

Лабораторна робота №1

Основи програмування в інтегрованій системі MatLab. Розв'язання функціональних рівнянь з однією змінною

Мета роботи: ознайомитися з програмуванням в інтегрованій системі автоматизації математичних та науково-технічних розрахунків MatLab. Розглянути правила побудови функцій на прикладі одного з методів розв'язання функціональних рівнянь з однією змінною, проаналізувати отримані помилки, сформулювати робочу функцію – *m*-файл.

1.1. Короткі теоретичні відомості по програмуванню в інтегрованій системі автоматизації математичних розрахунків MatLab.

Інтегрована система автоматизації математичних та науково-технічних розрахунків MatLab є добре апробованою надійною системою розв'язання математичних задач з поданням даних у вигляді матриць. Вона розроблена фірмою MathWorks Inc.

При створенні багатьох автоматизованих систем математичних обчислень розробники прагнули уникнути необхідності написання користувачем власних програмних кодів. Але такий підхід значно обмежував можливості користувача при розв'язанні задач, що використовують нові методи та алгоритми.

Розробники MatLab провели величезну роботу по створенню операторів і функцій для розв'язання багатьох практичних задач. До таких функцій належать як прості (транспонування матриць, обчислення косинуса та ін.) так і складні функції, що реалізують певні методи розв'язання задач. Число таких функцій сягає багатьох тисяч і невинно збільшується.

А для вирішення задач з використанням алгоритмів користувача, MatLab має потужну ***математично-орієнтовану мову програмування високого рівня***. Тому інтегрована система автоматизації математичних MatLab з успіхом застосовується в багатьох країнах Європи для розв'язання нових, нестандартних та найбільш складних математичних задач.

Мова програмування в системі MatLab нагадує Паскаль. Запис коду програми відбувається у редакторі системи, що викликається з рядка головного меню MatLab командами **File → New → M-file**, але попередньо може бути написаний у будь-якому текстовому редакторі. Файл коду програми називається *m*-файлом і має розширення *.m*, тобто можемо записати: *filename.m*. Система MatLab має власний редактор і відладчик. Формули створюваних методів записуються у формі, притаманній мовам програмування, але ця незручність компенсується високою швидкістю обчислень, яка наприклад на порядок вища ніж у системи MathCad.

Мова програмування системи MatLab використовує майже всі відомі засоби програмування, в тому числі об'єктно-орієнтоване і візуальне програмування. У

програмному коді можна використовувати також внутрішні функції MatLab. Це забезпечує великі можливості при створенні програм як початківцям так і досвідченим програмістам.

В системі MatLab реалізовані два підходи до розв'язання задач:

- ✓ з використанням внутрішніх функцій, що реалізовані розробниками MatLab,
- ✓ за допомогою написаних користувачем власних програм, код яких формується у вигляді *m*-файлів.

Щоб скористатися першим способом розв'язання задач, достатньо задати значення змінної, набрати потрібне ім'я внутрішньої функції з обов'язковими параметрами у рядку запрошення вікна «Command Window» і натиснути «Enter» (рис.1). Якщо у ході розв'язання задачі використовуються кілька методів, то слід викликати їх послідовно, вказуючи необхідні параметри та розділяючи їх крапкою з комою (;). Кожен сеанс роботи в MatLab (у середовищі прийнято іменувати їх сесіями) відображається у вікні «Command Window», він вміщує змінні, значення яких задавалися під час сесії, функції, що викликалися, і всі наслідки «проб і помилок» користувача. Як правило при збереженні сеансу користувача цікавлять лише значення змінних та результати обчислень, їх запам'ятовують у вигляді

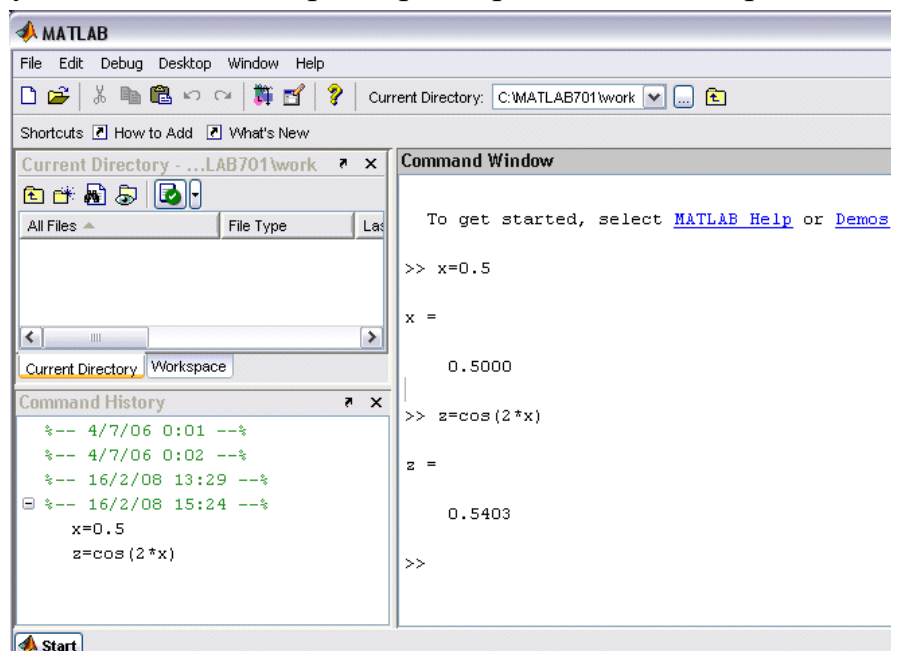


Рис. 1. Вікно системи MatLab при виконанні елементарних операцій.

mat-файлів з використанням функції *save filename*. Якщо потрібно записати значення лише одної або кількох змінних, то їх імена вказують наприкінці команди запам'ятовування, наприклад: *save filename X ; save filename X Y Z*. Якщо ж потрібно зберегти весь сеанс з відображенням всіх функцій, повідомлень про помилки та ін., то користуються функцією *diary filename*. Всі записані таким чином дані відкриваються у вікні «Command History» (рис.1).

