, коли умова істинна, застосовується ключове слово `break`.

### Завдання

1. Ввести з клавіатури цілі значення змінних `x`, `y`. Визначити результат для наступних логічних виразів:  
`x == 0 || y != 0`  
`x > 5 && y > 8`  
`x > 5 || y > 8`  
`!(x > 7 && y > 7)`
2. Ввести з клавіатури два числа. Визначити найбільше з них.
3. Ввести з клавіатури 4 числа і визначити, які додатні, а які від'ємні. Знайти максимальне число.
4. Ввести з клавіатури 3 числа. Знайти і вивести на екран суму, різницю, добуток і частку цих чисел.
5. Ввести з клавіатури 5 чисел. Вивести на екран від'ємні числа і знайти їх суму.
6. Ввести 6 чисел, знайти і вивести на екран максимальне і мінімальне число.

7. Ввести з клавіатури два числа. Якщо обидва числа додатні, визначити їх добуток; якщо одне з чисел додатне, а інше від'ємне, визначити їх суму; якщо обидва числа від'ємні, визначити їх частку.
8. Обчислити число перестановок з  $n$  елементів.
9. За допомогою циклу обчислити  $x^n$ , де  $n$  – ціле число ( $n > 0$ ).
10. Скласти програму, яка обчислює і виводить на екран  $n!$ , якщо  $n$  парне, і  $(n)!!$ , якщо  $n$  непарне.
11. Скласти програму, яка виводить на екран квадрати, куби, корені квадратні, корені кубічні всіх цілих чисел від 1 до  $n$  (ціле число  $n$  ввести з клавіатури).
12. Скласти програму для розв'язання лінійного рівняння  $ax + b = 0$ , в якій враховуються випадки:  $a=0, b \neq 0$  (вироджене рівняння);  $a \neq 0, b=0$ ;  $a=0, b=0$ .
13. Скласти програму для розв'язання квадратного рівняння  $ax^2 + bx + c = 0$ , в якій враховуються випадки від'ємного і рівного нулю дискримінанта.

## Методичні вказівки

- Перед написанням програми до завдань 8 – 13 скласти блок-схему алгоритму обчислень.
- Введення параметрів і змінних, необхідних для вирішення завдань, виконується з клавіатури.
- Перед введенням даних на екрані повинен з'являтися текст запиту на введення.

## Практична робота 3. «Функції в C/C++»

**Мета роботи:** Отримання навичок роботи з функціями в програмах на C/C++.

### Теоретичні відомості

Загальні відомості про функції в мові C см. в розд. 4.1.1. в посібнику [1].

### Завдання

1. Ввести з клавіатури довжини двох катетів прямокутного трикутника, вивести довжину гіпотенузи.
2. Скласти функцію для обчислення  $2^n/n!$  ( $n>0$ ).
3. Скласти функцію для визначення числа перестановок з  $n$  елементів.
4. Скласти функцію, яка обчислює  $x^n$ , де  $n$  – ціле число ( $n>0$ ).
5. Написати функцію для обчислення  $Y = \frac{1}{x^{5(e^x-1)}}$ .
6. Написати функцію для обчислення  $Y = (x + \lfloor x \rfloor) \sqrt{x \sin^2 \pi x}$ .
7. Написати функцію для обчислення  $Y = \arccos(\cos x^3)$ .
8. Скласти функцію, яка обчислює  $e^{x^2}$  у вигляді ряду.
9. Скласти функцію, яка обчислює і виводить на екран квадрати, куби, корені квадратні всіх цілих чисел від 1 до  $n$  ( $n$  – ціле число, яке вводиться з клавіатури).
10. Скласти функцію для знаходження суми, різниці, добутку і частки 2-х чисел.
11. Скласти функцію, яка повертає квадрат модуля  $x$ , якщо  $x < 0$  і квадратний корінь  $x$ , якщо  $x > 0$ , при  $x=0$  повертає 0.
12. Скласти функцію, яка повертає значення 0 – якщо число кратне 5, 1 – якщо число кратне 3, 2 – якщо число кратне 3 і 5, і 3 – якщо число не кратне 3 і 5.
13. Скласти функцію для обчислення середнього значення елемента масиву цілих чисел, кількості додатних, від'ємних і нульових елементів цього масиву.

### Методичні вказівки

- Функції, які використовуються у вирішенні завдань, визначати в заголовному файлі (з розширенням .h).
- Значення аргументу функції вводиться з клавіатури за запитом, що з'являється на екрані

## Практична робота 4. «Робота з рядками»

**Мета роботи:** Освоєння прийомів роботи з рядками в C/C++.

### Теоретичні відомості

Загальні відомості про роботу з рядками в мові див. у розд. 5.1.3 -5.1.6 посібнику [1].

### Створення заголовного файлу та файлу реалізації

Для створення зручних програмних додатків прототипи функцій відокремлюється від їх визначення. Оголошення функцій міститься в заголовному файлі .h, щоб воно було доступно будь-якому клієнтові. Визначення функцій містяться в файлі реалізації .cpp. При необхідності модифікувати функції, зміни торкаються тільки файлу реалізації.

Приклад додатка, що оперує з датами (день, місяць, рік), які представляються цілими числами. Додаток складається із заголовного файлу date.h, файлу реалізації date.cpp і керуючого файлу MyDate.cpp, що містить функцію main.

Заголовний файл date.h:

```
#ifndef DATE_H //Запобігання багаторазового
#define DATE_H //включення заголовного файлу
#include <iostream.h>

//Оголошення глобальних змінних та функцій
    int day, month, year;
    void set();
    void print();
#include "date.cpp" //Підключення файлу реалізації
#endif
```

Файл реалізації date.cpp:

```
void set() {
    cout<<"Input data\n";
    cin>>day>>month>>year;
};

void print() {
    cout <<day<<". "<<month<<". "<<year;
}
}
```

Керуючий файл MyDate.cpp:

```
#include "date.h"
void main() {
    set();
    print();
}
```

## **Завдання**

1. Ввести з клавіатури рядок і вивести на екран довжину цього рядка.
2. Ввести з клавіатури латинську букву і перевірити велика буква або мала.
3. Ввести 2 рядки і вивести їх об'єднання.
4. Ввести з клавіатури і вивести на екран своє прізвище, ім'я та по батькові, а також дату народження.
5. Введіть з клавіатури рядок і символ та визначте, на якій позиції в рядку перший раз зустрічається цей символ.
6. Введіть з клавіатури рядок і символ та визначте, на якій позиції в рядку останній раз зустрічається цей символ.
7. Ввести з клавіатури два рядки і здійснити порівняння цих рядків.
8. Скласти функцію, яка отримує у вигляді параметра букву російського алфавіту і повертає відповідну їй велику літеру.
9. Скласти функцію, яка отримує у вигляді аргументу букву латинського алфавіту і повертає велику літеру російського алфавіту, схожу з написання (a-> А, b-> В, e-> Е, з-> С, u-> І і т. д.).
10. Virішити завдання 4–8 за допомогою створених в програмі функцій. Ввести з клавіатури рядок та вивести на екран довжину цього рядка.