Для реалізації власних алгоритмів і методів розв'язання задач скористайтесь редактором і відладчиком, які дозволяють отримати власну функцію для розв'язання задачі у вигляді *m*-файлу. При написанні власного коду і роботі над помилками користувач має викликати її у вікні «Command Window» із вказуванням необхідних параметрів, *при цьому у рядку «Current Directory» (рис.1) потрібно вказати повний шлях до папки, в якій знаходиться m-файл*. Якщо створений *m*-файл має помилки, то їх зміст і номер рядка з помилкою виводяться у поточному рядку командного вікна.

Серед *m*-файлів розрізняють **файли коду** та **файли-сценарії**. У файлах коду міститься програмний код, який може складатися з однієї чи кількох функцій. У файлах-сценаріях послідовно записують значення змінних, імена внутрішніх функцій та *m*-файлів, які використовуються для розрахунків.

Файли коду. У *m*-файлі, що містить код, ім'я вихідної функції повинно співпадати з іменем *m*-файлу (рис. 2). Внутрішня мова програмування MatLab дуже схожа з мовою програмування Pascal, в ній використовуються ті ж мовні конструкції, відмінності мають лише при зверненні до елементів матриць. Наприклад, якщо потрібно послідовно звертатися до елементів рядка чи стовпчика матриці в MatLab, то на відповідній позиції замість змінної пишуть двокрапку (:), тобто $M(i, :)$. При його створенні та перевірці редактор забезпечує синтаксичний контроль. Редактор має деякі відладочні можливості – він дозволяє встановити в тексті файлу контрольні точки, в яких призупиняється виконання програми і користувач

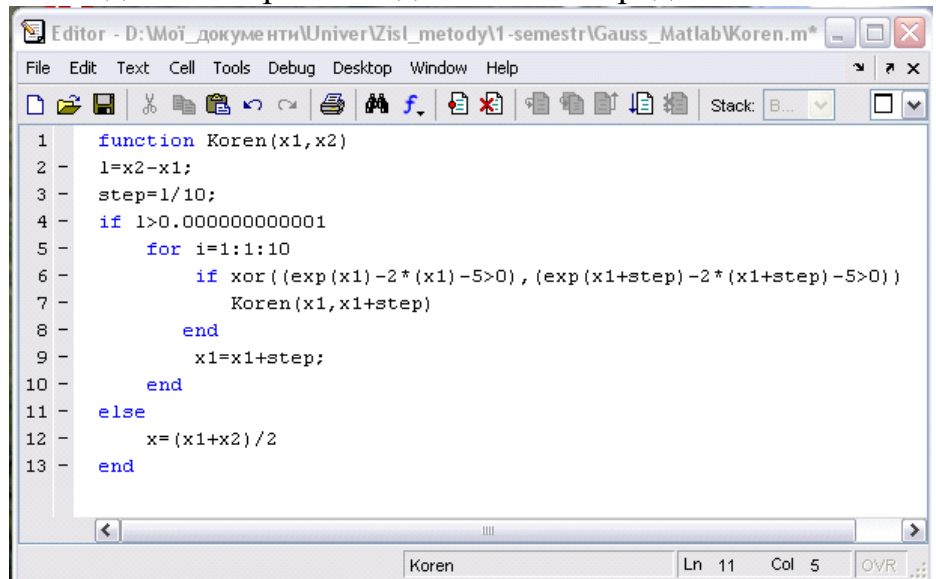


Рис.2. Приклад файлу-функції у редакторі MatLab.

може оцінити проміжні результати обчислень. Також є можливість покрокового виконання файлу, що особливо важливо при пошуку помилок. Редактор дозволяє зберегти файл в текстовому форматі з розширенням **.m**

Структура **m-файлу-функції** може бути представлена так:

```
function var=f_name(список_параметрів)
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами
var=вираз
```

Файли, записані з розширенням **.m**, можуть бути **файлами-сценаріями**. Такі файли можуть створюватися для задавання функції, введення даних у вигляді векторів чи матриць, побудови графіків із заданими параметрами і т.і. Всі записані в файлі-сценарії дані можливо вводити і в «Command Window», але форма файлу-сценарію більш компактна і його можна зберегти у вигляді файлу. При виклику такого файлу у «Command Window» за його ім'ям всі записані в файлі дані виводяться на екран, після чого можемо викликати файл коду, який їх використовує, реалізуючи певний метод обчислень.

Файл-сценарій (Script-файл) має наступну структуру:

%Основний коментар

%Додатковий коментар

Тіло файлу з будь-якими виразами.

Наприклад: визначимо функцію: $f(x) = 1 + \frac{x^3}{12}$. Файл сценарій запишемо у вигляді:

```
function y = stepfunc(x)
```

```
y = 1 + x^3/12;
```

При виклику функції у «Command Window» відобразяться наступні записи:

```
» stepfunc(3)
```

```
ans =
```

```
3.2500
```

Іноді корисним є розрахунок значень функції за допомогою команди `feval`. У її параметрах функція повинна бути представлена, як рядок:

```
» feval ('stepfunc', 3)
```

```
ans =
```

```
3.2500
```

Матриці в середовищі MatLab. Всі дані, що вводяться в середовищі MatLab інтерпретуються, як матриці. При введенні матриць розділовим знаком між елементами рядка є пробіл, а між самими рядками – крапка з комою. Наприклад:

```
» M = [1 2 3; 4 5 6; 7 8 9]
```

```
M =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

Для вказування певного елемента матриці, користуються індексами:

```
» M (2;2)
```

```
ans =
```

```
5
```

Можемо також задавати матриці у вигляді арифметичних виразів:

```
» C = [1+3^2 cos(0.5) sqrt(9)]
```

```
C =
```

```
10 0.8776 3
```

Задавати матрицю можна як змінну діапазону:

```
» U = 0:0.5:3
```

```
U =
```

```
0 0.5000 1.0000 1.5000 2.0000 2.5000 3.0000
```

При використанні її в якості аргументу отримаємо матрицю того ж розміру:

```
» sin(U)
```

```
ans =
```

```
0 0.4794 0.8415 0.9975 0.9093 0.5885 0.1411
```

В середовищі MatLab є функції для створення матриць з нульовими елементами (їх часто використовують, як матрицю для запису результатів обчислень) і одиничних матриць, аргументами цих функцій є їх розмір, що задається у вигляді цілих чисел.