## **Методичні вказівки**

- Користувальницькі функції, які використовуються у вирішенні завдань, визначати в заголовному файлі
- Про створення та підключення заголовного файлу див. розд. 9.1.4 посібнику [1].

## Практична робота 5. «Основні операції з матрицями в MATLAB»

**Мета роботи:** Освоєння основних операцій по створенню та перетворенню матриць, додаванню, множенню і діленню матриць в MATLAB.

### Теоретичні відомості

#### Пакет MATLAB для математичних обчислень

MATLAB – це високопродуктивне середовище для інженерних та математичних розрахунків, яке включає мову програмування, засоби візуалізації даних, а також засоби для створення графічного інтерфейса. Особливістю пакету MATLAB, назва якого означає «матрична лабораторія» (Matrix Laboratory), є те, що будь-які числа і змінні представляються у вигляді векторів, матриць або багатовимірних масивів.

#### Конструювання матриць в MATLAB

Основним режимом роботи в системі MATLAB є режим командного рядка, при якому команди набираються користувачем на клавіатурі у відповідь на запит системи, що має вигляд кутових дужок  $>>$ . На відміну від компілятора C++ система MATLAB являє собою інтерпретатор, тому команди, які надходять, оброблюються негайно.

Вираз у командному рядку може формуватися з викликів функцій, арифметичних і логічних операцій, операндами яких можуть бути числові, рядкові та логічні константи, масиви, змінні, виклики функцій, а також інші вирази.

Числовою структурою в MATLAB, яка найбільш часто використовується, є двовимірна прямокутна матриця розміром  $m \times n$ , в якій зручно зберігати різні числові данні. Для зберігання послідовності чисел (одновимірного масиву) використовується вектор-рядок  $1 \times n$  або вектор-стовпець  $n \times 1$ . Одне число в MATLAB вважається матрицею розміру  $1 \times 1$ . Матриці конструюються за допомогою оператора  $[]$ , наприклад, вектор-рядок  $A$  можна ініціалізувати одним з двох можливих способів:

$$A = [1 \ 2 \ 3 \ 4] \text{ або } A = [1, 2, 3, 4]$$

Прямокутна матриця представляється у вигляді перерахування рядків, які розділяються крапкою з комою:

$$A = [1 \ 2 \ 3 \ 4; \ 6 \ 7 \ 8 \ 9] \text{ або } A = [1, 2, 3, 4; 6, 7, 8, 9]$$

Нумерація рядків і стовпців в матрицях починається з 1 (на відміну від мови C/C++, в якій перший елемент одновимірного масиву або перші елементи рядків двовимірного масиву мають індекси 0). Щоб отримати доступ до елемента матриці  $B$ , який стоїть у  $i$ -ому рядку,  $j$ -му стовпці, потрібно записати  $B(i, j)$

(у C/C++ доступ до елемента двовимірного масиву виконується записом  $V[i][j]$ ).

Для доступу до рядків або стовпців матриці використовується операція двокрапка «:». Наприклад,  $A(:, k)$  – це  $k$ -й стовпець,  $A(k, :)$  –  $k$ -й рядок матриці  $A$ . Результатом операції  $A(:)$  є довгий стовпець, складений із стовпців матриці  $A$ . Відповідно  $A(k)$  – це  $k$ -й елемент вектора  $A$ . Якщо потрібно отримати доступ до частини рядків або стовпців матриці, використовується запис, наприклад, виду  $A(1:m, k)$  – це елементи в рядках від 1 до  $m$  стовпця  $k$ .

Операція двокрапки, записана у вигляді «first:last» генерує  $1 \times n$  матрицю (вектор) з послідовності чисел, інкрементованих на 1 від *first* до *last*. Якщо необхідно змінити крок збільшення чисел послідовності, використовується запис *step:last*, наприклад,

$A = 10:5:100$

$B = 0:0.02:1$

$C = 9:-1:0$

Матриці можуть брати участь в операції об'єднання (конкатенації), наприклад,  $C=[A \ B]$  – горизонтальне об'єднання матриць  $A$  і  $B$ ,  $C=[A;B]$  – вертикальне об'єднання матриць  $A$  і  $B$ .

У MATLAB передбачено ряд функцій, які конструюють часто використовувані матриці:

$\text{ones}(n)$  – квадратна матриця  $n \times n$  з одиниць;

$\text{ones}(m, n)$  – прямокутна матриця  $m \times n$  з одиниць;

$\text{zeros}(n)$  – квадратна матриця  $n \times n$  з нулів;

$\text{zeros}(m, n)$  – прямокутна матриця  $m \times n$  з одиниць;

$\text{eye}(n)$  – одинична матриця розміру  $n$ ;

$\text{diag}(A)$  – діагональна матриця, утворена з елементів вектора  $A$ ;

$\text{rand}(m, n)$  – матриця  $m \times n$  з однорідно розподілених випадкових чисел від 0 до 1;

$\text{randn}(m, n)$  – матриця  $m \times n$  з нормально розподілених випадкових чисел від 0 до 1.

## Арифметичні операції в MATLAB

Арифметичні операції додавання «+», віднімання «-» і множення «\*» застосовуються до змінних у MATLAB за правилами матричної алгебри, тобто розміри матриць повинні бути узгоджені. У MATLAB існує два види поділу матриць, праве «/» і ліве «\»:

$A/B$  має результатом  $AB^{-1}$

$A \setminus B$  має результатом  $A^{-1}B$ .

Предбачені також операції піднесення до ступеня «^» і транспонування матриць «'».

Можливо поелементне виконання операцій множення, ділення і піднесення до ступеня, для цього перед знаком операції ставиться крапка, наприклад,  $A.*B$ .