```
» W = zeros(4,4);
```

 створення нульової матриці розміром 4×4

```
» H = ones(4,4);
```

 створення матриці з одиничними елементами
розміром 4×4

Операції над матрицями. З матрицями в середовищі MatLab можемо проводити наступні операції:

+	додавання
-	віднімання
*	добуток (множення)
^	піднесення до степеня
'	транспонування і спряження

При розв'язанні поставленого в лабораторній роботі завдання студентіві потрібно більш детально ознайомитися з особливостями програмування на внутрішній мові інтегрованої системи MatLab. Для цього у *додатку В* наведено відомості про особливості програмування в MatLab.

1.2. Завдання для виконання лабораторної роботи:

1) Створити програму на внутрішній мові середовища MatLAB та на іншій високорівневій мові програмування, що реалізує метод за варіантом (таблиця 1.1), провести тестування створеної програми на прикладі, проаналізувати отримані результати. Точність знаходження розв'язку $\epsilon = 0,01$. Методи відокремлення та уточнення коренів представлені у лекції 2.

Варіанти завдань до лабораторної роботи №1.

Таблиця 1.1

Варіант	Метод	Тестовий приклад*
1	метод хорд	$\cos(\sin(x^3)) - 0.7$ $[-3, 3]$
2	метод січних	$x^2 - 2x - 4$ $[-4, 4]$
3	метод простої ітерації	$\frac{\sin(x^3)}{5}$ $[-5, 5]$
4	метод дотичних (Ньютона)	$\frac{x^4 - 1}{x^3}$ $[-10, 10]$
5	метод половинного ділення	$x^2 \cos(x) - 0.2$ $[-2, 2]$
6	метод дотичних (Ньютона)	$1.5 - x^{1-\cos(x)}$ $[-10, 10]$
7	метод половинного ділення	$x^3 - 3^x + 10$ $[-5, 5]$
8	метод хорд	$\cos(x^2 - x + 1)$ $[-5, 5]$
9	метод січних	$e^x - 2x - 5$ $[-10, 10]$
10	метод дотичних (Ньютона)	$\frac{x^3 - 7x + 1}{3}$ $[-6, 6]$
11	метод хорд	$\cos(x^2) - 0.5$ $[-4, 4]$
12	метод простої ітерації	$\sin(x) - \lg(x)$ $(0, 10]$

Примітка: *Створені програми перевірити на прикладі рівнянь (за варіантом), що вводяться користувачем з клавіатури.

2) Оформити звіт з лабораторної роботи у відповідному бланку і переслати його викладачу до вказаного терміну на виконання лабораторної роботи.

3) Для підрахунку похідної формула чисельного диференціювання для першої похідної:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

де Δx – будь-яке мале припущення, наприклад 0,05.

Додаток А

Приклад заповнення бланку звіту

Лабораторна робота № 1.

Тема:	Розв'язок рівнянь з однією змінною
Завдання	Створити програму на внутрішній мові середовища MatLAB (чи GNU Octave) та на мові програмування ..., що реалізує метод хорд* (варіант 1), провести тестування створеної програми на прикладі рівняння $\cos(\sin(x^3)) - 0.7 = 0$ на проміжку $[-3, 3]$. Точність знаходження розв'язку $\varepsilon = 0,01$.
Короткий хід виконання	<Коротко описується порядок підготовки вхідних даних для першого і другого етапу розв'язання поставленого завдання, використані формули з короткими поясненнями. Вказуються особливості алгоритму реалізації задачі мовою програмування. >
Лістинг реалізації методу	<Наводиться лістинг реалізації задачі у середовищі MATLAB (чи GNU Octave) та на мові ..., в якому є основні коментарі по етапам виконання поставленого завдання (реалізований метод, а не інтерфейс).>
Результати	<Представляється результат (у числовому та графічному вигляді) реалізації задачі у MATLAB (чи GNU Octave) та на мові ..., на прикладі, наведеному у варіанті.>
Висновок	<Проводиться аналіз та порівняння результатів, отриманих при реалізації методу на різних мовах програмування, оцінка достовірності та точності результату.>
Виконав	Студент групи КС-171(1) Гончаренко А.В.

Додаток В

Інтегрована система автоматизації математичних та науково-технічних розрахунків MATLAB

MATLAB (MATrix LABoratory – матрична лабораторія) – одна з найдавніших, ретельно пророблених і апробованих систем автоматизації математичних розрахунків. Вона побудована на розширеному поданні матричних операцій. Але синтаксис мови програмування системи продуманий настільки ретельно, що матричне спрямування MATLAB не відчувається користувачами, яких не цікавлять матричні обчислення.

Система MATLAB з точки зору її розробників пропонується, як мова програмування високого рівня для математичних та науково-технічних обчислень. Вона відобразила в собі не тільки весь досвід розвитку та комп'ютерної реалізації чисельних методів за останні три десятиліття, а і досвід розвитку математики за всю історію Людства. Системою користуються більш ніж 500 000 тисяч легально зареєстрованих користувачів.

Система MATLAB має на сьогоднішній день 12 реалізацій, 12-а реалізація відома як версія 6.0. Розробник системи фірма – MathWorks. Inc – піклуючись про рейтинг системи доповнила її пакетами програм, що призначені для розв'язання складних сучасних технічних задач (таблиця В1).

Перелік основних програмних продуктів класу MATLAB.

Таблиця В1

№	Назва пакету програм	Призначення.
1	2	3
1	MATLAB for Windows	Ядро системи, основні бібліотеки для розв'язання науково-технічних, інженерних та математичних задач
	а) MATLAB Compiler	Компілятор для програм на мові програмування системи MATLAB. Транслює коди цих програм в програми на мові С. Застосування компілятора забезпечує можливість створення виконуваних кодів (програм), час виконання яких для програм з великим числом циклічних операцій зменшується у 10-15 раз.
	б) MATLAB C Math Library	Бібліотека додаткових математичних функцій на мові С, що розширює можливості базової системи по розв'язанню математичних задач.
	в) MATLAB C++ Math Library	Бібліотека додаткових математичних функцій на мові C++, що є більш сучасним варіантом мови С. Бібліотека розширює можливості базової системи по розв'язанню математичних задач.
2	Simulink for Windows	Система імітаційного моделювання моделей, що складаються з графічних блоків із заданими властивостями (параметрами). Компоненти моделей можуть бути одиничними графічними блоками та субмоделями, з особистою структурою та зв'язками. До складу моделей можна включати з бібліотеки Simulink джерела сигналів різного виду, віртуальні реєструючі пристрої, графічні засоби анімації.
3	Simulink Real Time Workshop (RTW)	Підсистема імітаційного моделювання в реальному масштабі часу (при наявності додаткових апаратних засобів у вигляді плат розширення ПК), що підключається до Simulink.
4	Excel Link	Програмний засіб для створення інтерфейсу зв'язку з табличним процесором класу Excel, створених фірмою Microsoft Inc. Пакет забезпечує двосторонній зв'язок та обмін даними, перегляд, редагування та обробка даних з середовища MATLAB в Excel, підготовка Excel-додатків та ін.
5	NAG Foundation Toolbox	Одна з найбільших бібліотек математичних функцій, створена групою The Numerical Algorithms Group Ltd. Пакет вміщує понад 240 функцій, організованих у вигляді М-файлів.

1	2	3
6	Neural Networks Toolbox	Пакет прикладних програм, що має засоби для побудови нейронних мереж, які ґрунтуються на поведінці нейрона. Пакет забезпечує ефективну підтримку проектування, навчання та моделювання багатьох відомих мережових парадигм, від базових моделей персептрона до найсучасніших асоціативних та самоорганізуючихся мереж. Пакет застосовується для обробки сигналів, нелінійного управління, фінансового моделювання та ін.
7	Spline Toolbox	Пакет прикладних програм для роботи зі сплайнами. Підтримує одновимірну та багатовимірну сплайн-інтерполяцію, апроксимацію. Забезпечує подання і відображення складних масивів даних і підтримку графіки.
8	Statistics Toolbox	Пакет прикладних програм по статистиці. Вміщує великий набір засобів генерації випадкових чисел, векторів, матриць та масивів з різними законами розподілу, а також великий набір статистичних функцій. Має кілька інтерактивних інструментів для динамічної візуалізації та аналізу даних.
9	Optimization Toolbox	Пакет прикладних програм для розв'язання задач оптимізації і систем нелінійних рівнянь.
10	Fuzzy Logic Toolbox	Пакет прикладних програм для розв'язання задач, що стосуються теорії розмитих множин.
11	Partial Differential Equations Toolbox	Пакет прикладних програм для розв'язання систем диференціальних рівнянь у частинних похідних. Надає ефективні засоби для розв'язання систем диференціальних рівнянь, у тому числі і жорстких. В пакеті використовується метод кінцевих елементів.
12	Symbolic Math Toolbox	Пакет прикладних програм для розв'язання задач у символьному (аналітичному) вигляді, з реалізацією точної арифметики довільної розрядності.
13	Control System Toolbox	Пакет для моделювання аналізу і проектування систем автоматичного керування: безперервних та дискретних.
14	Nonlinear Control Design Toolbox	Пакет призначений для побудови нелінійних систем контролю та управління. Він реалізує метод динамічної оптимізації для систем керування.