## Завдання

1. Створіть одиничну матрицю, одиничний вектор.
2. Створіть прямокутну матрицю з одиниць і прямокутну матрицю з нулів.
3. Створіть матрицю з рівномірно розподілених випадкових чисел від 0 до 1. Видаліть рядок. Видаліть стовпець.
4. Створіть квадратну і прямокутну матриці з нормально розподілених випадкових чисел від 0 до 1. Запишіть перший рядок і останній стовпець прямокутної матриці в вектори.
5. Створіть квадратну матрицю з рівномірно розподілених випадкових чисел від 0 до 9. Транспонується матрицю. Знайдіть детермінант вихідної і транспонованої матриць. Переконайтеся, що їх значення співпадають.
6. Створіть магічну матрицю. Виведіть на екран який-небудь елемент, стовпець і рядок матриці. За допомогою функції `sum` перевірте основну властивість магічної матриці для будь-яких стовпчика і рядка.
7. Вираз `M(k)` з одним індексом дає доступ до елементів матриці, розгорнутим в один стовпець. Така матриця утворюється з вихідної, якщо поспіль виписати її стовпці. Переконайтеся в цьому, вивівши на екран елемент  $M(m \times n - 1)$  довільної матриці  $m \times n$ .
8. Задайте дві квадратні матриці  $A$ ,  $B$ . Обчисліть  $C=A*B$  і  $D=B*A$ . Переконайтеся в тому, що результати не збігаються.
9. Задайте дві матриці однакового розміру  $A$  і  $B$ . Помножте одну матрицю на іншу поелементно  $A.*B$ . Розділіть матриці поелементно праворуч ( $A./B$ ) і зліва ( $A.\B$ ). Розділіть матриці поелементно в іншому порядку:  $B./A$  і  $B.\A$ . Порівняйте результати.
10. Задайте невироджену квадратну матрицю  $A$  і вектор  $X$ . Обчисліть  $F=A*X$ . Потім знайдіть результат лівого ділення матриць:  $X1=A\F$ . Переконайтеся, що  $X1$  збігається з  $X$ .
11. Задайте не одиничну матрицю (наприклад,  $M=[2 \ 2, 0 \ +1]$ ) Здійсніть поелементне піднесення до ступеня ( $M.^2$ ).
12. Для матриці випадкових чисел  $M$  обчисліть  $\cos(M)$ ,  $\exp(M)$ ,  $\sqrt{M}$ ,  $\log(M)$ .

## Методичні вказівки

Кожне завдання виконайте по можливості за допомогою двох різних способів



## Практична робота 6. «Функції у MATLAB»

**Мета роботи:** Освоєння прийомів роботи з функціями у MATLAB.

### Теоретичні відомості

#### Елементарні математичні функції MATLAB

MATLAB містить безліч функцій, призначених для роботи з матрицями і для створення нових матриць. Загальна форма використання функції в MATLAB така:

<список вихідних аргументів> = <ім'я функції> (<список вхідних аргументів або їх значення>).

Наприклад, виклик функцій  $\max(A)$  і  $\min(A)$ , які повертають максимальний і відповідно мінімальний елементи вектора  $A$ , виглядає як

$A_{\max} = \max(A)$

$A_{\min} = \min(A)$

Список вихідних аргументів (якщо їх два або більше) записується у квадратні дужки. Наприклад, наведені вище функції при наявності другого вихідного аргументу мають вигляд:

$[A_{\max}, i] = \max(A)$

$[A_{\min}, i] = \min(A)$

і повертають другому аргументу відповідно індекс елемента з максимальним або мінімальним значенням. Якщо максимальних (мінімальних) значень кілька, то повертається номер першого з них.

У MATLAB вбудовані ті ж елементарні математичні функції, що є в бібліотеці `math` мови C (див. розд. «1.2.4. Функції в C/C++»). В якості вхідних аргументів можуть виступати матриці. При цьому функція застосовується почленно до кожного елемента матриці, а значенням, що повертається, також є матриця.

#### Тригонометричні функції:

$\sin(z)$  – синус числа  $z$ ;

$\arcsin(z)$  – головне значення арксинуса (в радіанах);

$\cos(z)$  – косинус;

$\arccos(z)$  – головне значення арккосинуса;

$\tan(z)$  – тангенс;

$\arctan(z)$  – головне значення арктангенса;

$\sec(z)$  – секанс;

$\operatorname{arcsec}(z)$  – арксеканс;

$\csc(z)$  – косеканс;

$\operatorname{arccsc}(z)$  – арккосеканс;

$\cot(z)$  – котангенс;

$\operatorname{arccot}(z)$  – арккотангенс;

#### Гіперболічні функції:

$\sinh(z)$  – гіперболічний синус ;  
 $\operatorname{arcsinh}(z)$  – зворотний гіперболічний синус ;  
 $\cosh(z)$  – гіперболічний косинус ;  
 $\operatorname{arccosh}(z)$  – зворотний гіперболічний косинус ;  
 $\tanh(z)$  – гіперболічний тангенс ;  
 $\operatorname{arctanh}(z)$  – зворотний гіперболічний тангенс ;  
 $\operatorname{sech}(z)$  – гіперболічний секанс ;  
 $\operatorname{arcsech}(z)$  – зворотний гіперболічний секанс ;  
 $\operatorname{csch}(z)$  – гіперболічний косеканс ;  
 $\operatorname{arccsch}(z)$  – зворотний гіперболічний косеканс ;  
 $\operatorname{coth}(z)$  – гіперболічний котангенс ;  
 $\operatorname{arccoth}(z)$  – зворотний гіперболічний котангенс .

### **Експоненціальні та інші функції:**

$\exp(z)$  – експонента числа  $z$ ;  
 $\ln(z)$  – натуральний логарифм ;  
 $\log_{10}(z)$  – десятковий логарифм ;  
 $\operatorname{sqrt}(z)$  – квадратний корінь з числа  $z$ ;  
 $\operatorname{surd}(x, n)$  – корінь  $n$ -го степеня з числа  $x$  ;  
 $\operatorname{abs}(z)$  – модуль числа  $z$  ;

### **Функції округлення:**

$\operatorname{fix}(z)$  – округлення до цілого у бік нуля ;  
 $\operatorname{floor}(z)$  – округлення до цілого у бік  $-\infty$ ;  
 $\operatorname{ceil}(z)$  – округлення до цілого у бік  $+\infty$ ;  
 $\operatorname{round}(z)$  – звичайне округлення до найближчого цілого;  
 $\operatorname{mod}(x, y)$  – модуль залишку від ділення  $x$  на  $y$  ;

Для обчислення модуля комплексного числа в MATLAB служить функція  $\operatorname{abs}(Z)$ . Часто використовуваної функцією є також  $\operatorname{sum}(A)$ , яка обчислює суму елементів вектора  $A$ .

## **Файли-сценарії і файли-функції в MATLAB**

Програми в MATLAB оформляються у вигляді окремих файлів з розширенням `.m` (m-файли). Ці файли можуть являти собою сценарії (Script-файли або файли-сценарії) і функції (файли-функції).

За допомогою Script-файлів оформляють як основну програму, що управляє від початку до кінця всім обчислювальним процесом, так і окремі фрагменти коду. Файли-сценарії – це найпростіший тип m-файлів, тому що вони не містять вхідних і вихідних аргументів. Файли-функції служать зазвичай для багаторазового використання свого коду. Головною ознакою файла-функції є наявність у файлі першого рядка виду

```
function <список-вихідних-аргументів> = <ім'я-функції>
(<список-вхідних-аргументів>)
```

У простому випадку функції однієї змінної заголовок має вигляд:

```
function y = myfunc (x),
```

де myfunc – ім'я функції (m-файлу).

## Функції інтерактивного обміну даними в MATLAB

Найпростіше висновок значення змінної на екран здійснити видаленням символу «;» після команди присвоєння. У MATLAB також наявна функція disp, яка здійснює виведення значень зазначеної змінної або тексту в командне вікно.

```
disp (<змінна або текст, заключений в апострофи>)
```

Однак за допомогою цієї команди неможливо виконати виведення декількох змінних і, зокрема, об'єднати текст з числовими значеннями змінних. Цю проблему можна вирішити, об'єднуючи декілька змінних x1, x2, ... xn в один рядок, тим самим створюючи єдиний об'єкт для виводу:

```
disp ([x1 x2 ... xn])
```

Іншим способом виведення декількох змінних і тексту є використання функції sprintf. Ця функція має такий же синтаксис, що й printf в мові C, наприклад, у записі

```
Y = sprintf ('<текст1> %g <текст2>', X)
```

змінна Y отримує значення текстового рядка, що складається з лексеми <текст1>, значення числової змінної X в форматі %g і лексеми <текст2>. Остання розташовується після значення X. Надалі цю функцію можна використовувати при виведенні в команді disp:

```
disp (Y)
```

Введення інформації з клавіатури в діалоговому режимі можна здійснити за допомогою функції input, звернення до якої має вигляд:

```
x = input ('<запит>')
```

При виконанні цієї команди на екрані, в командному вікні з'являється текст <запит> (наприклад, запит на введення будь-яких даних). Програма зупиняється з переходом в режим очікування закінчення введення інформації. Після введення з клавіатури і натиснення клавіші Enter введена інформація записується в змінну x, і виконується перехід до наступної команди програми.

Якщо потрібно вибрати один з варіантів майбутніх обчислювальних дій, то зручно використовувати функцію menu. Ця функція, яка створює поточне вікно меню користувача, має наступний формат звернення:

```
k = menu('Заголовок меню', 'Альтернатива_1', ...  
'Альтернатива_n')
```

Після появи віконця меню програма тимчасово призупиняється, і система очікує вибору однієї з кнопок меню з потрібною альтернативою. Для управління форматом виводу чисел в командному вікні служить команда `format` з параметрами. Для представлення чисел з фіксованою точкою використовується за умовчанням формат `short`, який відображає 5 значущих цифр, наприклад, 3.1416. Якщо потрібно більше значущих цифр, застосовується команда

```
format long
```

Після якої числа з фіксованою крапкою відображаються з 15 значущими цифрами, наприклад, 3.14159265358979. Команда

```
format long e
```

встановлює формат відображення 15 значущих цифр для чисел з плаваючою крапкою: 3.141592653589793e +000.

## Завдання

1. Відкрийте текстовий редактор. Складіть файл-сценарій для обчислення будь-якого алгебраїчного виразу. Коефіцієнти в цьому виразі введіть з клавіатури. Збережіть його в домашній каталог з ім'ям `my_file.m`. Запустіть його. Дайте команду `help my_file`.
2. У текстовому редакторі створіть програму обчислення площі кола, радіус якого вводиться з клавіатури. Збережіть цей файл. Викличте з командного рядка написану програму.
3. Складіть `m`-файл-функцію з двома вхідними аргументами і одним вихідним аргументом. Обчисліть функцію при введених з клавіатури  $a$ ,  $x$ :

$$y(x) = 1 - \frac{ax^2}{2}.$$

Задайте  $x$  у вигляді вектора, виведіть  $y(x)$  за допомогою функції `sprintf`.

4. Перевірте дію команди `input` в режимі командного рядка. В якості строкової змінної введіть який-небудь вираз, наприклад `sin(2*x)`,  $x$  при цьому має бути визначений. Обчисліть значення заданої функції за допомогою функції `eval`.
5. Складіть `m`-файл-функцію з кількома вхідними та вихідними аргументами. Обчисліть:

$$x_1(t) = \cos(2at), \quad x_2(t) = \sin(2bt).$$

де  $a$ ,  $b$  - скаляри,  $t$ ,  $x_1$ ,  $x_2$  - вектори. Організуйте введення-виведення за допомогою функцій інтерактивного обміну.

## Практична робота 7. «Алгоритми розгалуження і цикли в MATLAB»

**Мета роботи:** Освоєння алгоритмів розгалуження та циклів, їх реалізації в пакеті MATLAB.

### Теоретичні відомості

#### Умовні оператори в MATLAB

Умовні оператори MATLAB нагадують за формою аналогічні оператори мови C. Існує скорочена форма:

```
if (умова)
    оператори
end
```

і повна форми цього оператора

```
if (умова)
    оператори-1
else
    оператори-2
end
```

Допустима також конструкція з множинними умовами наступного вигляду:

```
if (умова-1) оператори-1
    elseif (умова-2)
        оператори-2
    elseif (умова-3)
        оператори-3
    . . .
else
    оператори
end
```

Завершується умовний оператор MATLAB ключовим словом end.

#### Оператор вибору в MATLAB

Оператор вибору (оператор перемикачання) має в MATLAB структуру, аналогічну структурі оператора switch в C/C++:

```
switch (управляючий вираз)
case константа-1
    оператори-1
case константа-2
    оператори-2
.....
otherwise
```

```
оператори-N  
end
```

Цей оператор здійснює розгалуження обчислень залежно від значення змінної або виразу, розташованого за словом `switch`. Якщо значення, отримане при обчисленні цього виразу, збігається з якою-небудь константою після слова `case`, то виконується група операторів, наступна за цією константою. Якщо ж значення виразу не збігається ні з однією з констант після слів `case`, то виконуються оператори, які слідують за словом `otherwise`. Конструкція оператора `switch` завершується словом `end`.

## Оператори циклу в MATLAB

У мові MATLAB є оператор циклу з передумовою та оператор циклу з параметром (або арифметичний оператор). Цикл з передумовою має синтаксис:

```
while (умова)  
    оператори  
end
```

Оператори в середині циклу виконуються лише у разі, якщо виконана умова після слова `while`. Серед цих операторів зазвичай є такий, який змінює значення однієї із змінних, зазначених в умові циклу, забезпечуючи тим самим вихід циклу.

Арифметичний оператор циклу має вигляд:

```
for параметр = поч:крок:кін  
    оператори  
end
```

де параметр – ім'я керуючого параметру циклу («лічильника» циклу); поч – початкове значення цього параметра; крок – значення кроку, з яким він повинен змінюватися; кінець – кінцеве значення параметра. Якщо параметр крок не вказаний, його значення приймається за замовчуванням рівним одиниці. Для дострокового виходу з циклу застосовується оператор `break`.