15	Robust Control Toolbox	Пакет має засоби для проектування і аналізу багатопараметричних стійких систем керування. Це системи з модельними похибками, динаміка яких відома не повністю чи параметри яких можуть змінюватися в процесі моделювання.
16	LMI Control Toolbox	Пакет забезпечує інтегроване середовище для постановки і розв'язання задач лінійного програмування. Призначений попередньо для проектування систем керування, пакет дозволяє розв'язувати будь-які задачі лінійного програмування практично в будь-якій сфері діяльності.
17	System Identification Toolbox	Пакет вміщує засоби для створення математичних моделей динамічних систем на основі спостереження вхідних-вихідних даних.
18	Signal Processing Toolbox	Пакет призначений для аналізу, моделювання та проектування пристроїв обробки різноманітних сигналів, забезпечення їх фільтрації та інших перетворень.
19	Image Processing Toolbox	Пакет подає широкий спектр засобів для цифрової обробки і аналізу зображень.
20	Wavelet Toolbox	Пакет подає повний набір програм для дослідження багатовимірних нестационарних явищ. Методи пакету розширюють можливості користувача в тих областях, де зазвичай застосовується техніка Фур'є-розкладань. Пакет може застосовуватися для обробки звуків мови і аудіосигналів, телекомунікації, при вирішенні задач геофізики, фінансів, медицини.

Система MATLAB може використовуватися в режимі безпосередніх обчислень – командному режимі, а також – в режимі програмування. Система MATLAB багатопланова, і тому може виникнути питання – для чого використовувати програмування, якщо величезну кількість задач є можливість розв’язати за допомогою внутрішніх та додаткових ресурсів системи. Та навіть у найбільшій і найпотужнішій системі нереально передбачити всіх засобів для розв’язання специфічних задач, що стоять перед користувачем. Мова програмування системи MATLAB дозволяє їх ефективно розвиватися, створювати і використовувати в системі найновіші методи розв’язання сучасних задач.

Програмування в системі MATLAB.

Програмами в MATLAB є М-файли текстового формату, які вміщують запис програми у вигляді програмних кодів. Засоби мови програмування MATLAB:

- дані різних типів,
- константи і змінні,
- оператори, в тому числі і оператори математичних виразів,
- вбудовані команди і функції,
- функції користувача,
- керуючі структури,
- системні оператори і функції,
- засоби розширення мови.

Коди програм в системі MATLAB пишуться на мові високого рівня, яка є достатньо зрозумілою для користувачів середньої кваліфікації в області програмування. Мова програмування MATLAB є типовим інтерпретатором. Це значить, що кожна інструкція програми розпізнається і зразу виконується, що полегшує забезпечення діалогового режиму роботи з системою. Етап компіляції всіх інструкцій відсутній. Висока швидкість виконання програми забезпечена

- наявністю заздалегідь відкомпільованого ядра, в якому зберігаються критичні, щодо швидкості виконання, інструкції, наприклад такі, як цикли і базові математичні функції,
- ретельним відпрацюванням системи контролю синтаксису програм в режимі інтерпретації.

Інтерпретація значить, що MATLAB не створює виконуваних кінцевих програм. Вони існують у вигляді М-файлів. Для виконання програм необхідно знаходитися у середовищі MATLAB. Але для програм на мові MATLAB створені компілятори, що транслюють програми MATLAB в коди мови програмування С та С++. Це вирішує завдання створення виконуваних програм, що на початку створюються у середовищі MATLAB.

Основні типи даних.

Структура типів даних системи MATLAB подана на рис.1.

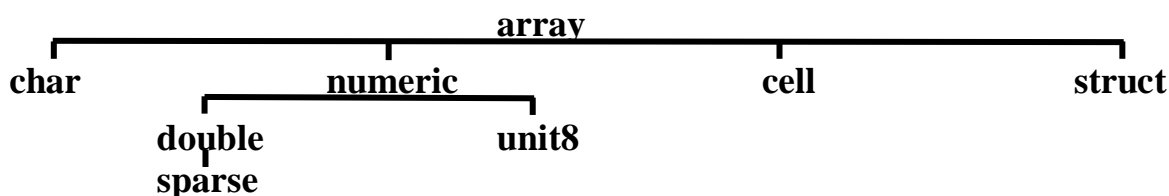


Рис. 1. Структура типів даних в MATLAB.

Типи даних **array** та **numeric** є віртуальними, оскільки до них не можна віднести, будь-які змінні. Вони служать для визначення і комплектування деяких типів даних.

Таким чином, в MATLAB визначені шість основних типів даних, що є багатовимірними масивами:

- **double** – числові масиви з числами подвоєної точності,

- **char** – рядкові масиви з елементами-символами,
- **sparse** – розріджені матриці з елементами-числами подвоєної точності,
- **cell** – масиви комірок, котрі у свою чергу можуть бути масивами,
- **struct** – масиви записів з полями, які також можуть вміщувати масиви,
- **uint8** – масиви 8-розрядних цілих чисел без знаків (математичні операції з ними не передбачені), застосовуються для службових цілей.

Крім того, передбачений ще один тип даних **UserObject**, який належить до типів даних (об'єкти), що визначаються користувачем.

Кожному типу даних відповідають деякі характерні для нього операції, що називаються **методами**. Дочірні типи даних, що розташовані на рис.1 нижче батьківських типів, наслідують від останніх їх методи, що є ознакою наслідування об'єктів. Начолі ієрархії типів даних знаходяться дані типу **array**, а значить, що всі види даних в MATLAB є масивами.

Мова програмування MATLAB включає в себе всі засоби, що необхідні для реалізації різних видів програмування:

- процедурного,
- операторного,
- функціонального,
- логічного,
- структурного (модульного),
- об'єктно-орієнтованого,
- візуально-орієнтованого.

В основі **процедурного, операторного та функціонального** типів програмування лежать процедури, оператори і функції, які використовуються як основні об'єкти мови MATLAB. **Логічне** програмування реалізується в MATLAB за допомогою логічних операторів та функцій. В системі наглядно реалізований принцип **структурного** програмування – більшість функцій і команд мови є закінченими модулями, обмін даними між якими відбувається через їх вхідні параметри, хоча можливий і через глобальні змінні. Програмні модулі оформлені у вигляді текстових М-файлів, які зберігаються на диску і підключаються до програм по мірі необхідності. На відміну від багатьох інших мов програмування, застосування модулів не вимагає попереднього об'явлення, а для створення і відлагодження самостійних модулів MATLAB має всі необхідні засоби. Переважна більшість команд і функцій системи MATLAB представляється у вигляді таких модулів.

Об'єктно-орієнтоване програмування також широко представлене в системі MATLAB. Воно особливо проявляється при програмуванні задач графіки. **Візуально-орієнтоване** програмування в MATLAB представлено, в основному, в пакеті Simulink при моделюванні заданих блоками пристроїв та систем.

Двоїстість операторів і функцій в MATLAB полягає у можливості їх виконання як з програм, так і в режимі прямих обчислень. Під **командами** ми будемо розуміти засоби, що керують периферійними пристроями, а під **операторами** – засоби, що виконують операції з конкретними операндами (даними).

Функція перетворює одні дані у інші. Для багатьох функцій характерне повернення значень у відповідь на звертання до них зі вказівкою списку вхідних параметрів. Наприклад: $\sin(x)$. Більшість функцій можна використовувати в арифметичних виразах.

Для операторів (а також команд), які не повертають значення, їх застосування в арифметичних виразах абсурдно. Наприклад: вираз **2*plot(x, y)**, де **plot(x,y)** – оператор побудови графіка функцій, буде сприйнятий системою як помилка.

Всі функції, що повертають єдине значення (чи один масив), записуються малими буквами:

f_name(Список_параметрів)

Функції, що повертають ряд значень чи масивів, наприклад X, Y, Z, записуються так:

[X, Y, Z, ...]=f_name(Список_параметрів)

Оскільки мова програмування системи MATLAB орієнтований на структурне програмування, у ньому немає номерів рядків і програмних операторів безумовного переходу GOTO. Є лише керуючі структури наступних типів:

- умовні вирази **if...else...elseif...end**,
- цикли **for...end**, **while...end**, .

Їх форма схожа на використовувану в мові Pascal (область дії керуючих структур виділяється їх заголовком і словом **end**, але без слова **begin**).

Для дострокового завершення функції і виходу з неї використовуються інструкції **return** та **break**.

Ще один оператор розгалуження є оператор переключення **switch...case...otherwise...end**. Рядків з ключовим словом **case** може бути багато, а рядок з ключовим словом **otherwise** повинен бути один.

Для створення М-файлів використовують як вбудований редактор системи, так і будь-який текстовий редактор, що підтримує формат ASCII. Підготований і записаний на диск М-файл стає частиною системи. Його можливо викликати як з командного рядка, так і з іншого М-файлу. Розрізняють два типи М-файлів: файли-сценарії та файли-функції. В процесі свого створення вони проходять синтаксичний контроль за допомогою вбудованого в систему MATLAB редактора і відладчика М-файлів.

Файл-сценарій (Script-файл) – запис серії команд без вхідних і вихідних параметрів, має наступну структуру:

```
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами.
```

Властивості файлів-сценаріїв:

- вони не мають вхідних і вихідних аргументів, змінні цих файлів є глобальними,
- працюють з даними з робочої області,
- в процесі виконання не компілюються,
- є зафіксованою у вигляді файлу послідовністю операцій, повністю аналогічно тій, що використовується у робочій області.

Наприклад, побудуємо файл сценарію з назвою **p_red**, що буде графік синусоїди червоною лінією:

```
%Plot with color red
% Буде графік синусоїди лінією червоного кольору
%с виділеною масштабною сіткою в інтервалі [xmin, xmax]
x=xmin:0.1:xmax;
plot(x, sin(x),'r')
grid on
```

В даній програмі перші три рядка є коментарями: перший рядок – основний коментар, який виводиться при виконанні команд **lookfor** та **help ім'я_каталогу**. Повний коментар виводиться при виконанні команди **help ім'я_файлу**. Знак % повинен починатися з першої позиції рядка, інакше коментар не сприйматиметься командою **help** і може видавати помилку. Далі у прикладі йде безпосередньо перелік операцій. Для того, щоб скористатися файлом необхідно задати початкові дані (перегляд файлу – команда **type p_red**) і звернутися до файлу:

```
>> xmin=-10;
>> xmax=10;
>> p_red
```

М-файл-функція є об'єктом мови програмування системи MATLAB. Файл-функція є повноцінним модулем з точки зору структурного програмування, оскільки вміщує вхідні та

вихідні параметри і використовує апарат локальних змінних. Структура такого модуля виглядає так:

```
function var=f_name(список_параметрів)
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами
var=вираз
```

Файл-функція має наступні властивості:

- починається з об'явлення типу **function**, після якого вказується ім'я змінної – вихідного параметру, ім'я самої функції та список її вхідних параметрів,
- функція повертає своє значення і може використовуватися у вигляді **f_name(список_параметрів)** в математичних виразах,
- всі змінні, що використовуються у тілі функції є локальними, тобто діють тільки у межах функції,
- файл-функція є самостійним програмним модулем, який спілкується з іншими модулями через свої вхідні та вихідні параметри,
- правила виводу коментарів ті ж, що і у файлів-сценаріїв,
- файл-функція є засобом розширення системи MATLAB,
- при звертанні до файла-функції, він компілюється і виконується, а створені машинні коди зберігаються в робочій області системи MATLAB.

Остання конструкція **var=вираз** вводиться, якщо треба, щоб функція повертала результат обчислень. Якщо файл-функція завершується рядком із точкою з комою (;), то для повертання значення функції використовується програмний оператор **return**.

Якщо функція має більше одного вхідного параметра, то вони вказуються у квадратних дужках після слова **function**. Структура такого модуля має вигляд:

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

Така функція нагадує повноцінну процедуру. Її не можна використовувати безпосередньо в математичних виразах, оскільки вона повертає не єдиний результат, а їх множину. Тому функція використовується як окремий елемент програм виду:

```
[var1,var2,...]=f_name(список_параметрів)
```

Після застосування елемента вихідні змінні стають визначеними і їх можливо використовувати в подальших математичних обчисленнях та інших сегментах програми.

Змінні у файлах-сценаріях є глобальними, а у файлах-функціях – локальними. Застосування глобальних змінних у програмних модулях може приводити до небажаних результатів, але у деяких випадках застосування глобальних змінних у файлах-функціях є більш ефективним. Відповідальність за безпомилкову роботу в таких випадках бере на себе програміст.

Об'являє змінні модуля функції глобальними команда:

```
global var1 var2 ...
```

Застосування підфункцій. Підфункції записуються в тілі основних функцій і мають ідентичну структуру. Вони не належать до внутрішніх функцій, вбудованих в ядро MATLAB. Вони не можуть використовуватися за межами М-файлу, що визначає основну функцію. Приклад функції з підфункцією:

```
function [mean,stdev] = statv(x)
%Statv Interesting statistics.
%Приклад функції з вбудованою підфункцією
```

```

n = length (x)
mean = avg(x,n)
stdev = sqrt(sum ((x-avg(x,n)).^2)/n)
%_____
function m = avg(x,n)
%Mean subfunction
m = sum(x)/n;

```

У цьому прикладі середнє значення елементів вектора *x* знаходиться за допомогою підфункції **avg(x,n)**, тіло якої записано в тілі основної функції **statv**. При використанні функції **statv** отримаємо наступні результати:

```

>> V=[1 2 3 4 5]
V =
     1     2     3     4     5
>>[a,m]=statv(V)
a =
     3
m =
    1.4142
>>statv(V)
ans =
     3
>> help statv
Statv Interesting statistics.

```

Приклад функції з вбудованою підфункцією

Для перегляду лістингу М-файлу використовують команду **type <name>**.

Приватні каталоги. Для запису М-файлів використовуються каталоги, що називаються батьківськими. Вони вміщують групи файлів певного функціонального призначення. Але з версії MATLAB 5.0 з'явилася можливість в батьківських каталогах створювати приватні каталоги – PRIVATE. Файли, що в них входять, доступні тільки файлам батьківського каталогу. Файли приватних каталогів передивляються інтерпритатором системи MATLAB в першу чергу. Застосування приватних каталогів дозволяє використовувати функції, написані користувачем, надаючи їм імена оригінальних файлів.

Використання рядкових виразів. Рядкові вирази зазвичай не обчислюються, а просто повторюють рядок:

```

>> '2+3'
ans =
    2+3

```

Але за допомогою функції **eval('Рядковий_вираз')** рядкові вирази, що є математичними виразами можуть бути обчислені:

```

>>eval('2+3')
ans =
     5
>>eval('2*sin(1)')
ans =
    1.6829

```

Функція **feval('Ім'я_функції',x1,x2,...)** дозволяє передавати в обчислювану функцію список її аргументів. В цьому випадку обчислювана функція записується тільки своїм іменем. Наприклад:

```

>>feval('prod',[1 2 3])
ans =
     6

```

```
>>feval('sum',[1 2 3; 4 5 6],2)
ans =
    6
   15
```

Видача повідомлення про помилку. В ході обчислень можуть виникати помилки, наприклад ділення на нуль. В таких випадках повинні видаватися повідомлення про помилку. Для виводу повідомлень про помилки служить команда

```
error('Повідомлення про помилку')
```

при виконанні якої обчислення припиняються і видається повідомлення про помилку, що задане в апострофах. Наприклад:

```
>> function f=sd(x)
if x==0 error('Помилка – ділення не нуль'), end
f=sin(x)/x
```

Результати виконання програми:

```
>>sd(1)
f =
    0.8415
ans =
    0.8415
```

```
>>sd(0)
??? Error using ==> sd
```

Помилка – ділення не нуль

Якщо призупинення програми є небажаним, то можливо використовувати команду попереджувального повідомлення:

```
warning('Попереджувальне повідомлення')
```

Ця команда виводить повідомлення, що вказане у апострофах, але не заважає подальшому виконанню програми. Ознакою помилки є символи '??', в ознакою попередження слово **warning** з відповідним повідомленням.

При створенні функцій зі спеціальними властивостями будуть потрібні наступні функції:

nargin – повертає число вхідних параметрів даної функції

nargout – повертає число вихідних параметрів даної функції.

Робота з багатовимірними масивами. При задаванні масиву елементи рядка розділяються пробілами, а елементи різних рядків відділяються знаком ; . При необхідності розширення масиву – додавання нової “сторінки” – користуються оператором : (двокрапка). Наприклад заданий масив:

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

При додаванні нової сторінки з тим же розміром можливо задати:

```
>> M(:,2)= [10 11 12; 13 14 15; 16 17 18]
M(:,1) =
    1    2    3
    4    5    6
    7    8    9
M(:,2) =
   10   11   12
   13   14   15
   16   17   18
```

Після явного виклику такого масиву отримаємо:

```
>> M
M(:,:,1) =
    2    2    3
    5    5    6
    7    8    9
M(:,:,2) =
   10   11   12
   13   14   15
   16   17   18
```

Щоб отримати доступ до певного елемента масиву треба вказати номер рядка і стовпчика в якому знаходиться елемент:

```
>> M(2,2,1)
ans =
    5
>> M(2,3,2)
ans =
   15
```

При створенні сторінок можна заповнювати їх константами і випадковими величинами. При заповненні константами вони вказуються вищенаведеним способом або, якщо весь масив треба заповнити однаковими константами:

```
>> M(:,:,1)=0
M(:,:,1) =
    1    1    1
    1    1    1
    1    1    1
```

Для створення масивів заповнених різним чином використовуються наступні функції:

ones – створення масивів з одиничними елементами,

zeros – створення масивів з нульовими елементами,

rand – створення масивів з випадковими елементами, розподіленими за рівномірним розподілом,

randn – створення масивів з випадковими елементами, розподіленими за нормальним розподілом.

Наприклад:

```
>> Z=zeros(2,2,3)
Z(:,:,1) =
    0    0
    0    0
Z(:,:,2) =
    0    0
    0    0
Z(:,:,3) =
    0    0
    0    0
>> R=randn(3,2,2) =
R(:,:,1) =
   -1.6656   -1.1465
    0.1253    1.1909
    0.2877    1.1892
R(:,:,2) =
   -0.0376   -0.1867
    0.3273    0.7258
    0.1746   -0.5883
```

Якщо хоча б одна розмірність масиву буде нульовою – масив буде пустим:


```
>> A=randn(3,3,0)
```

```
A =
```

Empty array: 3-by-3-by-3-by-0

При використанні функції конкатенації отримуємо масив, що формується об'єднанням з початкових масивів:

cat(2,A,B) – повертає масив [A,B], що об'єднаний по стовпчикам,

cat(1,A,B) – повертає масив [A,B], що об'єднаний по рядкам.

Наприклад:

```
>> M1=[1 2; 3 4]
```

```
M1 =
```

```
1    2
3    4
```

```
>>M2=[5 6; 7 8]
```

```
M2 =
```

```
5    6
7    8
```

```
>>cat(1,M1,M2)
```

```
ans =
```

```
1    2
3    4
5    6
7    8
```

```
>>cat(2,M1,M2)
```

```
ans =
```

```
1    2    5    6
3    4    7    8
```

```
>>M=cat(3,M1,M2)
```

```
M(:,:,1) =
```

```
1    2
3    4
```

```
M(:,:,2) =
```

```
5    6
7    8
```

Для обчислення розмірності масиву існує функція:

ndims(A) – повертає розмірність масиву A (якщо вона більше чи дорівнює 2).

Оператори і функції системи MATLAB

Повний набір операторів вхідної мови системи MATLAB і відповідних функцій виводяться командою

```
>>help ops
```

Арифметичні оператори та функції

Функція	Назва	Оператор	Синтаксис
plus	плюс	+	M1 + M2
uplus	унарний плюс	+	+ M
minus	мінус	-	M1 – M2
uminus	унарний мінус	-	- M
mtimes	матричне множення	*	
times	почленне множення масивів	.*	
mpower	зведення до ступеня матриці	^	
power	почленне зведення до ступеня масиву	.^	
mldivide	ділення матриць зворотнє (справа-наліво)	\	

mrdivide	ділення матриць зліва-направо	/	
ldivide	почленне ділення масивів (справа-наліво)	.\	
rdivide	почленне ділення масивів (зліва-направо)	./	
kron	тензорне множення Кронекера	kron	

Оператори відношення та їх функції

Функція	Назва	Оператор	Синтаксис
eq	Дорівнює	==	x == y
ne	Не дорівнює	~=	x ~= y
lt	Менше ніж ...	<	x < y
gt	Більше ніж ...	>	x > y
le	Менше чи дорівнює	<=	x <= y
ge	Більше чи дорівнює	>=	x >= y

Логічні оператори

Функція	Назва	Позначення
and	Логічне і	&
or	Логічне чи	
not	Логічне ні	~
xor	Виключаюче чи	
any	Вірно, якщо всі елементи вектора дорівнюють нулю	
all	Вірно, якщо всі елементи вектора не дорівнюють нулю	

[] – використовуються для формування векторів та матриць.

**Рекомендована література
до вивчення дисципліни „Чисельні методи в інформатиці”.**

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Лаборатория базовых знаний, 2002. – 632 с.
2. Бахвалов Н.С., Лапин А.В., Чижонков Е.В. Численные методы в задачах и упражнениях. – М.: Высшая школа, 2000. – 192 с.
3. Метьюз Дж., Фінк К. Численные методы. Использование MatLab. – М.: 2001. – 583 с.
4. Фельдман Л.П., Петренко А.І., Дмитрієва О.А. Чисельні мтеоди і інформатиці. – К.: Видавнича група ВНУ, 2006. – 480 с.
5. Супруненко О.О. Чисельні методи в інформатиці. Курс лекцій: для студентів, які навчаються за напрямом підготовки 050101 „Комп’ютерні науки”, 050103 „Програмна інженерія”. – Черкаси: ЧНУ, 2009. – 132 с.