## Постановка задачі

Відповідно свого варіанту (див. розд. 9.3) скласти програму в MATLAB для обчислення значень функції  $y(x)$  та суми ряду, який апроксимує дану функцію. Аргумент функції  $x$  змінюється від  $a$  до  $b$  із кроком  $(b-a)/(k-1)$ , де  $k$  – кількість точок. Наближене значення функції  $y(x)$  обчислюється за допомогою степеневих рядів (стовпчик «Сума» у таблиці розд. 6.3) для двох випадків:

- для заданого числа членів ряду  $n$ ;
- для заданої точності  $\epsilon$  (наприклад 0.0001).

Точне значення функції  $y$  (стовпчик «Функція»), яке також обчислюється програмою, служить для перевірки отриманих наближених значень.

## Варіанти

№	Функція	Діапазон зміни аргументу	n	Сума
1	$y = 3^x$	$0 \leq x \leq 1$	5	$S = 1 + \frac{\ln 3}{1!}x + \frac{\ln^2 3}{2!}x^2 + \dots + \frac{\ln^n 3}{n!}x^n + \dots, n \geq 0$
2	$y = \ln x$	$2 \leq x \leq 10$	10	$S = \frac{x-1}{x} + \frac{(x-1)^2}{2x^2} + \dots + \frac{(x-1)^n}{nx^n} + \dots, n \geq 1$
3	$y = \ln x$	$0.1 \leq x \leq 0.5$	5	$S = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} + \dots + (-1)^{n+1} \frac{(x-1)^n}{n} + \dots, n \geq 0$
4	$y = \frac{1}{2} \ln \left( \frac{x+1}{x} \right)$	$1 \leq x \leq 10$	5	$S = \frac{1}{2x+1} + \frac{1}{3(2x+1)^3} + \dots + \frac{1}{(2n-1)(2x+1)^{2n-1}} + \dots, n \geq 1$
5	$y = \frac{1}{2} \ln \left( \frac{1+x}{1-x} \right)$	$0 \leq x \leq 1$	5	$S = x + \frac{x^3}{3} + \dots + \frac{x^{2n+1}}{2n+1}, n \geq 1$
6	$y = \frac{1}{2} \ln \left( \frac{x+1}{x-1} \right)$	$2 < x \leq 10$	5	$S = \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots + \frac{1}{(2n+1)x^{2n+1}}, n \geq 0$
7	$y = \sin x$	$0 \leq x \leq 1$	5	$S = x - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots, n \geq 0$
8	$y = \cos x$	$0 \leq x \leq 1$	5	$S = 1 - \frac{x^2}{2!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots, n \geq 0$
9	$y = \frac{\pi}{4}$	$0 \leq x \leq \frac{\pi}{2}$	10	$S = \sin x + \frac{\sin 3x}{3} + \dots + \frac{\sin(2n+1)x}{(2n+1)} + \dots, n \geq 0$
10	$y = \frac{x}{2}$	$-1 \leq x \leq 1$	5	$S = \sin x - \frac{\sin 2x}{2} + \dots + (-1)^n \frac{\sin nx}{n} + \dots, n \geq 1$
11	$y = \sqrt{1-x}$	$0 \leq x \leq 1$	5	$S = 1 - \frac{1}{2}x - \frac{1}{8}x^2 - \dots - \frac{(2n-3)!!}{(2n)!!}x^n + \dots, n \geq 1$
12	$y = \sqrt{1+x}$	$0 \leq x \leq 1$	5	$S = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \dots + (-1)^{n+1} \frac{(2n-3)!!}{(2n)!!}x^n + \dots, n \geq 1$
13	$y = \frac{1}{1-x}$	$0 \leq x \leq 0.5$	5	$S = 1 + x + x^2 + \dots + x^n + \dots, n \geq 0$

14	$y = \frac{1}{1+x}$	$0 \leq x \leq 0.5$	5	$S = 1 - x + x^2 - x^3 + \dots + (-1)^n x^n + \dots, n \geq 0$
15	$y = \frac{1}{(1+x)^2}$	$0 \leq x < 1$	10	$S = 1 - 2x + 3x^2 - 4x^3 + \dots + (-1)^n (n+1)x^n + \dots, n \geq 0$
16	$y = \frac{1}{(1+x)^3}$	$0 \leq x < 1$	10	$S = 1 - 3x + 6x^2 + \dots + (-1)^n \frac{(n+1)(n+2)}{2} x^n + \dots, n \geq 0$
17	$y = \frac{1}{\sqrt{1+x}}$	$0 \leq x \leq 1$	5	$S = 1 - \frac{1}{2}x + \frac{3}{8}x^2 + \dots + (-1)^n \frac{(2n-1)!!}{(2n)!!} x^n + \dots, n \geq 1$
18	$y = \frac{1}{\sqrt{(1+x)^3}}$	$0 \leq x \leq 1$	5	$S = 1 - \frac{3}{2}x + \frac{15}{8}x^2 + \dots + (-1)^n \frac{(2n+1)!!}{(2n)!!} x^n + \dots, n \geq 0$
19	$y = e^{-2x}$	$0 \leq x \leq 1$	5	$S = 1 - 2x + \frac{(2x)^2}{2} + \dots + (-1)^n \frac{(2x)^n}{n!} + \dots, n \geq 0$
20	$y = \arcsin x$	$0 \leq x \leq 1$	5	$S = x + \frac{1}{3}x^3 + \frac{3}{40}x^5 + \dots + \frac{(2n-1)!!}{(2n+1)(2n)!!} x^{2n+1} + \dots, n \geq 0$
21	$y = \arctg x$	$0 \leq x \leq 1$	10	$S = x - \frac{x^3}{3} + \frac{x^5}{5} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots, n \geq 0$
22	$y = \arctg x$	$2 \leq x \leq 5$	5	$S = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots + (-1)^{n+1} \frac{1}{(2n+1)x^{2n+1}}, n \geq 0$
23	$y = \text{arcctg} x$	$0 \leq x \leq 1$	10	$S = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{2n+1} + \dots, n \geq 0$
24	$y = \frac{e^x + e^{-x}}{2}$	$0 \leq x \leq 1$	5	$S = 1 + \frac{x^2}{2!} + \dots + \frac{x^{2n}}{(2n)!} + \dots, n \geq 0$
25	$y = \frac{e^x - e^{-x}}{2}$	$0 \leq x \leq 1$	5	$S = x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots, n \geq 0$

## Методичні вказівки

1. Виконання роботи зводиться до обчислення при фіксованому аргументі  $x$  наступних величин: точного значення функції  $y(x)$ ; суми ряду  $S_n$  при фіксованому  $n$ ; суми ряду  $S_\varepsilon$ , що апроксимує  $y(x)$  з заданою точністю  $\varepsilon$ .
2. Алгоритм обчислень функції та ряду може бути наступним. Спочатку оголошуються змінні:  $x$  (значення аргументу),  $n$  (число членів ряду, що враховуються),  $\varepsilon$  (величина членів, що відкидаються), інтервали зміни аргументу:  $a$  і  $b$ . За запитом користувачем вводяться значення  $n$ ,  $\varepsilon$ ,  $a$ ,  $b$ . Корисно ввести у програму функції (наприклад, `sum1` і `sum2`), які



розраховують ряд при фіксованому  $n$  і при заданій точності  $\varepsilon$  відповідно. В цих функціях може бути корисною функція (наприклад, `term`) для обчислення членів ряду, яка також визначається користувачем. Для розрахунку значень функції  $y$  та ряду при різних значеннях  $x$  запускається цикл `for` із параметром. При кожному проході циклу для поточного  $x$  обчислюються  $y(x)$  і значення функцій `sum1` і `sum2`, котрі записуються в стовпчики `x`, `sn`, `se`, `y` відповідної таблиці:

<code>x</code>	<code>Sn</code>	<code>Se</code>	<code>y</code>
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....

Ця таблиця створюється за допомогою функції `fprintf` форматowanego виведення в MATLAB. Кількість рядків у таблиці має бути не менш 11.

3. Функції у програмі MATLAB мають бути оформлені в окремих `m`-файлах.

## Практична робота 8. «Двовимірна графіка в MATLAB»

**Мета роботи:** Освоєння прийомів побудови графіків функцій одного аргументу в графічному вікні MATLAB.

### Теоретичні відомості

Функції двовимірної графіки в MATLAB призначені для відображення залежностей типу  $y(x)$  в декартовій (прямокутній) системі координат. При цьому зазвичай будуються дві осі – горизонтальна  $X$  і вертикальна  $Y$ , і задаються координати  $x$  і  $y$ , що визначають вузлові точки функції  $y(x)$ . Ці точки з'єднуються один з одним відрізками прямих, тобто здійснюється лінійна інтерполяція для проміжних точок. Оскільки MATLAB – матрична система, сукупність точок  $y(x)$  задається векторами  $X$  і  $Y$  однакового розміру.

Функція `plot(X, Y)` будує графік функції  $y(x)$ , причому координати точок  $(x, y)$  якої беруться з векторів однакового розміру  $Y$  і  $X$ . Якщо  $X$  або  $Y$  – матриця, то будується сімейство графіків за даними, що містяться в стовпцях матриці.

Якщо другий аргумент у зазначеній функції відсутній, тобто маємо запис `plot(Y)`, то буде побудований графік  $y(i)$ , на якому значення  $y$  взяті з вектора  $Y$ , а  $i$  являє собою індекс відповідного елемента. Якщо  $Y$  містить комплексні значення, то виконується команда `plot(real(Y), imag(Y))`.

У команді `plot(X, Y, S)` перші два аргументи описують функцію  $y(x)$ , а третій аргумент, рядкова змінна  $S$ , задає тип лінії графіка. Звернення виглядає наступним чином:

```
plot(<x>, <y>, <'колір лінії, тип лінії, маркер точок'>);
```

Колір лінії	Тип лінії	Маркер точок
c – блакитний	- безперервна	. точка
m – фіолетовий	-- штрихова	+ плюс
y – жовтий	: пунктирна	* зірочка
r – червоний	-. штрих-пунктирна	o коло
g – зелений		x хрестик
b – синій		
w – білий		
k – чорний		

Якщо потрібно відобразити координатну сітку, задати масштаб осей, створити заголовок, підписи графіків і осей, то використовують команди, наведені в наступній таблиці:

<code>grid[on,off]</code>	Вмикає / вимикає сітку на графіку
<code>title('заголовок графіка')</code>	Створює напис заголовка графіка
<code>xlabel('підпис осі Ox')</code>	Створює підпис осі Ox
<code>ylabel('підпис осі Oy')</code>	Створює підпис осі Oy
<code>axis([xmin xmax ymin ymax])</code>	Масштабування осей ординат
<code>text(a,b,'текст')</code>	Створює текстовий напис, який починається в точці з координатами a,b

Можливо відображати кілька графіків в одному вікні. Це досягається за допомогою функції `subplot()`, що має наступний синтаксис:

```
subplot(<число рядків>, <число стовпців>, <номер координатної осі>)
```

Наприклад, для відображення двох графіків один під одним можна записати

```
figure(1);
subplot(2,1,1); % Графічне вікно розділяється на два рядки
                % і один стовпець, задається позиція «1» для першого
                % графіка
plot(x1,y1); % Відображення першого графіка
subplot(2,1,2); % Задається позиція «2» для другого графіка
plot(x2,y2); % Відображення другого графіка
```

Функція `fplot`, що має один з форматів запису

```
fplot (<ім'я функції>, limits)
```

будує графік функції (функцій) в інтервалі `limits=[xmin, xmax]`. Як ім'я функції може використовуватися М-файл або символічна запис функції, тобто рядок типу `'sin(x)'`, `'sin(x)/x'`, або `'[sin(x), fun1(x), fun2(x)]'`.

Можна наступним чином встановити розміри графіка по осі значень функції:

```
limits=[xmin xmax ymin ymax].
```

## Завдання

1. Побудуйте графіки функції  $\sin(x)$ ,  $\cos(x)$ ,  $x.^2$ . Для побудови графіка задайте вектор `x=xпоч:крок:xкін`, а потім використовувати команду побудови графіків `plot(x, sin(x))`. Зробіть висновок про оптимальний крок.
2. Змініть за допомогою позиції Edit колір графіка і нанесіть написи на графік, використовуючи позицію Insert. Збережіть графік у файл. Вивчіть інші можливості позицій Edit і Insert

3. Побудуйте параметричний графік `plot(sin(x), cos(x))`.
4. Фігури Ліссажу. Побудуйте графік в полярних координатах `t=0:pi/100:4*pi; plot(sin(w*t), cos(v*t))`. Спочатку візьміть співмірні значення частот  $w$  і  $v$  ( $w/v$  – раціональне число), наприклад  $w = 2, v = 3$ . Потім візьміть неспівмірні частоти, наприклад,  $w=2, v=pi$ . Переконайтеся, що незалежно від діапазону зміни аргумента  $t$  вийде незамкнута крива.
5. Побудувати в одному вікні графіки трьох функцій:  $q / (x.^2 + q.^2)$ , при різних значеннях  $q=1, q=0.5, q=0.1$ , причому колір одного з графіків повинен відрізнятися від кольору його вузлових точок.
6. Побудуйте функцію  $\sin(x)/x$  за допомогою функції `fplot`.
7. Побудуйте графік функції  $\exp(x)/x$  в логарифмічному масштабі. Вектор  $x$  задається наступним чином `x=logspace(-1,3)`. Нанесіть масштабну сітку за допомогою команди `grid on`.
8. Побудуйте графік експоненційної функції в логарифмічному масштабі по осі  $y$ . Вектор  $x$  задайте самостійно.
9. Підпишіть графік за допомогою команди `title`. Підпишіть осі. Додайте у поле графіка легенду.
10. Побудуйте графік функції  $0.5*x + 0.1*\sin(25*x)$ . Використовуючи команду `hold on` на цьому ж графіку побудуйте графік  $\text{abs}(x)$ .
11. Побудуйте в одному графічному вікні 4 графіка (2 по горизонталі, 2 по вертикалі):  $\sin(x)/x, \sin(5*x), \cos(2*x + 0.2), \cos.^2(x)$  в інтервалі  $-3*pi:3*pi$ .

## Методичні вказівки

- Щоб повернути масштаб по осях до штатних значень (прийнятим за замовчуванням), використовуйте команду `axis('auto')`.
- Якщо потрібно перемістити початок відліку в лівий верхній кут (матрична система координат, коли відлік по вертикальній осі відбувається з верхнього лівого кута), використовуйте команду `axis('ij')`.
- Щоб повернутися до декартової системи координат (початок відліку в нижньому лівому куті) використовується команда `axis('xy')`.
- Щоб встановити однаковий діапазон зміни змінних по осях  $X, Y$ , потрібно записати `axis('square')`.

## Практична робота 9. «Тривимірна графіка в MATLAB»

**Мета роботи:** Освоєння прийомів роботи з тривимірною графікою в MATLAB.

### Теоретичні відомості

В MATLAB тримерный график задается в виде поверхности, на которой расположены точки с координатами  $z$ , определенными в узлах сетки на плоскости  $xy$ . Соседние точки на поверхности соединяются отрезками прямых. Для построения поверхности в трехмерном пространстве служат функции `mesh` и `surf`: первая строит поверхность в виде цветных проволочек, соединяющих соседние точки, а вторая – поверхность в виде закрашенных сегментов.

Для отображения функции двух переменных  $z=f(x,y)$  нужно с помощью функции `meshgrid` создать квадратную сетку на плоскости  $xy$ , узлы которой будут задаваться векторами  $X$  и  $Y$  с повторяющимися значениями. Затем на созданной сетке определить вектор значений  $z=f(x,y)$  и применить одну из функций: `mesh`, `surf` или `plot3`. Например,

```
[X,Y] = meshgrid(-8:.5:8);  
R = sqrt(X.^2 + Y.^2);  
Z = sin(R) ./ (R+0.001);  
mesh(X,Y,Z)
```

По умолчанию для построения сеточной поверхности используется текущая карта цветов. Если нужно построить одноцветную поверхность, используется свойство `EdgeColor`. Так команда

```
mesh(X,Y,Z, 'EdgeColor', 'black')
```

построит поверхность из черных проволочек.

Функция `surf` применяется аналогично `mesh`. Цвета поверхности можно задать командой `colormap`, а отображение шкалы цветов – командой `colorbar`:

```
surf(X,Y,Z)  
colormap hsv  
colorbar
```

Команды типа `meshc()`, `surfc()` в дополнение в дополнение к тримерным поверхностям строят проекцию на плоскости  $xy$  линий постоянного уровня. Команды `meshz()`, `surfz()` дополняют поверхности плоскостью отсчета на нулевом уровне, закрывая пьедесталом поверхность ниже этого уровня.

Функция `plot3(x,y,z)`, где  $x, y, z$  – одномерные массивы одинакового размера, строит точки с координатами  $x(i), y(i), z(i)$  и соединяет их прямыми линиями.

### Завдання

1. Побудувати за допомогою функції `plot3()` тривимірну просторову спіраль:

```
x=2sint; y=cost;
```

де  $t, x, y$  – вектора однакового розміру. Встановити діапазон для  $t$ :  $t = -9 \cdot \pi : 0.05 : 9 \cdot \pi$ . Окреслити осі координат. Змінити зображення і колір графіка. Додати пояснення до зображення у вигляді легенди `legend('Тривимірна спіраль')`.

- Побудувати за допомогою функції `mesh()` просторову сітчасту фігуру:

$$z = A \exp(-x^2 - y^2)$$

Для змінних задати інтервал  $(-5 : 0.1 : 5)$ . Коефіцієнт  $A$  замінити на  $2, 10$ . Створити напис 'Z-поверхня'. Для порівняння застосувати `plot3` замість `mesh`.

- Побудувати за допомогою функції `mesh()` просторову фігуру:

$$R = \sqrt{x^2 + y^2} + 0.001, \quad z = \frac{\sin R}{R}$$

Для змінних задати інтервал  $(-15 : 0.2 : 15)$ . Побудувати проекції ліній постійного рівня.

- Те ж, що в завданні 3, але з п'єдесталом відліку на нульовому рівні.
- Побудувати за допомогою функції `surf()` просторову суцільну фігуру завдання 2. Створити заголовок, підписи осей. Задати різні кольорні теми.
- Побудувати за допомогою функцій `sphere()` і `plot3()` стандартну сферичну поверхню радіуса 5. Задати число перетинів, наприклад, 10 (за замовчуванням – 20). Замінити функцію `plot3()` на `mesh()`, `surf()`.
- Імовірнісний розподіл Гаусса (нормальний розподіл) має від  $f(x) = A \exp(-x^2)$ . Цей розподіл ілюструється в MATLAB функцією `peaks`, яка має вигляд: `peaks`, которая имеет вид:

$$\begin{aligned} z &= 3 * (1-x) . ^2 . * \exp(- (x . ^2) - (y+1) . ^2) ... \\ &- 10 * (x/5 - x . ^3 - y . ^5) . * \exp(-x . ^2 - y . ^2) ... \\ &- 1/3 * \exp(- (x+1) . ^2 - y . ^2) \end{aligned}$$

Побудувати поверхні `z=peaks` за допомогою `surf(z)`. Порівняти з поверхнею на основі функції `mesh(z)`.

- Задати для розподілу  $[X, Y, Z] = \text{peaks}$  кольорові масивні рівні за допомогою `pcolor(X, Y, Z)`. Вивести в графічне вікно шкалу кольорів (функція `colorbar`).
- Для функції `peaks` побудувати лінії рівня за допомогою функції `contour()`. Наприклад, `contour(peaks, 10), grid, title('Лінії рівня')`.
- Те ж, що в завданні 9, але лінії рівня повинні бути з кольорною заливкою (функція `contourf`).

## Практична робота 10. «Ітераційні процеси для розв'язання нелінійних рівнянь»

### Теоретичні відомості

#### Метод Ньютона розв'язання нелінійних рівнянь

Метод Ньютона, також відомий як метод дотичних, – це ітераційний чисельний метод знаходження кореня (нуля) заданої функції. Вихідне рівняння

$$f(x) = 0 \quad (1)$$

представляється у вигляді розкладання в околі кореня  $x_k$ :

$$f(x_k) + f'(x_k)(x - x_k) = 0 \quad (2)$$

де  $f'(x_k) = \left. \frac{df}{dx} \right|_{x=x_k}$ . Розв'язання рівняння (1) приймається за чергове наближення

$x_{k+1}$ :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (3)$$

Метод Ньютона має просту геометричну інтерпретацію: графік функції замінюється дотичною до нього в точці  $(x_k, f(x_k))$  і за чергове наближення  $x_{k+1}$  приймається абсциса точки перетину її з віссю  $OX$ . Використовуючи цю інтерпретацію легко отримати розрахункову формулу (3) методу Ньютона.

З формули (3) видно, що збіжність послідовності  $x_{k+1}$  до кореня залежить від властивостей функції  $f'(x_k)$ , і не завжди збіжність має місце. Тому важливо правильно вибрати початкове значення  $x$  для ітерацій.

#### Функції MATLAB для розв'язання нелінійних рівнянь

У MATLAB основною функцією для знаходження кореня нелінійного рівняння є функція `fzero`, яка може мати декілька вхідних аргументів:

`fzero(@fun, x0)` – повертає значення  $x$ , при якому  $\text{fun}(x) = 0$ ;  $x_0$  – початкове («стартове») значення аргументу  $x$ . Значення, яке повертається, близьке до кореня або дорівнює `NaN`, якщо корінь не знайдений (відсутній);

`fzero(@fun, [x1 x2])` – повертає значення  $x$ , при якому  $\text{fun}(x) = 0$ ; `[x1 x2]` – заданий інтервал пошуку;

`fzero(@fun, x.tol)` – повертає результат із заданою похибкою `tol`;

`fzero(@fun, x.tol.trace)` – видає на екран інформацію про кожну ітерацію.

Всі виклики функції `fzero()` знаходять корінь, якщо графік лівій частині рівняння  $\text{fun}(x) = 0$  перетинає вісь  $x$ . Якщо графік є дотичним осі, то точка

дотику також є коренем, але такий корінь функція `fzero` визначити не може. Крім того, `fzero` може обчислити тільки один корінь, для обчислення інших наявних коренів доводиться повторно використовувати цю функцію.

Всі корені нелінійного рівняння на заданому проміжку, а також точки дотику з віссю  $x$  можна знайти за допомогою іншої функції – `fsolve`. Ця функція з пакету `Optimization Toolbox` (у складі `MATLAB`) може використовуватися і для розв'язання систем нелінійних рівнянь виду  $f(x)=0$  методом найменших квадратів. Вона шукає не тільки точки перетину, а й точки дотику.

Для розв'язання систем нелінійних рівнянь можна також використовувати функцію `solve` з пакету `Symbolic Math Toolbox`. Ця функція оперує з символьними виразами та здатна видавати результат також в символьній формі, а якщо такого немає, то вона дає розв'язок в чисельному вигляді, наприклад,

```
>> solve('0.15*x-sin(x+0.1)')
ans =
-0.11764813657749357727344695568323
```

## Постановка задачі

Побудувати графік і знайти корені нелінійного рівняння відповідно свого варіанту:

- за допомогою ітерацій методом Ньютона (вивести на екран результати проміжних ітерацій і графік зміни похибки на кожній ітерації);
- використовуючи функцію `fzero`;
- використовуючи функцію `fsolve`.

## Варіанти

№	Уравнение $f(x)=0$	Интервал $[a;b]$
1	$x \arctg x - 1 = 0$	$[0.0; 2.0]$
2	$x^3 - x^2 + 6x - 3 = 0$	$[0.0; 1.0]$
3	$e^x - \frac{1}{x} - 3 = 0$	$[1.0; 2.0]$
4	$\arctg 2x - \frac{1}{1+x} = 0$	$[0.0; 1.0]$
5	$e^x - 10 \ln x - 8 = 0$	$[2.0; 3.0]$
6	$\sqrt{x} - \operatorname{tg}(x(0.5 - x)) - 1 = 0$	$[0.0; 1.0]$
7	$\sin \sqrt{x} - \cos \sqrt{x} + 2\sqrt{x} = 0$	$[0.0; 0.5]$



8	$x^4 + 2x^3 - x - 2 = 0$	[0.5; 1.5]
9	$x^3 - e^x - 3 = 0$	[2.0; 3.0]
10	$x^6 - 3x^2 + x = 0$	[0.2; 0.8]
11	$\sqrt[3]{5-x} - x = 0$	[1.0; 2.0]
12	$x^2 - 10\ln x = 0$	[0.0; 2.0]
13	$x^2 - \cos x = 0$	[0.0; 2.0]
14	$10\ln x - \arctg x = 0$	[0.0; 2.0]
15	$x^2 \arctg x - 1 = 0$	[0.0; 2.0]
16	$2x^2 + \ln x - 1 = 0$	[0.0; 1.0]
17	$x - \arctg \sqrt[3]{x} - 1 = 0$	[1.0; 2.0]
18	$x^2 - 0.2e^x = 0$	[0.5; 1.0]
19	$x^2 - \ln(x+1) = 0$	[0.5; 2.0]
20	$x^5 - x - 2 = 0$	[1.0; 1.5]
21	$x - 0.5 - \sqrt[4]{x} = 0$	[1.0; 2.0]
22	$x - \tg x = 0$	[1.0; 2.0]
23	$x^4 - e^x = 0$	[1.0; 2.0]
24	$2x^2 - \ln x - \tg x = 0$	[0.0; 1.5]
25	$\exp(x^2 - x^3) - \sin x = 0$	[0.5; 1.5]

### Методичні вказівки

Перед розв'язанням нелінійного рівняння рекомендується побудувати графік функції  $f(x)$  для наближеного визначення коренів і інтервалів, у межах яких вони знаходяться.

## Література

1. Обчислювальна техніка та алгоритмічні мови. Частина 1. Мова програмування C++. Методичні вказівки до виконання лабораторних робіт /укл. О.Ю. Гаєвський – К.: ФЕА НТУУ «КПІ», 2015. – 48 с.
2. Шилдт Г.. C++. Базовый курс. – М: Вильямс.–2014, 624 с.
3. Грицюк Ю., Рак Т. Програмування мовою C++. – Львів: Вид-во ЛДУ БЖД, 2011. – 292 с.
4. Лазарев Ю.Ф. Начала программирования в среде MatLAB: Учебное пособие. - К.: НТУУ "КПИ", 2003. – 424 с.
5. Поршнеv С. В. MATLAB 7. Основы работы и программирования. Учебник — М.: ООО «Бином-Пресс», 2011. — 320 с.: