

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ  
Факультет інформаційних технологій

В.Б. ХОЦКІНА  
І.Н. ВДОВИЧЕНКО  
**РОБОТА В ПАКЕТІ МАТЛАВ**  
**НАВЧАЛЬНИЙ ПОСІБНИК**

м. Кривий Ріг  
2023 р.

Рецензенти:

А.І. Купін – доктор технічних наук, професор, завідувач кафедри комп'ютерних систем та мереж Криворізького національного університету;

І.О. Музика, кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, декан факультету інформаційних технологій Криворізького національного університету;

В.С. Лисенко, кандидат економічних наук, доцент кафедри інформатики та прикладного програмного забезпечення Державного університету економіки і технологій.

Рекомендовано до друку Вченою радою Державного університету економіки і технологій протокол № 9 від 23.02.2023р.

В.Б. Хоцкіна, І.Н. Вдовиченко Робота в пакеті MATLAB: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. – 130 с.

Навчальний посібник призначений для студентів другого (магістерського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення» денної та заочної форм навчання.

Подано теоретичний матеріал з основ роботи в пакеті MATLAB: прості обчислення, робота з масивами, високорівнева графіка, інтерактивне середовище для побудови графіків.

Теоретичний матеріал закріплено практичною частиною завдань, які відповідають темам, що розглядаються. Для перевірки знань за кожною темою надаються індивідуальні завдання.

Навчальний посібник розроблено у відповідності до навчального плану з метою надання здобувачам вищої освіти допомоги у освоєнні теоретичного матеріалу та при виконанні ними лабораторних та індивідуальних завдань з освітньої компоненти «Робота в пакеті MATLAB».

## ЗМІСТ

ПЕРЕДМОВА .....	5
Історія розвитку MatLab .....	7
Робоче середовище MatLab .....	10
Арифметичні обчислення .....	13
Формати виведення результату обчислень .....	14
Формати чисел.....	14
Специфіка виконання арифметичних і логічних операцій .....	15
Використання елементарних функцій.....	16
Вбудовані елементарні функції .....	17
Приклади на округлення і залишок від ділення .....	18
Основні об'єкти MatLab.....	18
Поняття про математичний вираз .....	18
Дійсні числа.....	18
Константи і системні змінні.....	19
Використання змінних. Перенесення рядку в сесії .....	19
Текстові коментарі .....	20
Змінні і привласнення їм значень.....	21
Оператори і функції .....	21
Індивідуальні завдання за темою «Лінійний обчислювальний процес» ...	23
Програмування розгалужень .....	24
Логічні вирази і вирази відносин .....	24
Умовний оператор if .....	25
Організація розгалуження.....	25
Індивідуальні завдання за темою «Умовний оператор if» .....	36
Керуючі конструкції мови програмування .....	38
Цикл for.....	38
Індивідуальні завдання за темою «Оператори циклу» Цикл <i>for</i> .....	46
Ітераційні цикли. Цикл while, знаходження суми ряду.....	47
Індивідуальні завдання за темою «Ітераційні цикли» Цикл <i>while</i> .....	50
Індивідуальні завдання за темою «Поєднання розгалуженого і циклічного обчислювальних процесів» .....	53
Робота з масивами .....	56
Формування векторів і матриць .....	56
Особливості задання векторів і матриць.....	56
Використання оператора двокрапка (:)	57
Введення, додавання і віднімання векторів .....	58
Звернення до елементів вектора .....	64
Застосування функцій обробки даних до векторів .....	65
Поелементні операції з векторами .....	66
Добуток векторів .....	68
Індивідуальні завдання за темою «Робота з векторами. Обробка даних в векторах» .....	76
Двомірні масиви, матриці .....	77

Введення матриць, прості операції .....	77
Звернення до елементів матриці.....	79
Додавання, віднімання, множення, транспонування.....	80
Перемноження матриці і вектору.....	81
Індивідуальні завдання за темою «Робота з матрицями. Операції з матрицями».....	85
Застосування функцій обробки даних до матриць .....	87
Індивідуальне завдання за темою: «Застосування функцій обробки даних до матриць» .....	92
Графіка MATLAB.....	92
Побудова графіків відрізками прямих. Графіки функцій однієї змінної ..	92
Оформлення графіків .....	95
Індивідуальні завдання за темою: «Графіки функцій однієї змінної» .....	96
Кольорові плоскі кругові діаграми.....	98
Кольорові об'ємні кругові діаграми.....	100
Індивідуальні завдання за темою: «Кругові і стовбцеві діаграми» .....	105
Графіки параметрично-заданих функцій .....	109
Індивідуальні завдання за темою «Графіки параметрично заданих функцій» .....	111
Високорівнева графіка .....	112
Побудова циліндра у вигляді трьохмірної фігури.....	112
Виведення графіків в різні графічні вікна за допомогою команди figure	112
Індивідуальні завдання за темою: «Побудова циліндра. Виведення графіків в різні графічні вікна» .....	118
Високорівнева графіка. Побудова сфери як трьохмірної фігури. ....	118
Виведення різних графіків в одному графічному вікні за допомогою команди subplot .....	122
Індивідуальні завдання за темою: «Виведення декількох графіків в одному графічному вікні».....	128
ВИСНОВКИ.....	129
СПИСОК ЛІТЕРАТУРИ.....	130

## ПЕРЕДМОВА

Сучасна комп'ютерна математика пропонує цілий набір інтегрованих програмних систем і пакетів програм для автоматизації математичних розрахунків: Eureka, Gauss, TK Solver, Derive, Mathcad, Mathematica, Maple V ін. MatLab – це ретельно опрацьована та перевірена часом система автоматизації математичних розрахунків, побудована на розширеному поданні та застосуванні матричних операцій. Спектр можливостей відображається у назві системи – MATrix LABoratory – матрична лабораторія. Синтаксис мови програмування системи продуманий настільки ретельно, що ця орієнтація майже не відчувається користувачами, яких не цікавлять безпосередньо матричні обчислення.

Матриці широко застосовуються в складних математичних розрахунках, наприклад при вирішенні задач лінійної алгебри та математичного моделювання статичних і динамічних систем і об'єктів. В даний час MatLab вийшла далеко за межі спеціалізованої матричної системи і стала однією із найбільш потужних універсальних інтегрованих систем. Визначення «інтегрована» говорить про те, що в цій системі об'єднані зручна оболонка, редактор виразів і текстових коментарів, обчислювач і графічний програмний процесор. В новій версії використовуються такі потужні типи даних, як багатовимірні масиви, масиви осередків, масиви структур, масиви Java і розріджені матриці, що відкриває можливості застосування системи при створенні і налагодженні нових алгоритмів матричних і заснованих на них паралельних обчислень і великих баз даних.

MatLab – це високорівнева мова і інтерактивне середовище для програмування, чисельних розрахунків і візуалізації результатів. За допомогою MatLab можна аналізувати дані, розробляти алгоритми, створювати моделі та додатки.

Система MatLab являє собою унікальний сплав універсальних програмних і алгоритмічних засобів з широкою гамою спеціалізованих додатків.

Matlab – це високопродуктивна мова для технічних розрахунків.

Типове використання MATLAB – це:

- математичні обчислення;
- створення алгоритмів;
- моделювання;
- аналіз даних, дослідження і візуалізація;
- наукова і інженерна графіка;
- розробка додатків, включаючи створення графічного інтерфейсу.

Вхідна мова і середовище програмування MatLab дуже близькі до сучасних систем візуального програмування на базі універсальних алгоритмічних мов типу Basic, C++, Java, Object Pascal. По ряду аспектів MatLab поступається зазначеним системам (режим інтерпретації, невеликий запас візуальних компонентів). Однак з його бібліотекою чисельних методів ні за обсягом, ні за якістю не може зрівнятися жодна з систем програмування.

Крім того, в пакеті MatLab ретельно відпрацьовані засоби візуалізації результатів обчислень і відображення різних графічних об'єктів.

На базі ядра MatLab створені численні розширення, щоб забезпечити моделювання і аналіз систем в різноманітних сферах людської діяльності.

Багато навчальних закладів використовують MatLab при підготовці фахівців різного профілю. Для сучасного інженера і науково-технічного працівника MatLab є незамінним інструментом моделювання та дослідження різних прикладних систем, перш за все, за рахунок використання готових рішень. Але не менш важливо навчитися створювати нові додатки, використовуючи програмні та алгоритмічні засоби MatLab, а також можливість об'єднання модулів, розроблених в різних системах програмування.

Використання системи MatLab в навчальному процесі дозволить зблизити дисципліни, пов'язані з інформатикою та чисельними методами, які часто читаються автономно.

Система MATLAB складається з п'яти основних частин.

Мова MATLAB. Це мова матриць і масивів високого рівня з керуванням потоками, функціями, структурами даних, введенням-виведенням і особливостями об'єктно-орієнтованого програмування.

Середовище MATLAB. Це набір інструментів і пристосувань, з якими працює користувач або програміст MATLAB. Вона включає в себе засоби для управління змінними в робочому просторі MATLAB, введення і виведення даних, а також створення, контроль і налагодження М-файлів і додатків MATLAB.

Керована графіка. Це графічна система MATLAB, яка включає в себе команди високого рівня для візуалізації двовимірних і тривимірних даних, обробки зображень, анімації та ілюстрованої графіки. Також вона включає в себе команди низького рівня, що дозволяють повністю редагувати зовнішній вигляд графіки.

Бібліотека математичних функцій. Це велика колекція обчислювальних алгоритмів від елементарних функцій, таких як сума, синус, косинус, комплексна арифметика, до більш складних, таких як звернення до матриць, знаходження власних значень, функції Бесселя, швидке перетворення Фур'є.

Програмний інтерфейс. Це бібліотека, яка дозволяє писати програми на Сі та Фортрані, які взаємодіють з MATLAB. Вона включає засоби для виклику програм з MATLAB (динамічний зв'язок), викликаючи MATLAB як обчислювальний інструмент і для читання-запису MAT-файлів.

Simulink, супутня MATLAB програма, – це інтерактивна система для моделювання нелінійних динамічних систем. Вона являє собою середовище, кероване мишею. Ця програма дозволяє моделювати процес шляхом перетягування блоків діаграм на екрані і здійснювати маніпуляцію блоків. Simulink працює з лінійними, нелінійними, безперервними, дискретними, багатовимірними системами.

Blocksets – це доповнення до Simulink, які забезпечують бібліотеки блоків для спеціалізованих додатків, таких як зв'язок, обробка сигналів, енергетичні системи.

Real-Time Workshop – це програма, яка дозволяє генерувати С код з блоків діаграм і запускати їх на виконання на різних системах реального часу.

В цілому MatLab – це унікальна колекція реалізацій сучасних чисельних методів комп'ютерної математики. Вона увібрала в себе досвід, правила і методи математичних обчислень, накопичені за тисячі років розвитку математики. Це поєднання з потужними засобами графічної візуалізації і навіть анімаційної графіки. Систему з доданою до неї великою кількістю документації цілком можна розглядати як фундаментальний багатотомний електронний довідник з математичного забезпечення ЕОМ - від масових персональних комп'ютерів до супер-ЕОМ.

### Історія розвитку MatLab

Історію появи пакета MatLab пов'язують з ім'ям професора Кліва Б. Моулера. До переходу в фірму MathWorks він займався викладацькою і науково-дослідницькою діяльністю на кафедрах математики і в комп'ютерних центрах ряду університетів США (Нью-Мехіко, Мічиган, Стенфорд). Приблизно 30 років Моулер брав участь в розробці пакетів програм на мові Fortran для розв'язання задач лінійної алгебри (LINPACK) і в дослідженнях проблеми власних значень матриць (EISPACK). У 1980 році на міжнародній конференції AFIPS він представив доповідь, в якій вперше було озвучено назву MatLab.

Друге народження пакета MatLab пов'язують з Джеком Літлом – нинішнім президентом фірми MathWorks. Він на початку 80-х років минулого століття переніс програму MatLab на більш сучасні обчислювальні платформи VAX, Macintosh і IBM PC. Подальший розвиток пакета відбувся під егідою MathWorks. З моменту заснування фірми (1984 г.) К. Моулер є її беззмінним науковим керівником. Модернізацію та програмний супровід пакета MatLab забезпечують понад 1000 співробітників MathWorks.

Починаючи з 1984 року по теперішній час випущено більш ніж 50 версій MatLab. Першою версією була MatLab 1.0, яка вийшла у 1984 році. Недавно фірмою MathWorks випущена версія Matlab 9.7 R2019b.

В даний час реалізації пакету MATLAB налічують наступні версії:

1. MATLAB 1.0 – 1984 рік (випущений компанією MathWorks, написаний на С, працював на різних машинах).
2. MATLAB 2 – випущений в 1986 році.
3. MATLAB 3 – випущений в 1987 році.
4. MATLAB 3.5 – випущений в 1990 році (сумісний з MS-DOS).
5. MATLAB 4 – випущений в 1992 році (працював на Windows 3 та MAC).
6. MATLAB 4.2c – випущений в 1994 році.
7. MATLAB 5.0 – 1996 рік (випущений для усіх платформ).
8. MATLAB 5.1 – випущений в 1997 році.
9. MATLAB 5.1.1 R9.1 – випущений в 1997 році.
10. MATLAB 5.2 R10 – 1998 рік (це остання версія, яка працювала на Mac).
11. MATLAB 5.2.1 1 – випущений в 1998 році.
12. MATLAB 5.3 R11 – випущений в 1999 році.

13. MATLAB 5.3.1 R11.1 – випущений в 1999 році.
14. MATLAB 6.0 R12 – випущений в 2000 році, це був перший випуск віртуальної машини Java.
15. MATLAB 6.1 R12.1 – випущений в 2001 році.
16. MATLAB 6.5 R13 – випущений в 2002 році, включає в себе новий графічний користувацький інтерфейс для імпорту даних і надає графічне уявлення метаданих.
17. MATLAB 6.5.1 R13SP1 – випущений в 2003 році.
18. MATLAB 6.5.2 R13SP2 – випущений в 2003 році.
19. MATLAB 7 R14 – випущений в 2004 році, пакет представив вкладені функції і заново ввів Mac.
20. MATLAB 7.0.1 R14SP1 – випущений в 2004 році.
21. MATLAB 7.0.4 R14SP2 – випущений в березні 2005 року.
22. MATLAB 7.1 R14SP3 – випущений в вересні 2005 року (доступний для Windows XP-64 bit).
23. MATLAB 7.2 R2006a – випущений в березні 2006 року.
24. MATLAB 7.3 R2006b – випущений у вересні 2006 року.
25. MATLAB 7.4 R2007a – випущений в березні 2007 року (була введена функція Vsxfun для застосування бінарної операції).
26. MATLAB 7.5 R2007b – випущений у вересні 2007 року.
27. MATLAB 7.6 R2008a – випущений в березні 2008 року (зроблено покращення, пов'язані з можливостями об'єктно-орієнтованого програмування).
28. MATLAB 7.7 R2008b – випущений в жовтні 2008 року (зроблено оновлення до генераторів випадкових чисел).
29. MATLAB 7.8 R2009a – випущений в березні 2009 року, це був перший випуск для 32-бітної і 64-бітної Windows 7 Microsoft, інтерфейс с.NET framework був готовий.
30. MATLAB 7.9 R2009b – випущений в вересні 2009 року (було випущено перший випуск 64-бітної MAC-адреси Intel та було представлено новий оператор ~ ).
31. MATLAB 7.9.1 R2009bSP1 – випущений в квітні 2010 року (в цьому оновленні були виправлені помилки).
32. MATLAB 7.10a – випущений в березні 2010 року, це був останній випуск для 32-бітного Mac Intel.
33. MATLAB 7.11 R2010b – випущений у вересні 2010 року (додано допомогу для перерахувань).
34. MATLAB 7.11.1 R2010bSP1 – випущений в березні 2011 року (виправлення помилок та оновлення)
35. MATLAB 7.11.2 R2010bSP2 – випущений у квітні 2011 року (виправлення помилок та оновлення).
36. MATLAB 7.12 R2011a – випущений у квітні 2011 року (додано функцію генерації випадкових чисел).
37. MATLAB 7.13 R2011b – випущений у вересні 2011 року (змінені частини змінних).



38. MATLAB 7.14 R2012a – випущений у 2012 році (остання версія для підтримки 32-бітного Linux).
39. MATLAB 8 R2012b – випущений у 2012 році (випущено інтерфейс інструментальної смуги, перероблено систему документації).
40. MATLAB 8.1 R2013a – випущений в березні 2013 року (додано середовище модульного тестування).
41. MATLAB 8.2 R2013b – випущений у вересні 2013 року (додано тип даних таблиці, середовище виконання Java оновлено до версії 7).
42. MATLAB 8.3 R2014a – випущений у березні 2014 року (підтримка веб-камери USB).
43. MATLAB 8.4 R2014b – випущений у жовтні 2014 року (додано покращену користувальницьку панель інструментів, нові функції і пакети (для використання Python), лічильник веб-сторінок, гістограми, TCP-клієнт та інші).
44. MATLAB 8.5 R2015a – випущений у березні 2015 року. Це останній випуск, який підтримує Windows XP та Vista.
45. MATLAB 8.5 R2015aSP1 – випущений у жовтні 2015 року.
46. MATLAB 8.6 R2015b – випущений у 2015 році. Для роботи з графіками додано новий механізм виконання (LXE) та нові класи, такі як графіки та орграфіки.
47. MATLAB 9.0 R2016a – випущений у березні 2016 року. App Designer був доданий для створення та дизайну додатків. Введена нова кнопка паузи для того, щоб призупинити запуск під час виконання.
48. MATLAB 9.1 R2016b – випущений у вересні 2016 року (офіційний повзунок MatLab для Java, нові функції кодування та декодування JSON, додано новий «символьний» тип даних).
49. MATLAB 9.2 R2017a – випущений у березні 2017 року. Був доданий хмарний MatLab (MatLab Online), інфраструктура для модульного тестування, операції над великими масивами, функції запам'ятовування та створення діаграм теплових карт.
50. MATLAB 9.3 R2017b – випущений у вересні 2017 року. Додані нові функції, такі як власні функції та нові кольори діаграм з новими кольорами за замовчуванням.
51. MATLAB 9.4 R2018a – випущений у березні 2018 році. Додано різні нові функції для графіки та діаграм. Змінено структуру модульного контролю.
52. MATLAB 9.5 R2018a – випущений у вересні 2018 року.
53. MATLAB 9.6 R2019a – випущений у березні 2019 р (містить покращення для штучного інтелекту та аналітики, а також виправлення помилок).
54. MATLAB 9.7 R2019b – випущений у вересні 2019 року. Включає оновлення штучного інтелекту, нові продукти на підтримку робототехніки і нові ресурси для моделювання.

В поточний час понад десяток популярних комп'ютерних платформ можуть працювати з системою MatLab.

## Робоче середовище MatLab

Робота з пакетом MatLab починається із запуску пакета. Спочатку на екрані відкривається робоче середовище MatLab, представлене на рис. 1.

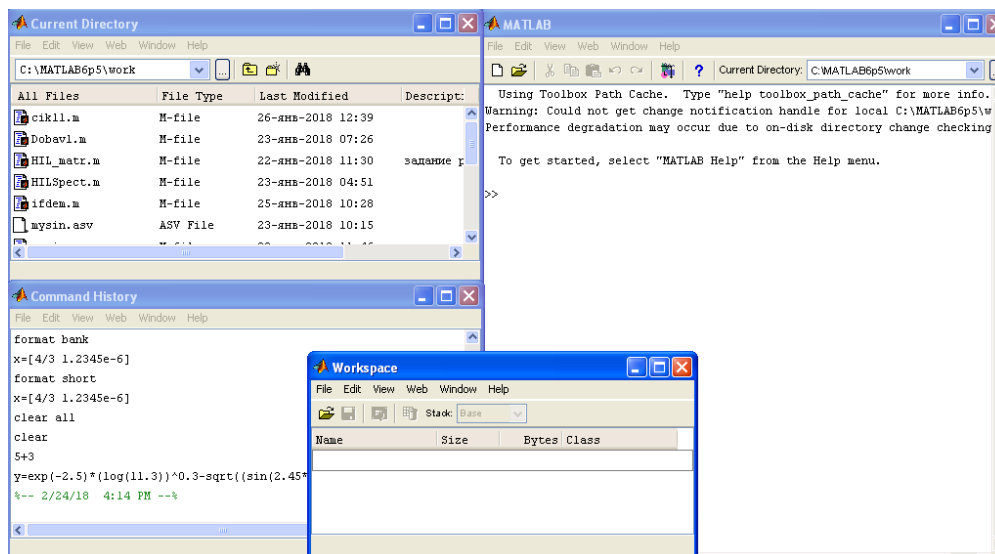


Рис. 1. Робоче середовище MatLab

Основними елементами робочого середовища є:

- меню;
- панель інструментів з кнопками і списком, що розкривається;
- вікна з вкладками *Workspace* і *Current Directory* для перегляду змінних і установки поточного каталогу;
- вікно *Command Window*, що служить для введення команд і виведення результату;
- вікно *Command History*, призначене для перегляду і повторного виконання раніше введених команд;
- рядок стану з кнопкою *Start*.

Для виклику на екран вікон існує команда меню *View*. Розкривши список команд за допомогою меню *View* потрібно вибрати рядок з найменуванням вікна, яке необхідно додати в робочу середу. Відкриті вікна в меню відзначені прапорцем (рис. 2).

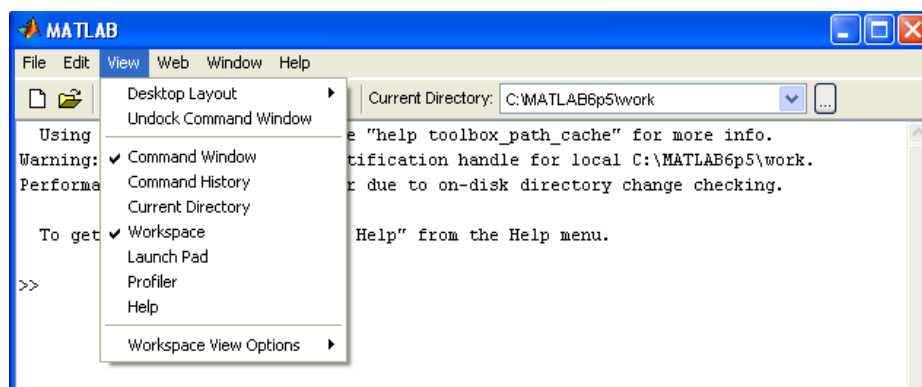


Рис. 2. Список команд меню *View*

Вікно *Command Window* являється основним вікном пакету. Воно служить для введення команд і виведення результату (рис. 3).

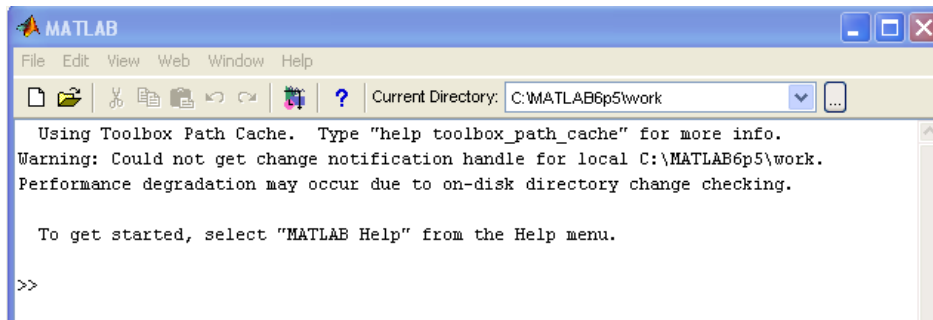


Рис. 3. Вікно *Command Window*

Вікно *Command Window* складається з наступних елементів:

- заголовка вікна, який містить назву вікна і три кнопки праворуч;
- робочої області, яка містить командний рядок. Позначкою командного рядку є наявність символу `>>`. Поряд знаходиться миготливий вертикальний курсор;
- смуги скролінгу (вертикальна – з правого боку вікна та горизонтальна – в нижній частині вікна).

Всі команди слід набирати в командному рядку після символу запрошення командного рядка `>>`. Набір будь-якої команди або виразу повинен закінчуватися натисканням на *Enter* для того, щоб програма MatLab виконала цю команду або вирахувала вираз, який введений в командний рядок робочої області. Для перегляду робочої області командного вікна зручно використовувати смуги скролінгу або клавіші *Home*, *End*, *PageUp*, *PageDown*. Якщо раптом після переміщення по робочій області командного вікна пропав командний рядок з миготливим курсором, необхідно натиснути клавішу *Enter*.

Інформація про змінні, які беруть участь в розрахунках відразу ж з'являється у вікні *Workspace* (рис. 4).

Name	Size	Bytes	Class
a	1x1	8	double array
ans	1x61	488	double array
x	1x61	488	double array
y	1x61	488	double array

Рис. 4. Вікно *Workspace*

Вікно складається із декількох стовпців. У першому стовпці *Name* записується ім'я змінної, вміст стовпця *Size* демонструє основний принцип роботи MatLab. Програма MatLab всі дані представляє у вигляді масивів. Стовпець *Bytes* показує скільки байтів пам'яті займає змінна. У стовпці *Class* вказується тип змінної.

Вікно *Command History* (рис. 5) містить історію команд. Це робить роботу користувача більш ефективною, позбавляючи від необхідності знову набирати введені раніше команди. Якщо клацнути на якій-небудь команді у вікні лівою кнопкою миші, то дана команда стає поточною. Поточна команда в вікні підсвічена синім. Для її виконання треба застосувати подвійне клацання миші. Зайву команду можна прибрати з вікна. Для цього її треба зробити поточною і видалити за допомогою клавіші *Delete*.

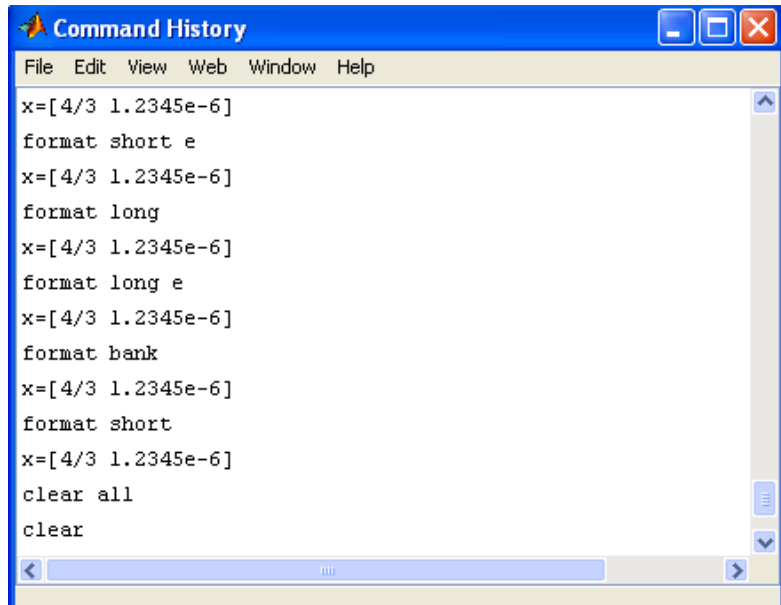


Рис. 5. Вікно *Command History*

Вікно *Current Directory* використовується для установки поточного каталогу (рис. 6).

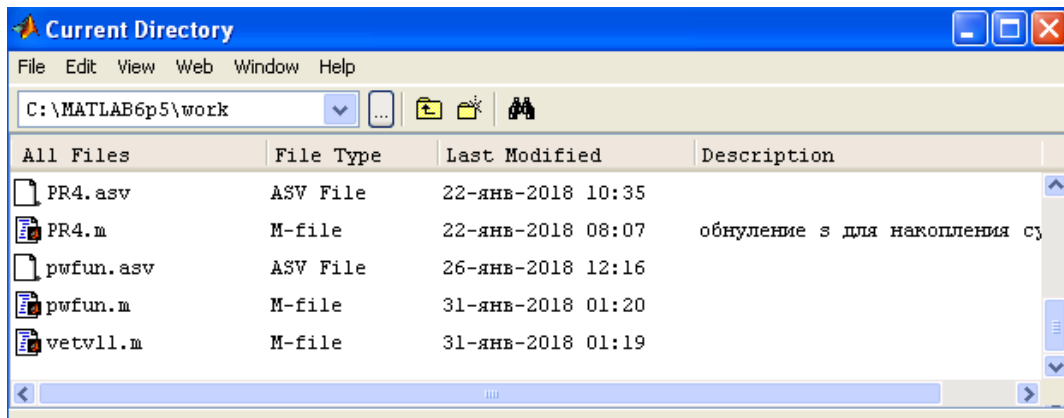
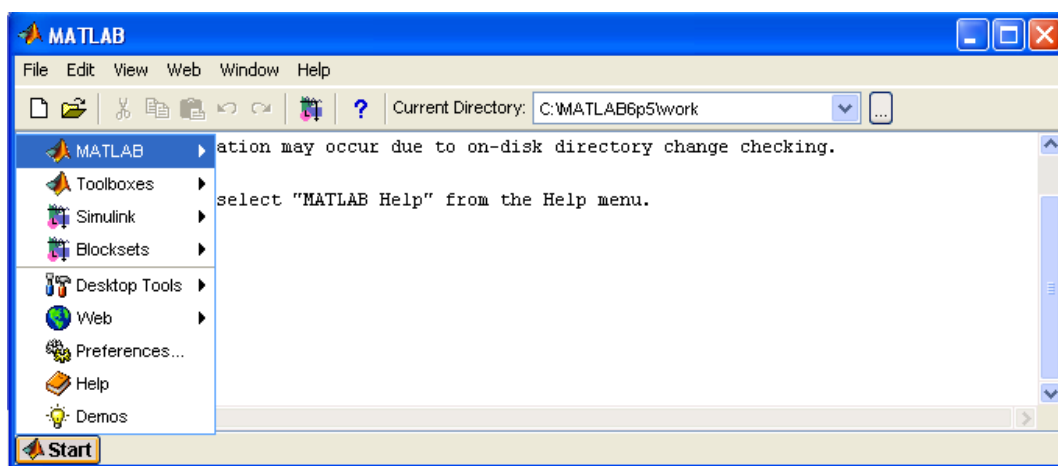


Рис. 6. Вікно *Current Directory*

У рядку стану робочого вікна MatLab розташована кнопка *Start*. При натисканні на кнопку *Start* відкривається меню, наведене на рис. 7. З його допомогою забезпечується доступ до всіх основних засобів MatLab.

Рис. 7. Спливаюче меню по кнопці *Start*

### Арифметичні обчислення

Пакет MatLab містить набір вбудованих функцій. Вбудовані математичні функції дозволяють знаходити значення різних виразів. MatLab надає можливість управління форматом виведення результату. Команди для обчислення виразів мають вигляд, властивий всім мовам програмування високого рівня. Наприклад, наберемо в командному рядку вираз:

```
>> 5+3
```

Натиснувши Enter отримаємо результат у командному рядку:

```
ans =
```

```
8
```

```
>>
```

Якщо при введенні у виразі використовуються десяткові дробки, то ціла частина відділяється від дробової *крапкою*.

Програма MatLab обчислює суму виразу, записує результат в спеціальну змінну *ans* і виводить її значення в командне вікно. Змінна *ans* створюється автоматично тоді, коли вираз, що обчислюється не присвоюється деякій змінній. Інформація про змінну *ans* відразу ж з'являється у вікні *Workspace* (рис. 4). У першому стовпці *Name* цього вікна записано ім'я змінної. Наступний стовпець *Value* показує значення змінної. Вміст стовпця *Size* демонструє основний принцип роботи MatLab. Програма MatLab всі дані представляє у вигляді масивів. Змінна *ans* є двовимірним масивом розміру один на один і займає 8 байт пам'яті, про що свідчить стовпець *Bytes*. В останньому стовпці *Class* зазначений тип змінної – *double array*, тобто масив, що складається з чисел подвійної точності. За замовчуванням всі числа представляються з подвійною точністю (*double*).

Після виводу результату у вікні *Command Window* наступним рядком після відповіді розташований командний рядок з символом `>>` і миготливим курсором. Це говорить про те, що навколишнє середовище MatLab готове до подальших обчислень.

## Формати виведення результату обчислень

### Формати чисел

Вид, в якому виводиться результат обчислень, залежить від формату виведення, встановленого в MatLab. За замовчуванням MatLab видає числові результати в *нормалізованій формі* з чотирма цифрами після десяткової крапки. Але при роботі з числовими даними можна задавати різні *формати* представлення чисел. Однак в будь-якому випадку усі обчислення проводяться з граничною *подвійною* точністю. Для установки формату представлення чисел використовується команда

```
>> format name
```

– де *name* – ім'я формату.

Для числових даних *name* може мати наступні представлення: *short* – коротке представлення в фіксованому форматі (5 знаків), *short e* – коротке представлення в експоненційному форматі (5 знаків мантиси і 3 знаки порядку), *long* – довге подання у фіксованому форматі (15 знаків), *long e* – довге подання у експоненційному форматі (15 знаків мантиси і 3 знаки порядку), *hex* – подання чисел в шістнадцятиричному форматі, *bank* – представлення для грошових одиниць. Приклад: для ілюстрації різних форматів розглянемо вектор, що містить два елементи-числа:

```
>> x=[4/3 1.2345e-6]
```

format short	1.3333	0.0000
format short e	1.3333e+000	1.2345e-006
format long	1.333333333333338	0.000001234500000
format long e	1.333333333333333e+000	1.234500000000000e-006
format bank	1.33	0.00

Задання формату позначається тільки на формі виведення чисел. Обчислення все одно відбуваються в форматі подвійної точності, а введення чисел можливе в будь-якому зручному для користувача вигляді. Необхідний формат виведення результату визначається користувачем з меню робочого середовища MATLAB. Виберемо в меню *File* пункт *Preferences*. На екрані з'явиться діалогове вікно *Preferences* (рис. 8):

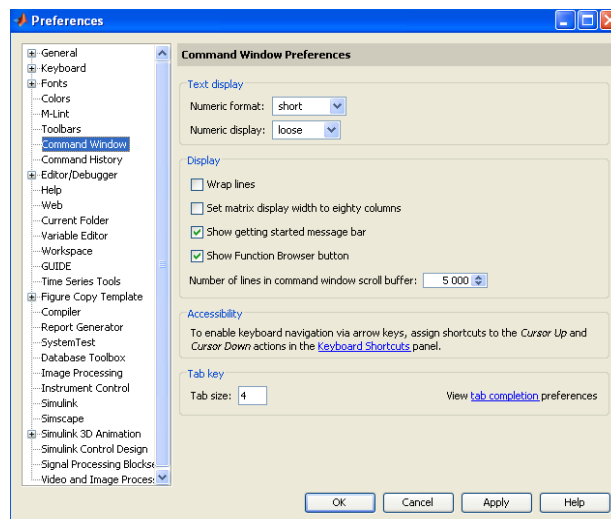


Рис. 8. Діалогове вікно *Preferences* MATLAB

Для установки формату виведення слід переконатися, що в списку лівої панелі діалогового вікна *Preferences* обраний пункт *Command Window*. Задання формату проводиться із списку *Numeric format* панелі *Text display* діалогового вікна. Натиснувши кнопку *OK* підтверджуємо установку вибраного формату і виконуємо закриття діалогового вікна.

Зараз встановлено короткий формат з плаваючою точкою *short* для виведення результатів обчислень, при якому на екрані відображаються тільки чотири цифри після десяткової крапки.

Наберемо в командному рядку  $100/3$  і натиснемо *Enter*. Результат виводиться у форматі *short*:

```
>> 100/3
ans =
    33.3333
```

Цей формат виведення збережеться для всіх наступних обчислень. За потреби необхідно встановити інший формат.

Якщо потрібно отримати результат обчислень більш точно, то слід вибрати на панелі *Text display* із списку *Numeric Format* значення *long*. Результат буде відображатися в довгому форматі з плаваючою крапкою *long* з чотирнадцятьма цифрами після десяткової крапки.

Інформацію про формати можна отримати, набравши в командному рядку команду *help* з аргументом *format*:

```
>> help format
```

У командному вікні з'являється опис кожного з форматів. Задавати формат виведення можна безпосередньо з командного рядка за допомогою команди *format*. Наприклад, для установки довгого формату виведення результатів обчислень з плаваючою крапкою слід ввести команду *format long e* в командному рядку:

```
>> format long e
```

Команду слід вводити малими літерами. Це особливість вбудованої довідки *help*. MATLAB розрізняє великі та малі літери.

Для більш зручного сприйняття результату MATLAB виводить результат обчислень через рядок після виразу, що обчислюється. Однак іноді буває зручно розмістити більше рядків на екрані, для чого слід в діалоговому вікні *Preferences* вибрати *compact* із списку *Numeric display*. Додавання порожніх рядків забезпечується вибором *loose* із списку *Numeric display*.

### Специфіка виконання арифметичних і логічних операцій

При складанні алгебраїчних виразів MatLab дозволяє використання традиційних знаків арифметичних операцій і символів спеціальних відносин.

Список наведено в таблиці:

Символи	Дія, що виконується
Операції над числовими величинами	

+	Покомпонентне додавання числових масивів однакової розмірності; додавання скалярної величини до кожного елемента масиву
-	Покомпонентне віднімання числових масивів однакової розмірності; віднімання скалярної величини з кожного елемента масиву
*	Множення матриць відповідно до правил лінійної алгебри (число стовпців першого співмножника має дорівнювати числу рядків другого співмножника); множення всіх компонентів масиву на скаляр
.*	Покомпонентне множення елементів масиву однакової розмірності
/	Ділення скаляра на скаляр; покомпонентне ділення всіх елементів масиву на скаляр
./	Покомпонентне ділення елементів масивів однакової розмірності
\	Ліве матричне ділення
.\	Покомпонентне ділення елементів (ліве поелементне ділення)
^	Зведення скаляра в будь-яку ступінь; обчислення цілого ступеня квадратної матриці
'	Обчислення сполученої матриці
.'	Транспонування матриці

### Використання елементарних функцій

Задано вираз: 
$$e^{-2.5} (\ln 11.3)^{0.3} - \sqrt{\frac{\sin 2.45\pi + \cos 3.78\pi}{\operatorname{tg} 3.3}}$$

Щоб обчислити результат необхідно в командному рядку ввести цей вираз відповідно до правил MatLab і натиснути клавішу *Enter*. При введенні виразу використовуються вбудовані функції MatLab: експонента, натуральний логарифм, квадратний корінь, тригонометричні функції. Аргументи функцій беруться в круглі дужки, імена функцій набираються малими літерами. Для введення числа  $\pi$  досить набрати *pi* в командному рядку. Арифметичні операції в MatLab виконуються в звичайному порядку, властивому більшості мов програмування:

- зведення в ступінь ^ ;
- множення і ділення \*, / ;
- додавання і віднімання +, - .

Для зміни порядку виконання математичних операторів слід використовувати круглі дужки.

Відповідно до правил MATLAB введемо в командному рядку заданий вираз:

```
>> clear
>> exp(-2.5)*log(11.3)^0.3-sqrt((sin(2.45*pi)+cos(3.78*pi))/tan(3.3))
```



ans =  
-3.2105

### Вбудовані елементарні функції

Вбудовані елементарні функції MatLab включають тригонометричні, гіперболічні, експоненціальні і логарифмічні функції, а також функції для роботи з комплексними числами і для округлення різними способами.

Основні математичні функції MatLab:

sqrt(x)	обчислення квадратного кореня
exp(x)	піднесення до ступеня числа e
pow2(x)	піднесення до ступеня числа 2
log(x)	обчислення натурального логарифма
log10(x)	обчислення десяткового логарифма
log2(x)	обчислення логарифма за основою 2
sin(x)	синус кута x, заданого в радіанах
cos(x)	косинус кута x, заданого в радіанах
tan(x)	тангенс кута x, заданого в радіанах
cot(x)	котангенс кута x, заданого в радіанах
asin(x)	арксинус
acos(x)	арккосинус
atan(x)	арктангенс
acot(x)	арккотангенс
pi	число pi
round(x)	округлення до найближчого цілого
fix(x)	усічення дробової частини числа
floor(x)	округлення до меншого цілого
ceil(x)	округлення до більшого цілого
mod(x)	залишок від ділення с урахуванням знаку
sign(x)	знак числа
factor(x)	розкладання числа на прості множники
isprime(x)	істинно, якщо число просте
rand	генерація псевдовипадкового числа з рівномірним законом розподілу
randn	генерація псевдовипадкового числа з рівномірним законом розподілу
abs(x)	обчислення модуля числа

### Приклади на округлення і залишок від ділення

- fix – округлення до найближчого цілого у напрямку до нуля:  

```
>> fix(1.8)           >> fix(-1.8)
ans =                 ans =
  1                   -1
```
- floor, ceil – округлення до найближчого цілого у напрямку до мінус нескінченності або плюс нескінченності:  

```
>> floor(3.2)        >> ceil(3.2)
ans =                 ans =
  3                   4
```
- round – округлення до найближчого цілого:  

```
>> round(4.1)        >> round(4.5)
ans =                 ans =
  4                   5
```
- mod – залишок від цілочисельного ділення (зі знаком другого аргументу):  

```
>> mod(5,2)          >> mod(5,-2)
ans =                 ans =
  1                   -1
```
- rem – залишок від цілочисельного ділення (зі знаком першого аргументу):  

```
>> rem(5,2)          >> rem(5,-2)
ans =                 ans =
  1                   1
```

## Основні об'єкти MatLab

### Поняття про математичний вираз

Основним поняттям всіх математичних систем є *математичний вираз*. Він задає те, що повинно бути обчислено в числовому, рідше символічному вигляді. Приклади простих математичних виразів:

```
3 * 5
1.215 * sin(x)
5 + exp(4) / 3
sqrt(y) / 2
sin(pi/4)
```

Математичні вирази будуються на основі чисел, констант, змінних, операторів, функцій і різних спеціальних знаків.

### Дійсні числа

*Число* являється найпростішим об'єктом мови MatLab, який представляє кількісні дані. Числа можна вважати константами, імена яких співпадають з їх значеннями. Числа використовуються в загальноприйнятому представленні про них. Вони можуть бути цілими, дробовими, з фіксованою і плаваючою комою. Можливо представлення чисел в науковому форматі з вказанням мантиси і порядку числа. Приклади представлення чисел:

```
0
```

2  
 -3  
 2.301  
 0.00001  
 123.456e-24  
 -234.456e10

Як прийнято в більшості мов програмування в мантисі чисел ціла частина відокремлюється від дробової не комою, а крапкою. Для відокремлення порядку числа від мантиси використовується символ *e*. Знак «плюс» у чисел не проставляється, а знак «мінус» у числа називають *унарним мінусом*. Пробіли між символами в числах не допускаються.

В MatLab не прийнято ділити числа на цілі і дробові, короткі і довгі і т. д., як прийнято в більшості мов програмування, хоч задавати числа в таких формах можна. Операції над числами в MatLab виконуються у форматі, який прийнято вважати форматом з подвійною точністю.

### Константи і системні змінні

*Константа* це попередньо визначене числове або символічне значення, представлене унікальним іменем. Наприклад, числа 1, -2, 1.23 являються безіменними числовими константами. Інші види констант в MatLab прийнято називати системними змінними, оскільки, з одного боку, вони задаються системою при її завантаженні, а з іншого можуть перевизначатися. Основні системні змінні, які застосовуються в системі MatLab:

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

### Використання змінних. Перенесення рядку в сесії

Як і в усіх мовах програмування, в MatLab передбачена можливість роботи зі *змінними*. Користувач не повинен піклуватися про те, які значення буде приймати змінна (комплексні, дійсні або тільки цілі). Для того, щоб присвоїти, наприклад, змінній *z* значення 1.45, достатньо написати в командному рядку *z = 1.45*. MatLab відразу виведе значення *z*:

```
>> z = 1.45
z =
    1.4500
```

Тут знак рівності використовується в якості *оператора присвоювання*. Часто не дуже зручно після кожного присвоювання отримувати ще і результат.

Тому в MatLab передбачена можливість завершати оператор присвоювання крапкою з комою для відміни виведення результату в командне вікно. Ім'ям змінної може бути будь яка послідовність букв і цифр без пробілу, яка починається з букви. Великі та малі літери відрізняються. Наприклад Mz і mZ являються двома різними змінними. Кількість символів в імені змінної, які сприймає MatLab складає 63, але такі довгі імена рідко бувають потрібні. Наприклад, необхідно знайти значення виразу:

$$\frac{\sin 1.3\pi}{\ln 3.4} + \sqrt{\frac{\operatorname{tg} 2.75}{\operatorname{tg} 2.75}}$$

$$\frac{\sin 1.3\pi}{\ln 3.4} - \sqrt{\frac{\operatorname{tg} 2.75}{\operatorname{tg} 2.75}}$$

Наберемо пошагово послідовність команд. Проміжні обчислення закінчуються крапкою з комою для подавлення виводу обчислень. Останній оператор присвоєння не завершується крапкою з комою для того, щоб отримати результат на екрані.

```
>> x=sin(1.3*pi)/log(3.4);
>> y=sqrt(tan(2.75)/tan(2.75));
>> z=(x+y)/(x-y)
```

```
z =
-0.2040
```

Можна було б ввести всю формулу і отримати той же результат.

```
>> (sin(1.3*pi)/log(3.4)+sqrt(tan(2.75)/tan(2.75)))/sin(1.3*pi)/...
log(3.4)-sqrt(tan(2.75)/tan(2.75))
ans =
-0.2040
```

Розглянутий приклад показує, що в деяких випадках математичний вираз, який вводиться, може виявитися настільки довгим, що для нього не хватить одного рядка. Тоді частину виразу можливо перенести на новий рядок за допомогою знаку крапка «...» (три або більше крапок).

Максимальне число символів в одному рядку командного режиму – 4096, а в m-файлі – не обмежено. У ранніх версіях в одному рядку було не більше 256 символів.

Таким чином, для введення довгих формул або команд в командний рядок необхідно поставити три крапки (підряд, без пробілів), натиснути клавішу Enter і продовжити набір формули на наступному рядку. Так можна розмістити вираз на декількох рядках. MatLab вираховує весь вираз після натискання на Enter в останньому рядку (в якому немає трьох точок, які ідуть підряд). MATLAB запам'ятовує значення всіх змінних, визначених під час сеансу роботи.

### Текстові коментарі

Оскільки MatLab використовується для досить складних обчислень, важливе значення має здійснювати їх опис. Це досягається завдяки

використанню текстових коментарів. Текстові коментарі вводяться за допомогою символу % за яким далі слідує необхідний текст. Наприклад  
% обчислення вектору значень аргументу

### Змінні і привласнення їм значень

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

В системі MatLab є можливість задавати змінним певних значень. Для цього використовується операція присвоювання, що вводиться знаком рівності:

Ім'я\_змінної = Вираз

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

*Ім'я змінної* (її *ідентифікатор*) може містити скільки завгодно символів, але запам'ятовується і ідентифікується тільки 31 початковий символ. Ім'я будь-якої змінної не повинно збігатися з іменами інших змінних, функцій і процедур системи, тобто воно повинно бути унікальним. Ім'я повинно починатися з літери. Може містити літери, цифри і символ підкреслення (  ). Неприпустимо включати в імена змінних пробіли і спеціальні знаки, наприклад +, -, \*, /.

### Знищення визначень змінних

У пам'яті комп'ютера змінні займають певне місце, яке називається *робочою областю* (workspace). Для очищення робочої області використовується функція *clear* в різних формах:

- *clear* знищення визначень всіх змінних;
- *clear x* знищення визначення змінної *x*;
- *clear a, b, c* знищення визначень декількох змінних.

Знищена (стерта в робочій області) змінна стає невизначеною. Використовувати невизначені змінні не можна, і такі спроби будуть супроводжуватися видачою повідомлень про помилку.

### Оператори і функції

*Оператор* – це спеціальне визначення для певної операції над даними – *операндами*. Наприклад, найпростішими арифметичними операторами є знаки суми +, віднімання -, множення \* і ділення /. Оператори використовуються спільно з операндами. Так в вираженні  $2 + 3$  знак + є оператором складання, а числа 2 і 3 – операндами.

*Функції* – це об'єкти, які мають унікальні імена, і виконують певні перетворення своїх аргументів і при цьому повертають результати цих перетворень. Повернення результату – відмінна риса функцій. При цьому результат обчислення функції з одним вихідним параметром підставляється на місце її виклику. Це дозволяє використовувати функції в математичних виразах. Приклад використання функції *sin* в обчислюваному виразі  $5 * \sin(\pi/3)$ .

Якщо функція повертає кілька значень, то вона записується у вигляді:

$$[y_1, y_2, \dots] = \text{func}(x_1, x_2, \dots)$$

де  $y_1, y_2, \dots$  список вихідних параметрів і  $x_1, x_2, \dots$  список вхідних параметрів (аргументів).

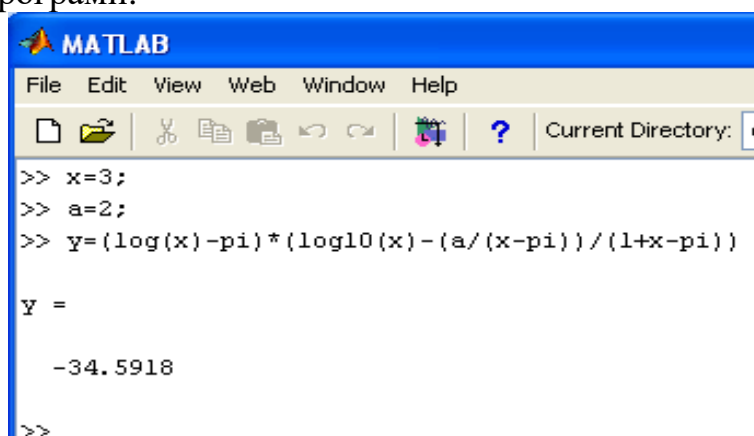
### Зразок виконання завдання

**Завдання 1.** Необхідно здійснити розв'язок лінійного обчислювального процесу і отримати в результаті значення  $Y$ .

$$Y = (\ln x - \pi) \frac{\lg x - \frac{a}{x - \pi}}{1 + x - \pi}$$

Для розв'язання необхідно задати значення вхідним змінним  $a$  та  $x$ . Потім обчислити значення  $y$  і вивести отриманий результат.

Розв'язання лінійних обчислювальних процесів можна виконувати в інтерактивному режимі. Для цього у вікні *Command Window* в рядку вводу набрати команди програми.



```

MATLAB
File Edit View Web Window Help
[Icons] ? Current Directory: C
>> x=3;
>> a=2;
>> y=(log(x)-pi)*(log10(x)-(a/(x-pi)))/(1+x-pi)

Y =

    -34.5918

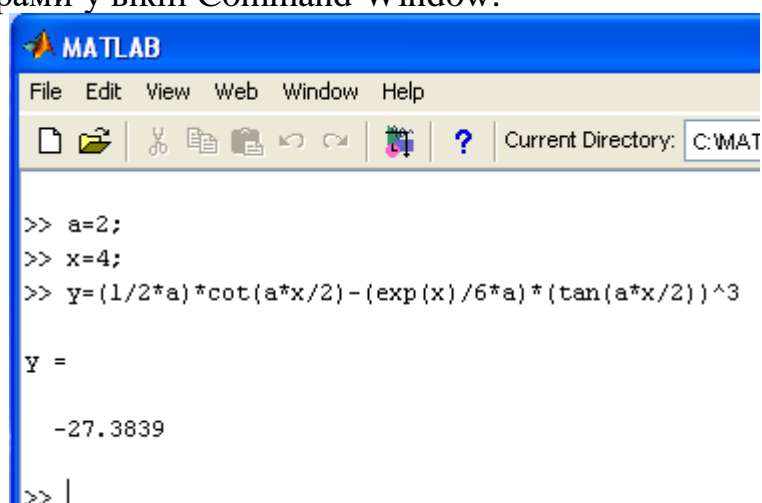
>>

```

**Завдання 2.** Задана умова:

$$Y = \frac{1}{2a} \cdot c \operatorname{tg} \frac{ax}{2} - \frac{e^x}{6a} \cdot \operatorname{tg}^3 \frac{ax}{2}$$

Лістинг програми у вікні *Command Window*:



```

MATLAB
File Edit View Web Window Help
[Icons] ? Current Directory: C:\MAT
>> a=2;
>> x=4;
>> y=(1/2*a)*cot(a*x/2)-(exp(x)/6*a)*(tan(a*x/2))^3

Y =

    -27.3839

>> |

```

Для розв'язання лінійних обчислювальних процесів спочатку введені вхідні значення змінних  $a$  та  $x$ , далі введено вираз для обчислення невідомої змінної  $y$  і отримано результат.

## Індивідуальні завдання за темою «Лінійний обчислювальний процес»

### Варіанти індивідуальних завдань

Обчислити значення змінних, зазначених в задачах. Значення вхідних даних задати самостійно. Вивести результат в різних форматах і зберегти отримані значення у файлі.

1.  $H = \frac{(\cos^2 \alpha + 0,7 \lg i^2)^3}{\pi} \sqrt{i}$
2.  $H = \frac{1}{2\sqrt{2} \cdot a} \cdot \left( \frac{1 - 3\cos^2 ax}{1 + \cos^2 ax} \right)$
3.  $Y = \frac{2}{b\sqrt{b^2 + c^2}} \cdot \left( \frac{(b^2 + c^2) \cdot \operatorname{tg}(bx/2)}{\sqrt{b^2 + c^2}} \right)$
4.  $Y = -\frac{2}{\pi \cdot a} \cdot \ln \left| \frac{a + \sqrt{\pi^2 + x^2}}{a - \sqrt{\pi^2 + x^2}} \right|$
5.  $Y = \frac{\sin 2\alpha + \cos^2 \alpha}{\lg(a^2 + 2)} e^a$
6.  $Y = \frac{\pi + (ax + b)^2}{a - (\pi x + b)^2} \cdot \frac{\sqrt[5]{(\pi x + b)^2}}{\sqrt[3]{(ax + b)^2}}$
7.  $Y = [\sin^2(\pi x) - 2]^{2x} - \operatorname{tg}[\sin^2(\pi x) - 2]^{4x}$
8.  $Y = \sqrt[3]{2 \cdot \ln(\pi - \sin^2 x)^2} - \sqrt[3]{\lg(\pi - \sin^2 x)^4}$
9.  $Y = \lg \left| 4 - \cos^2 \frac{x}{4} \right| - 2 \ln \left( 4 - \cos^2 \frac{x}{4} \right)^2$
10.  $Y = \frac{e^{2x} + e^{0,57x}}{\lg a^2} + \sin^2 x$
11.  $D = \frac{ae^b}{b(\sin \alpha + \operatorname{arctg} b)} + \ln a$
12.  $V = \frac{2t}{a \sin \beta(t + a)} \operatorname{ctg} \beta - \ln a$
13.  $S = \frac{k(x+i)^{\frac{\pi}{2}}}{2x} + \ln^2 k$
14.  $P = \frac{\ln(k + 20)}{\lg(a + \operatorname{ctg} k)} \sin \beta$

$$15. C = \frac{a \cdot \operatorname{ctgn}}{1 + \operatorname{lg}(i \cdot a)} + e^i$$

$$16. Y = \frac{\sqrt{\operatorname{lg}(x^4 + a^2)}}{abe^{xa}}$$

$$17. Y = \frac{\sin^4 x + \operatorname{ctg}^2 a - \ln x}{e^{xa}}$$

$$18. S = \frac{\operatorname{lg}(x^2 + b^4 + 1)}{\sqrt{2x^2 + \cos^2 b}}$$

$$19. Z = \frac{\sqrt[3]{x^2 \cos^2 \alpha}}{13,2 \operatorname{lg}(x^2 + 1,1)}$$

$$20. A = \frac{\cos k(2,05i + i^3)^{\frac{\pi}{2}}}{4,5(k^2 + 1)}$$

### Програмування розгалужень

#### Логічні вирази і вирази відносин

При виконанні розгалужуючих обчислювальних процесів використовуються умови. Умова може містити логічний вираз, складений з операцій відносин і логічних операцій або операторів. Для задання умови допустимі операції відносин, наведені в таблиці.

#### Операції відносин

Позначення	Операція відносин
==	Рівність
<	Менше
<=	Менше або дорівнює
>=	Більше чи рівно
~=	Не дорівнює

Задання більш складних умов проводиться із застосуванням операторів або операцій. Наприклад,  $-1 \leq x < 2$  складається в одночасному виконанні нерівності  $x \geq -1$  і  $x < 2$  і записується за допомогою логічного оператора and

and ( $x \geq -1, x < 2$ )

або еквівалентним чином із застосуванням логічної операції «і» &

( $x \geq -1$ ) & ( $x < 2$ )

Основні логічні операції і оператори і приклади їх запису наведені в таблиці.

#### Логічні вирази

Тип виразу	Вираз	Логічний оператор	Логічна операція
Логічне «і»	$x < 3$ і $k = 4$	and ( $x < 3, k = 4$ )	( $x < 3$ ) & ( $k = 4$ )
Логічне «або»	$x = 1$ або $x = 2$	or ( $x = 1, x = 2$ )	( $x = 1$ )   ( $x = 2$ )
Заперечення «не»	$a \neq 1.9$	not ( $a = 1.9$ )	~ ( $a = 1.9$ )



Оператори *not*, *end* і *or* є функціями, які повертають значення «істина» (логічна одиниця) або «брехня» (логічний нуль), такі ж значення приймає будь-який логічний вираз.

Для задання порядку виконання логічних операцій слід використовувати круглі дужки.

При програмуванні алгоритмів часто потрібно виконати той чи інший блок команд в залежності від деяких умов, тобто використовувати розгалуження алгоритму.

### Оператори розгалуження

Умовний оператор *if* і оператор перемикання *switch* дозволяють створити гнучкий алгоритм, що розгалужується, в якому при виконанні певних умов виконується відповідний блок операторів або команд MatLab.

#### Умовний оператор *if*

Оператор *if* може застосовуватися в простому вигляді, для виконання блоку команд при задоволенні деякої умови, або в конструкції *if-elseif-else* для написання алгоритмів, що розгалужуються.

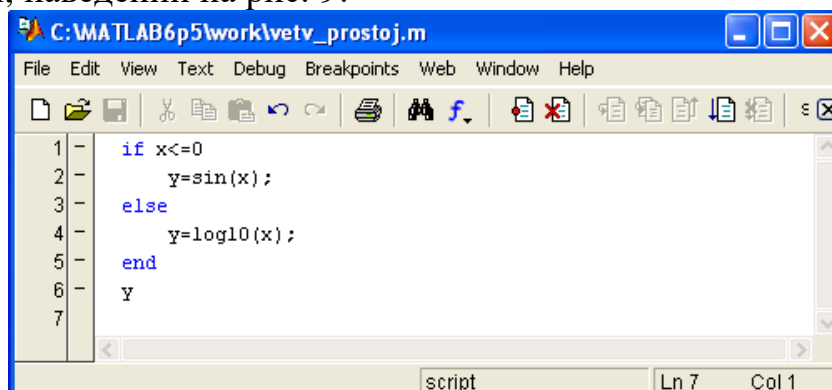
Синтаксис умовного оператора виглядає так:

```
if умова
    команди MatLab
end
```

Якщо умова вірна, то виконуються команди MatLab, розміщені між *if* і *end*, а якщо умова невірна, то відбувається перехід до команд, розташованих після *end*.

#### Організація розгалуження

У загальному вигляді оператор розгалуження представляє конструкцію *if-elseif-else*. Розглянемо на прикладі лістинг програми, яка демонструє роботу *if-else*. Залежно від виконання тієї чи іншої умови працює відповідна гілка програми. Якщо всі умови невірні, то виконуються команди, розміщені після *else*. Гілок може бути скільки завгодно або тільки дві. Розглянемо приклад розгалуження, наведений на рис. 9:

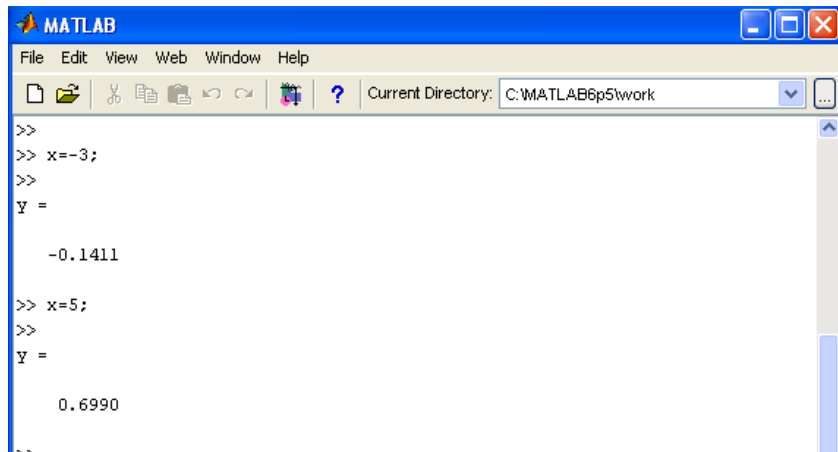


```
C:\MATLAB6p5\work\lvetv_prostoj.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 - if x<=0
2 -     y=sin(x);
3 - else
4 -     y=log10(x);
5 - end
6 - y
7
```

Рис. 9. Розгалужений обчислювальний процес

Наведений приклад має дві гілки розгалуження, тому для перевірки правильності написання програми необхідно почергово задавати змінній *x*

значення, які відповідають умовам, заданим у прикладі. Результат розв'язання даного прикладу представлено на рисунку 10.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>>
>> x=-3;
>>
Y =
    -0.1411

>> x=5;
>>
Y =
    0.6990

>>>

```

Рис. 10. Результат розв'язання прикладу, отриманий при значенні  $x=-3$  та при значенні  $x=5$

У разі наявності двох гілок використовується завершальне *else*, а *elseif* пропускається. Оператор *if* повинен закінчуватися *end*. При наявності у прикладі трьох гілок, використовується конструкція *if-elseif-else*.

Розглянемо приклад  $\neg$  обчислення кусочно-заданої функції. Система рівнянь складається із трьох гілок.

$$y(x) = \begin{cases} \sin x - 1, & x < -\pi \\ x/\pi, & -\pi \leq x < \pi \\ -\cos x, & x \geq \pi \end{cases}$$

Наберемо текст файл-програми в редакторі М-файлів (рис.11). Для цього треба розкрити меню File робочого середовища MatLab і в пункті New вибрати підпункт M-file або натиснути кнопку New M-file на панелі інструментів робочого середовища.

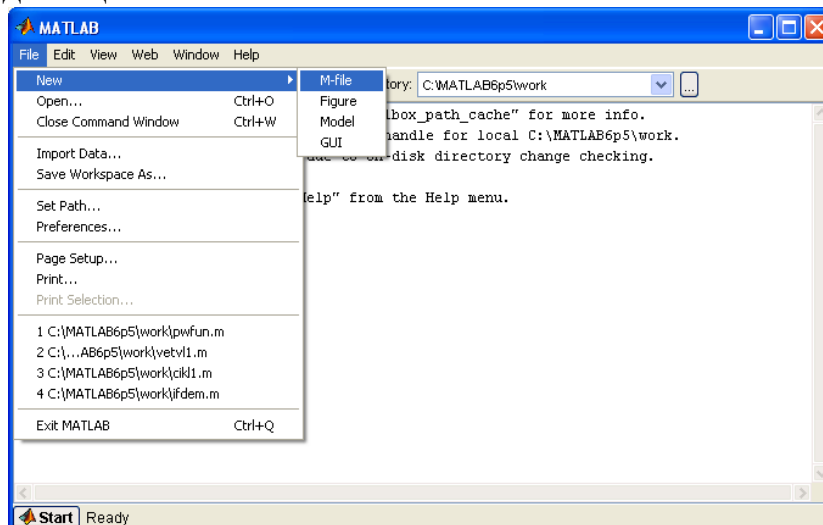


Рис. 11. Команди створення тексту файл-програми в редакторі М-файлів

Новий файл відкривається у вікні редактора М-файлів, яке наведено на рис. 12.

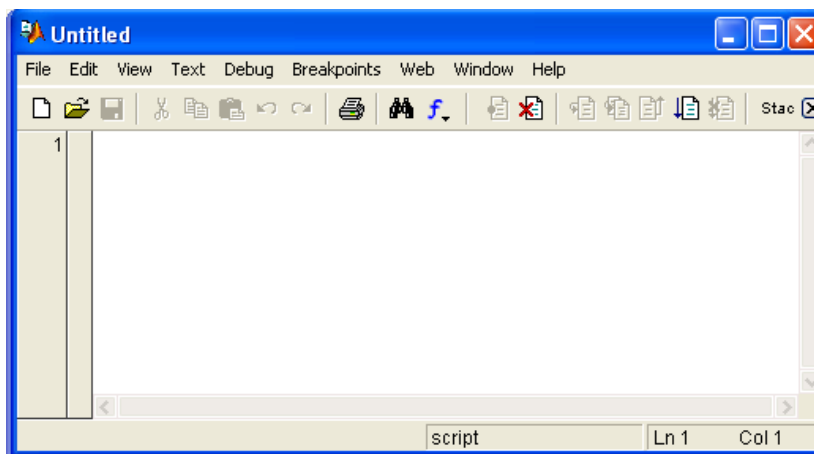


Рис. 12. Вікно редактора М-файлів MatLab

Вид рядка меню і панелі інструментів залежить від ширини вікна. Якщо воно досить вузьке, то частина інструментів переміщується в список, що розкривається. Наберемо в редакторі команди програми. Лістинг файл-функції для обчислення кусочно-заданої функції представлений на рисунку 13.

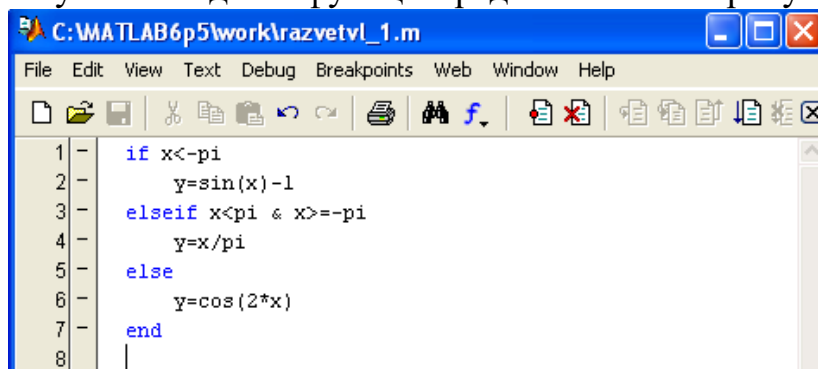


Рис. 13. Лістинг файл-функції

Збережемо файл вибравши в меню File редактора пункт Save as (рис. 14).

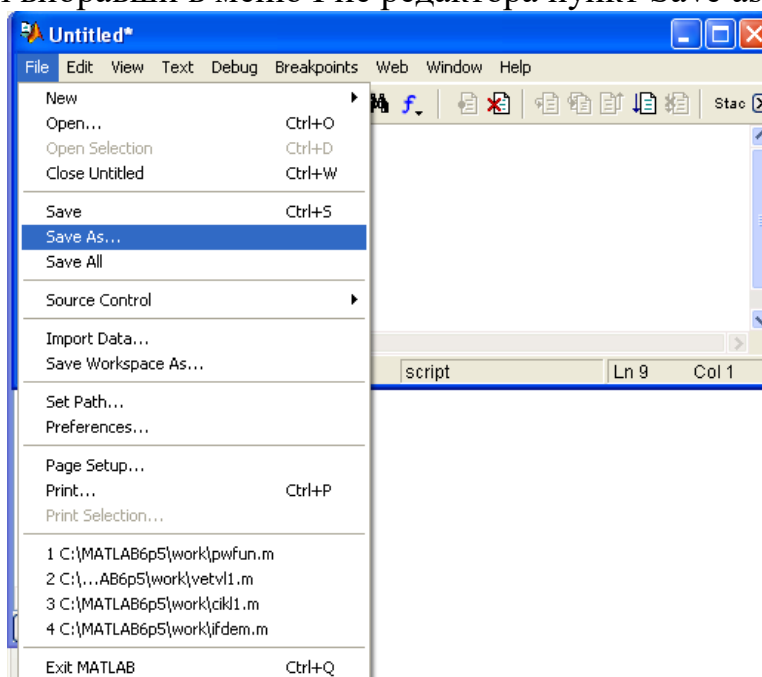


Рис. 14. Вікно збереження файлу

На екрані з'явиться вікно для збереження файлу, в якому в рядку Файл необхідно задати ім'я, наприклад razvetvl\_1 (рис.15).

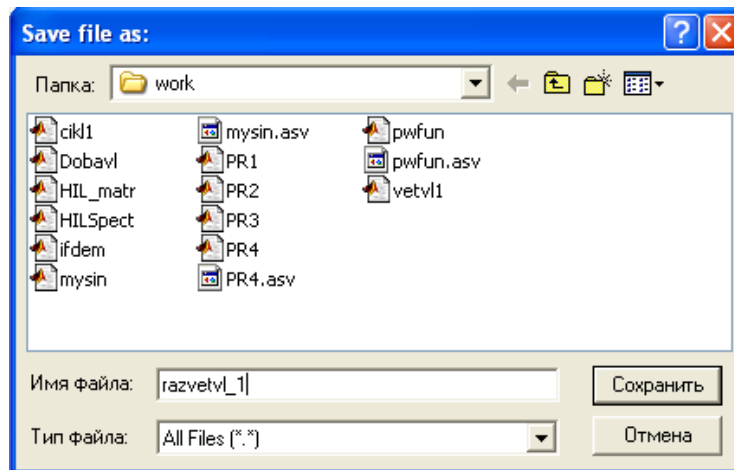
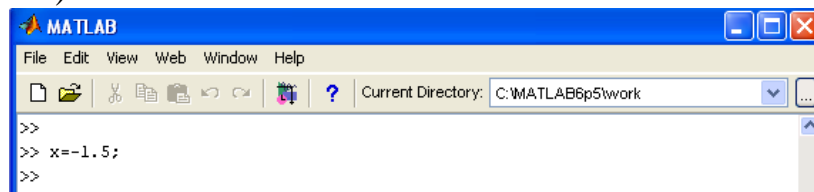


Рис.15. Вікно для збереження файлу

Для запуску на виконання команд файлу слід вибрати пункт Run в меню Debug або просто натиснути F5. Але перед запуском програми на виконання необхідно задати значення вхідній змінній  $x$ . Згідно умови система заданих рівнянь може мати три розв'язки, а змінна  $x$  відповідно може приймати три різні значення.

Розв'язання системи відбудеться по першій гілці, якщо значення  $x < -\pi$ , тобто  $x < -3,14$ . Наприклад  $x = -1.5$ . У вікні *Command Window* задаємо значення змінній  $x$  (Рис. 16).

Рис. 16. Введення значення вхідної змінної  $x$ 

Потім у вікні редактора М-файлів задаємо команду на виконання програми (Рис. 17).

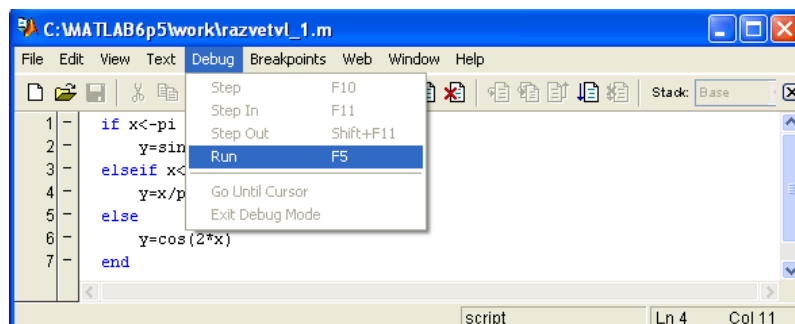
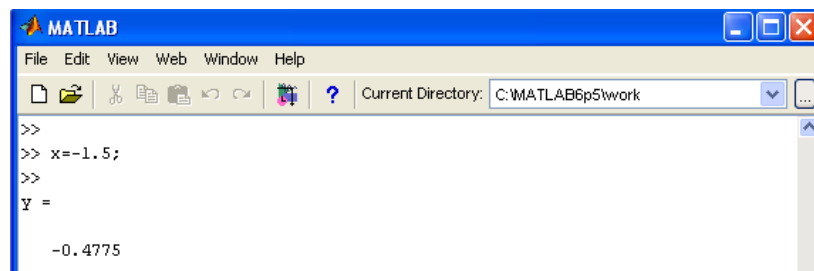


Рис. 17. Запуск на виконання команд файлу

У вікні *Command Window* з'явиться результат виконання програми (Рис. 18.).

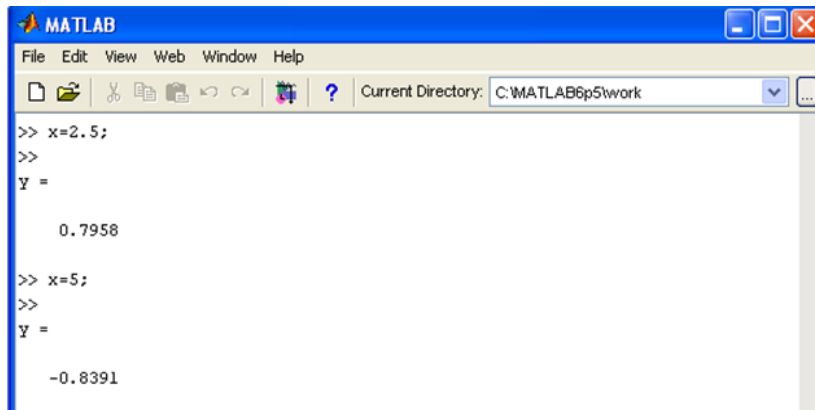
Рис 18. Вікно *Command Window* з результатом виконання програми

Розв'язання системи рівнянь виконано по першій гілці, тобто перевірено роботу першої умови системи рівнянь.

Розв'язання системи по другій гілці відбудеться, якщо значення  $x$  знаходиться в діапазоні  $x < p_1$  та  $x \geq -p_1$ , тобто значення  $x$  повинно бути менше ніж 3,14 але більше ніж -3,14. Наприклад  $x = 2.5$ .

Розв'язання системи по третій гілці відбудеться, якщо значення  $x > p_1$ , тобто  $x > 3,14$ . Наприклад  $x = 4$ .

Результат виконання програми `razvetv1_1` по другій і третій гілці програми представлено у вікні *Command Window* (Рис. 19).



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>> x=2.5;
>>
Y =
    0.7958
>> x=5;
>>
Y =
   -0.8391
  
```

Рис. 19. Вікно *Command Window* з результатом виконання програми

Розв'язання даної системи рівнянь виконано в редакторі М-файлів.

Рішення розгалужених обчислювальних процесів можна також виконувати в інтерактивному режимі. Для цього у вікні *Command Window* в рядку введення набрати рядок за рядком команди програми. Наприклад, задана система рівнянь. Необхідно обчислити значення  $y$ .

$$Y = \begin{cases} x \sin^2 x, & \text{якщо } x \leq 2 \\ \frac{x^e + \operatorname{ctg} x}{\sqrt{x}}, & \text{якщо } x \geq 6 \\ 0 - \text{в інших випадках} \end{cases}$$

Задана система рівнянь складається із трьох гілок, тобто система має три умови. Вибір гілки по якій відбувається розв'язання залежить від значення вхідної змінної. У вікні *Command Window* спочатку необхідно ввести значення вхідної змінної, потім рядок за рядком необхідно ввести команди програми. Після вводу останньої команди програми на екрані з'являється результат. Для перевірки правильності роботи програми по іншим гілкам системи необхідно ввести інше значення вхідної змінної і повторити дії.

Для перевірки роботи першої гілки програми заданої системи рівнянь треба здійснити рішення задавши значення  $x \leq 2$ , наприклад  $x = 2$  і отримати результат. Щоб перевірити правильність роботи програми по другій гілці розгалуження треба задати значення  $x \geq 6$ , наприклад  $x = 7$ , повторити команди програми і отримати результат. Для перевірки роботи програми по третій гілці розгалуження необхідно задати значення  $x$ , яке виключає значення, які задані в першій та другій умовах системи. Тобто  $x$  може приймати будь яке значення з інтервалу від 2 до 6. Таким чином значення  $x$  повинно бути більше 2

та менше 6, наприклад  $x = 4$ . Після вводу значення вхідної змінної  $x$  знову необхідно повторити команди програми і отримати результат.

Рішення даної системи рівнянь представлено на рисунку 20.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>> x=2;
>> if x<=2
y=x*sin(x)^2
elseif x>=6
y=(x^exp(1)+cot(x))/sqrt(x)
else
y=0
end

Y =

    1.6536

>> x=7;
>> if x<=2
y=x*sin(x)^2
elseif x>=6
y=(x^exp(1)+cot(x))/sqrt(x)
else
y=0
end

Y =

    75.3654

>> x=4;
>> if x<=2
y=x*sin(x)^2
elseif x>=6
y=(x^exp(1)+cot(x))/sqrt(x)
else
y=0
end

Y =

     0

>>

```

Рис. 20. Лістинг програми рішення системи рівнянь

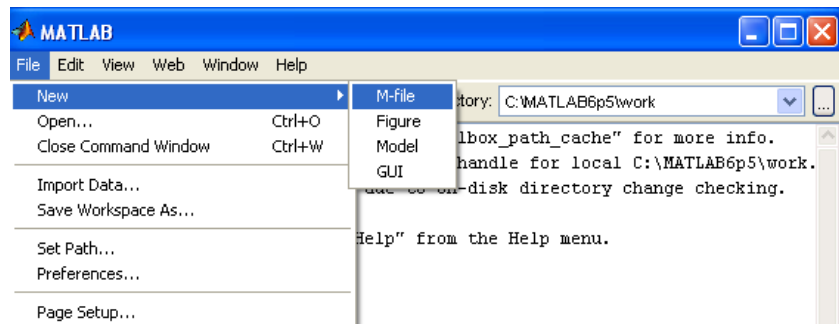
Як видно з прикладів, набір команд програми можна виконувати в редакторі М-файлів, а можна також виконувати в інтерактивному режимі.

### Зразок виконання завдань

**Завдання 1.** Задана система рівнянь:

$$Y = \begin{cases} x\sqrt{b^3} & \text{при } 0 < x < 4 \text{ и } b > 0 \\ b \sin^2 x & \text{при } x \geq 4 \text{ и } b = 1 \\ 0 & \text{в остальных случаях} \end{cases}$$

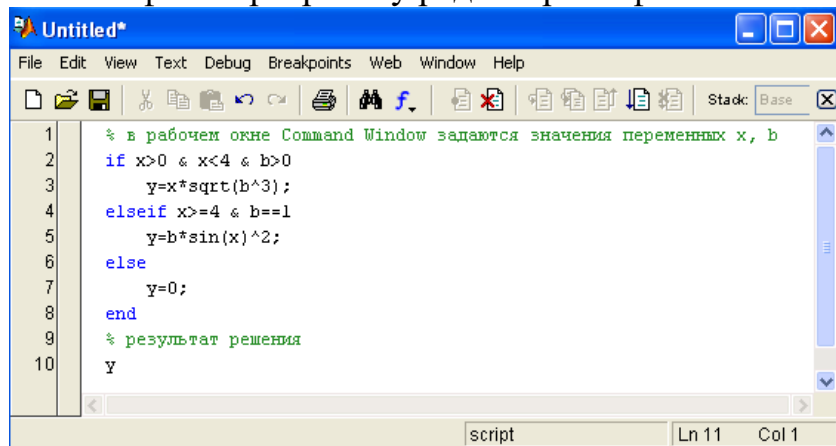
Для створення програми необхідно розкрити меню File робочого середовища MatLab, і у пункті New вибрати підпункт M-file.



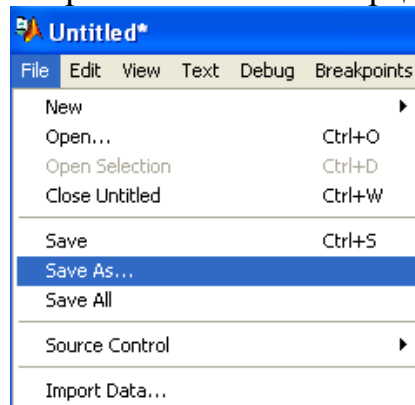
Після цього розкриється вікно редактора М-файлів Untitled, в якому і виконується введення програми.



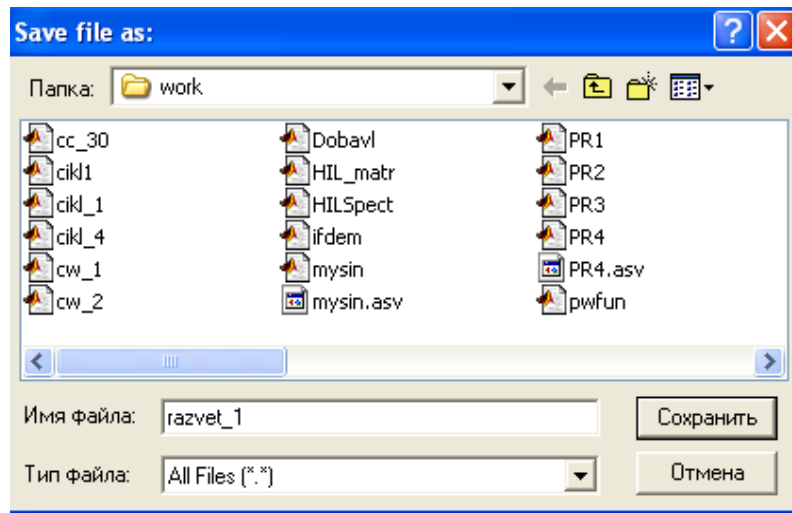
Наберемо текст файл-програми у редакторі М-файлів.



Збережемо програму вибравши в меню File редактора пункт Save as

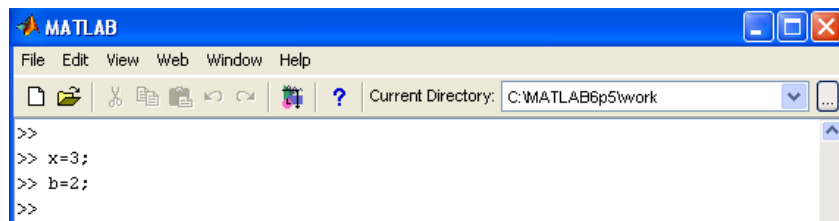


На екрані з'явиться вікно для збереження файлу, в якому в рядку «Імя файла» необхідно задати ім'я файлу, наприклад razvet\_1.

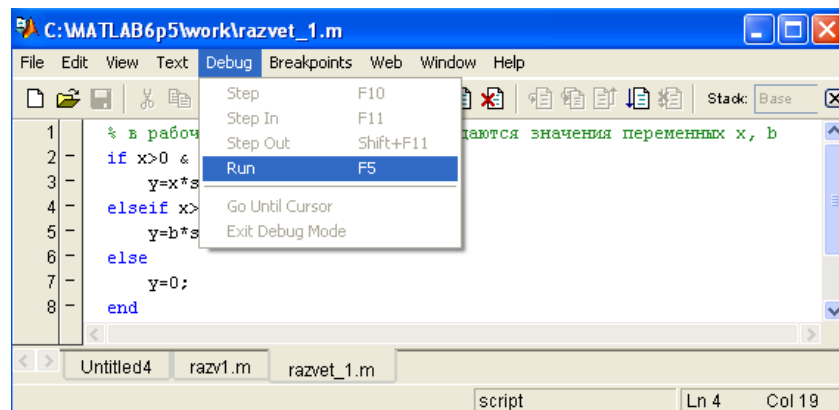


Для того, щоб виконати перевірку роботи програми, треба здійснити рішення програми три рази, так як згідно умові задачі існує три гілки розгалуження. Тобто, при різних значеннях вхідних даних розв'язання повинно виконуватись по різному, в залежності від введеного значення вхідної змінної. В робочому вікні *Command Window* пакету MatLab необхідно задати значення змінних, які потрібні для розв'язання. В даному прикладі це змінні  $b$  та  $x$ . В залежності від значення цих змінних, буде виконано обчислення по одній з трьох гілок прикладу. При значенні змінної  $b > 0$ , і при значенні змінної  $x > 0$  та  $x < 4$  обчислення виконується по першій гілці. Рішення по другій гілці виконується при значеннях  $b = 1$  і  $x \geq 4$ . По третій гілці рішення буде виконуватися при значеннях змінних  $b < 0$  і  $x = 0$ .

Введемо дані для рішення по першій гілці в робочому вікні *Command Window*.

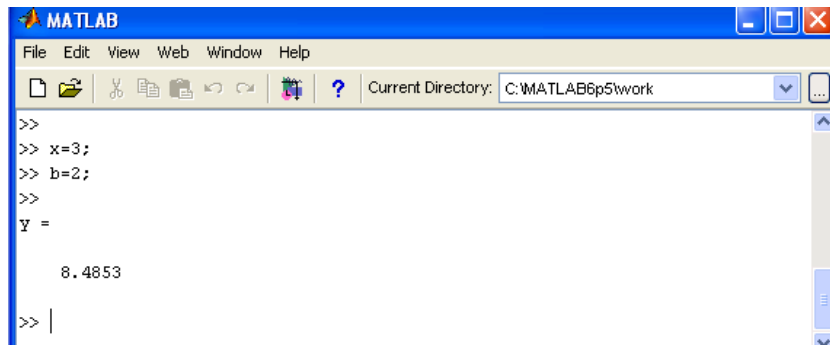


Для запуску на виконання команд файлу слід перейти у вікно редактора з текстом програми, у ньому вибрати пункт Run меню Debug або натиснути клавішу F5:



У результаті виконання програми у робочому вікні виводиться результат, отриманий при розв'язанні програми, тобто при  $x = 3$  і  $b = 2$  отримуємо:





```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
>> x=3;
>> b=2;
>>
Y =
    8.4853
>> |

```

Необхідно виконати рішення три рази, присвоюючи даним значення, при яких рішення виконується по кожній із гілок системи, тобто для другої гілки системи значення змінних  $x = 7$  та  $b=1$ .



```

>> x=7;
>> b=1;
>>
Y =
    3

```

При рішенні системи рівнянь по третій гілці потрібно задати значення змінних, виключаючи дві перших умови, тобто задати вхідним змінним такі значення  $x = -3$  і  $b = -2$ .



```

>> x=-3;
>> b=-2;
>>
Y =
    0
>>

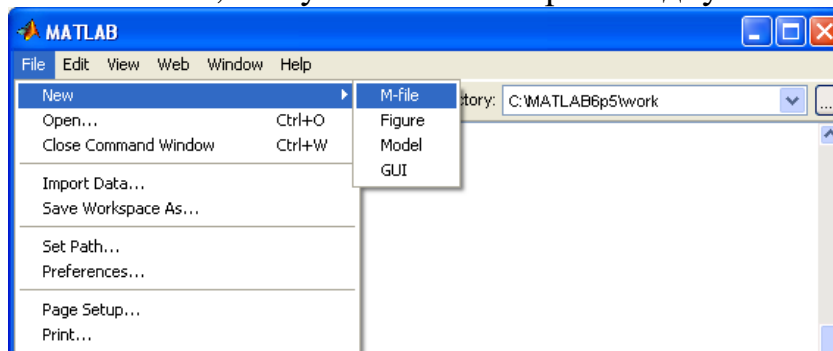
```

При перевірці роботи програми отримані результати, які доводять, що відповідно до значення вхідних змінних відбувається розв'язання системи рівнянь по відповідній гілці, тобто по гілці, яку задовольняє введене значення.

**Завдання 2.** Задано систему рівнянь:

$$F = \begin{cases} \frac{b^x + a^x}{x^2} \lg ax & \text{при } 0,5 < x \leq 2, \quad a > 0 \\ \sqrt[3]{a^2 \cos^2(bx)} & \text{при } x > 2,5, \quad a^2 \neq x \\ a^2 + x & \\ 0 & \text{в остальных случаях} \end{cases}$$

Розкриємо меню File, і в пункті New виберемо підпункт M-file:



```

MATLAB
File Edit View Web Window Help
File
  New
  Open... Ctrl+O
  Close Command Window Ctrl+W
  Import Data...
  Save Workspace As...
  Set Path...
  Preferences...
  Page Setup...
  Print...
  M-file
  Figure
  Model
  GUI
  Directory: C:\MATLAB6p5\work

```

Наберемо текст файл-програми у редакторі M-файлів:

```

1  % в рабочем окне Command Window задаются значения переменных a, b, x
2  if x>0.5 & x<=2 & a>0
3      y=(b^x+a^x)/x^2*log10(a*x);
4  elseif x>2.5 & a^2 ~ x
5      y=((a^2*cos(b*x)^2)^(1/3))/(a^2+x);
6  else
7      y=0;
8  end
9  % результат решения
10 y

```

Збережемо програму вибравши в меню File редактору пункт Save as та задаємо ім'я файлу, наприклад `razvet_2`.

Далі для того, щоб перевірити роботу програми, у робочому вікні *Command Window* потрібно задати значення змінних, які необхідні для рішення. Задаємо значення змінним  $b = 3$ ,  $a = 1.5$ ,  $x = 2$ , які відповідають першій умові системи рівнянь.

```

>> b=3;
>> a=1.5;
>> x=2;
>> |

```

Запустимо програму на рішення:

```

1  % в рабоч
2  if x>0.5
3      y=(b
4  elseif x>
5      y=((a
6  else
7      y=0;
8  end
9  % результат решения
10 y
11 |

```

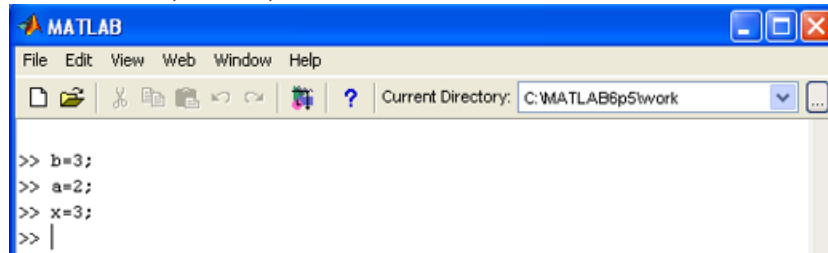
В робочому вікні *Command Window* отримаємо результат:

```

>> b=3;
>> a=1.5;
>> x=2;
>>
y =
    1.3419

```

Результат отримано по першій умові системи. Необхідно перевірити рішення усіх гілок системи. Для перевірки другої гілки системи потрібно задати наступні значення:  $b = 3$ ,  $a = 2$ ,  $x = 3$ .

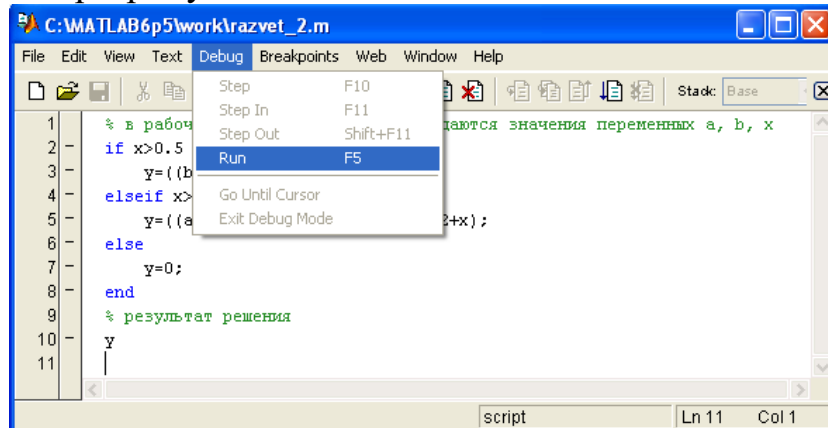


```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>> b=3;
>> a=2;
>> x=3;
>> |
  
```

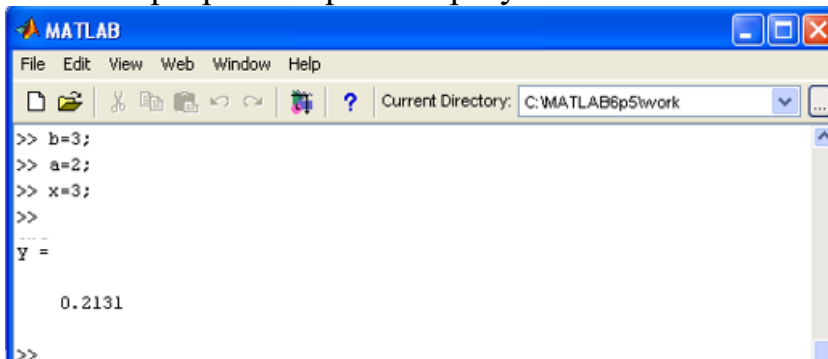
Потім запустити програму на виконання.



```

C:\MATLAB6p5\work\vrazvet_2.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
1  % в робоч
2
3  if x>0.5
4      y=(b
5  elseif x>
6      y=(a
7  else
8      y=0;
9  end
10 % результат рішення
11 y
script Ln 11 Col 1
  
```

В результаті рішення програми отримати результат.

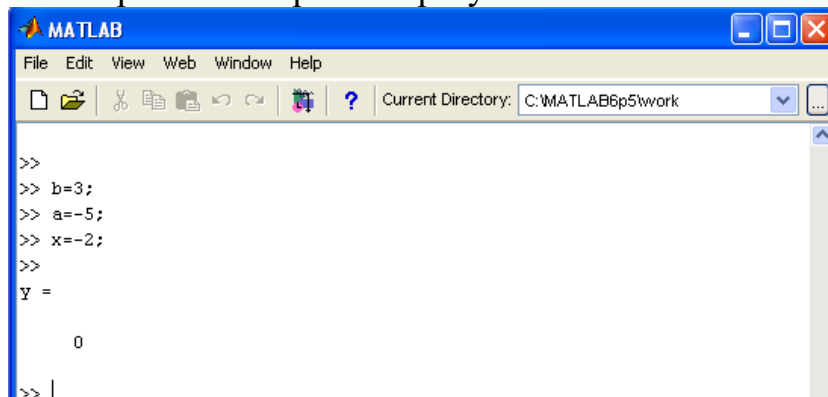


```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>> b=3;
>> a=2;
>> x=3;
>>
>>
y =
    0.2131
>>
  
```

Щоб перевірити третю гілку системи треба задати значення:  $b = 3$ ,  $a = -5$ ,  $x = -2$ , та виконавши рішення отримати результат.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>>
>> b=3;
>> a=-5;
>> x=-2;
>>
>>
y =
    0
>> |
  
```

Таким чином, здійснивши розв'язання програми тричі з різними значеннями вхідних змінних, і отримавши результати можна зробити висновок, що програма складена вірно. В залежності від значення вхідної змінної розв'язання відбувається згідно умови по відповідній гілці системи рівнянь.

## Індивідуальні завдання за темою «Умовний оператор if»

### Варіанти індивідуальних завдань

$$1. Y = \begin{cases} ax + \sqrt{b}, & \text{якщо } x \leq 2 \\ \frac{1}{x} \lg x, & \text{якщо } 2 < x \leq 6 \\ 10 - & \text{в інших випадках} \end{cases}$$

$$2. Y = \begin{cases} n \cdot (x^2 + e^x), & \text{якщо } x \leq 2 \\ \frac{\ln^2 x}{\lg(\sin(x^2 + 4))}, & \text{якщо } x \geq 6 \\ 7 - & \text{в інших випадках} \end{cases}$$

$$3. Y = \begin{cases} \sqrt[n]{ax^2}, & \text{якщо } x \leq 32, a = 5 \\ \frac{|\ln| + e^n}{x}, & \text{якщо } x \geq 46 \\ n - & \text{в інших випадках} \end{cases}$$

$$4. Y = \begin{cases} \sqrt[n]{x^e}, & \text{якщо } x \leq 2 < n \\ \frac{\ln n}{e^x + x}, & \text{якщо } x \geq 6 \\ \pi - & \text{в інших випадках} \end{cases}$$

$$5. Y = \begin{cases} \frac{\cos^2 x + a \lg(x^2)}{\pi} \sqrt{ax} & \text{якщо } x > 0, a > 0 \\ 10 + e & \text{якщо } x = 0, a = 0 \\ a \sin x + a^2 x & \text{в інших випадках} \end{cases}$$

$$6. F = \begin{cases} \ln^2(x+2) & \text{якщо } x > 0 \\ \frac{y}{2} + \sin^3 z & \text{якщо } 1 < z < 2, y > 0 \\ e + \pi & \text{в інших випадках} \end{cases}$$

$$7. Y = \begin{cases} 10\sqrt{ae^a}, & \text{якщо } 1 \leq a \leq 4 \\ \frac{\lg a}{a+10}, & \text{якщо } a \geq 6 \\ 10^\pi - & \text{в інших випадках} \end{cases}$$

$$8. Y = \begin{cases} ax^2 + \lg^3 N, & \text{якщо } N = 10 \\ e^N \cdot \ln x, & \text{якщо } x \geq 6, N = 5 \\ Na - x - & \text{в інших випадках} \end{cases}$$

$$9. Y = \begin{cases} n \cdot (x^2 + e^x), & \text{якщо } x \leq 2 \\ \frac{\ln^3 x}{\lg(\sin(x^2 + 4))}, & \text{якщо } x \geq 6 \\ 0 - & \text{в інших випадках} \end{cases}$$

10.  $Y = \begin{cases} \lg x^2 + e^x, & \text{якщо } 1.5 < x \leq 2, a = 4 \\ \frac{\sin^2(x)}{x}, & \text{якщо } x \geq 6 \\ |x| - & \text{в інших випадках} \end{cases}$
11.  $Y = \begin{cases} e^{\sin N}, & \text{якщо } N = 3 \\ \lg^2 N + ex, & \text{якщо } N \geq 6 \\ 3\pi - & \text{в інших випадках} \end{cases}$
12.  $Y = \begin{cases} \operatorname{tg} x & \text{якщо } 0 < x < \frac{\pi}{2} \\ \operatorname{bc} \operatorname{tg} x & \text{якщо } \frac{\pi}{2} < x < \pi \\ 5 & \text{в інших випадках} \end{cases}$
13.  $T = \begin{cases} x^{y+1} + e^{y-1} & \text{якщо } x > 0 \\ \frac{|y-x|^3}{2 + \operatorname{tg} z} & \text{якщо } x < 0; 1 < z < 2 \\ xuz & \text{в інших випадках} \end{cases}$
14.  $H = \begin{cases} (x+y)/(5 + \sin^2 z) & \text{якщо } x > 0 \\ \sqrt{|y + \operatorname{tg}^2 z|} & \text{якщо } y < 0, 0 < z < 3 \\ y \cdot e^x & \text{в інших випадках} \end{cases}$
15.  $K = \begin{cases} y^x + 3z & \text{якщо } y < 0, z > 0 \\ e^{|x-y|} + \frac{x}{2} & \text{якщо } y < 0, x > 0 \\ x^2 & \text{в інших випадках} \end{cases}$
16.  $F = \begin{cases} \ln^2(x+2) & \text{якщо } x > 0 \\ y/2 + \sin^3 z & \text{якщо } 1 < z < 2, y > 0 \\ \pi & \text{в інших випадках} \end{cases}$
17.  $Y = \begin{cases} x\sqrt{b^3} & \text{якщо } 0 < x < 4 \text{ и } b > 0 \\ b \sin^2 x & \text{якщо } x \geq 4 \text{ и } b = 1 \\ e^x & \text{в інших випадках} \end{cases}$
18.  $A = \begin{cases} 2^{-x\sqrt{x+y}} & \text{якщо } x > 0 \text{ и } y < 0 \\ \sqrt{e^{x+1} + \sin z} & \text{якщо } 0 < x < 5 \text{ и } z = 2 \\ 0 & \text{в інших випадках} \end{cases}$
19.  $K = \begin{cases} \frac{x-y/2}{2(x+y)} & \text{якщо } y \leq 0 \text{ и } x \neq 0 \\ x/\sin z & \text{якщо } 1 > z > 2 \text{ и } x > 0 \\ x^y & \text{в інших випадках} \end{cases}$

$$20. H = \begin{cases} x^2 - 5 & \text{якщо } x > 2 \\ \lg x + 4 & \text{якщо } x = 2 \\ x^3 - 2 & \text{в інших випадках} \end{cases}$$

## Керуючі конструкції мови програмування

### Оператори циклу

Схожі і повторювані дії виконуються за допомогою операторів циклу *for* і *while*. Цикл *for* призначений для виконання повторюваних дій задану кількість разів – арифметичний цикл, а *while* - для дій, число яких заздалегідь не відомо, але відома умова продовження циклу.

### Цикл *for*

Використання *for* здійснюється наступним чином:

```
for count = start : final
    команди MatLab
end
```

Вираз *count* – це змінна циклу, *start* - початкове значення змінної циклу, *final* - кінцеве значення змінної циклу, а *step* - крок, на який збільшується *count* при кожному наступному заході в цикл. Цикл закінчується, як тільки значення *count* стає більше за *final*. Змінна циклу може приймати не тільки цілі, але і дійсні значення будь-якого знаку.

Приклад застосування циклу *for*. Нехай потрібно вивести графіки сімейства кривих для  $x$  що належать інтервалу  $[0, 2\pi]$ . Сімейство кривих задано функцією  $y(x, a) = e^{-ax} \sin x$ , що залежить від параметра  $a$ , для значень параметра  $a$  від  $-0,1$  до  $0,1$  з кроком  $0,02$ . Можна, звичайно, послідовно обчислювати  $y(x, a)$  і будувати її графіки для різних значень  $a$  від  $-0,1$  до  $0,1$ . Але набагато зручніше використовувати цикл *for*.

Наберемо текст файл-програми в редакторі М-файлів. Для цього необхідно розкрити меню File робочого середовища MatLab, в пункті New вибрати підпункт M-file (рис. 21). Після цього відкриється вікно редактора М-файлів Untitled, в якому і виконується набір програми. Відкрити вікно Untitled для набору програми можна також натиснувши кнопку New M-file на панелі інструментів робочого середовища.

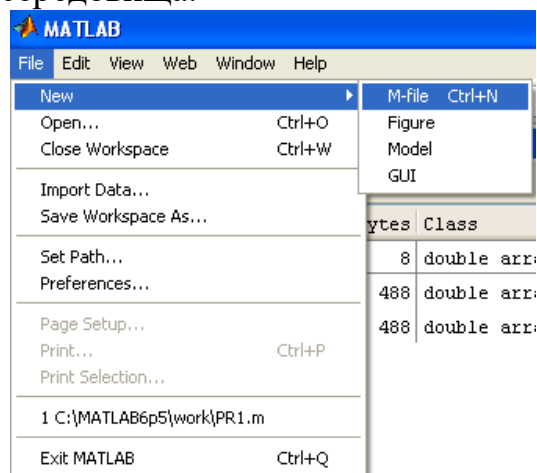


Рис. 21. Команди виклику вікна Untitled для набору програми

Новий файл для набору рядків програми відкривається у вікні Untitled редактора М-файлів, яке наведено на рис. 22.

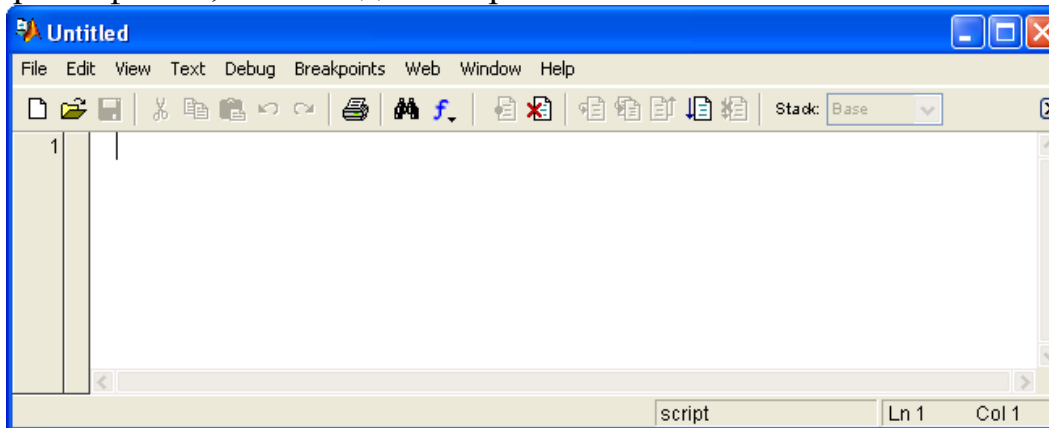


Рис. 22. Вікно Untitled редактора М-файлів MatLab

Наберемо у редакторі команди програми (рис. 23).

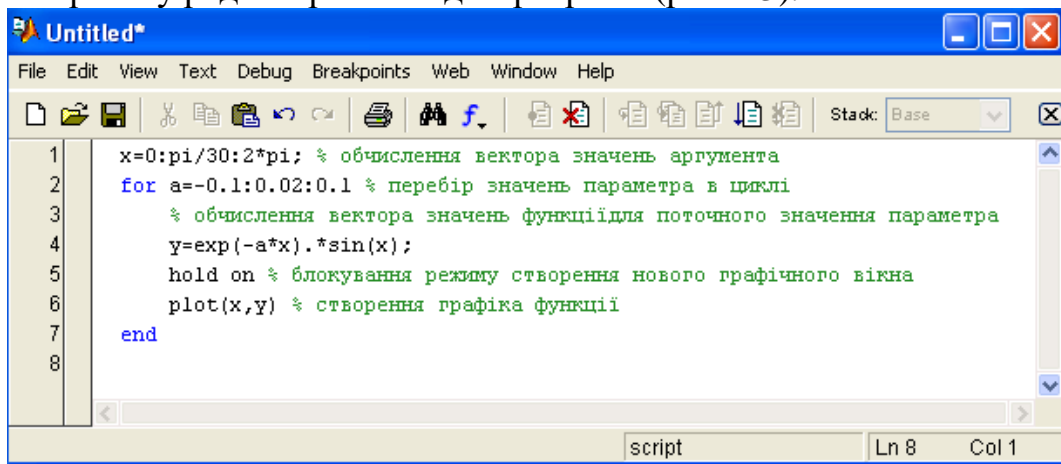


Рис. 23. Набір команд програми у вікні редактора

Набравши рядки програми необхідно зберегти файл з ім'ям PR1, вибравши в меню File редактора пункт Save as (рис. 24).

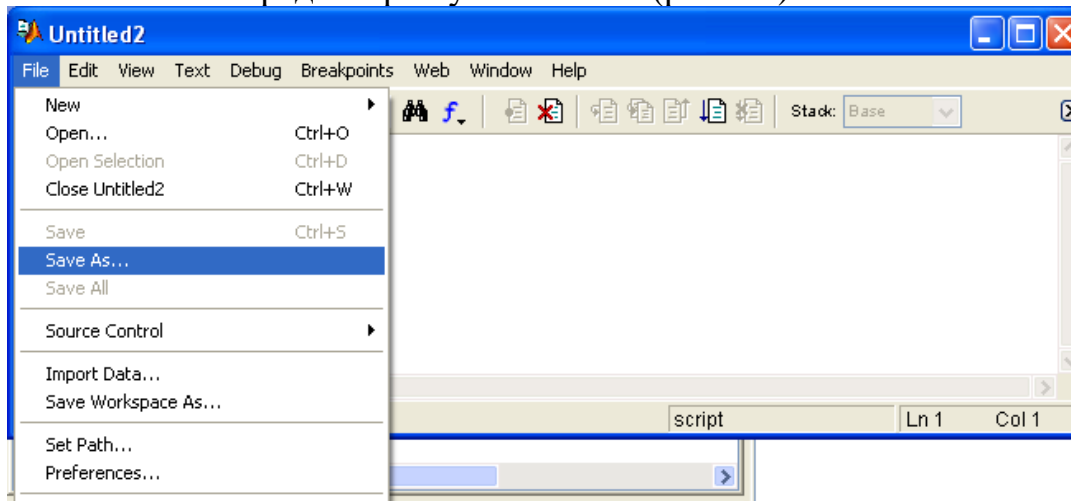


Рис. 24. Вікно збереження файлу

На екрані з'явиться вікно для збереження файлу, в якому в рядку Ім'я файлу необхідно задати ім'я файлу, наприклад PR1 (рис. 25).

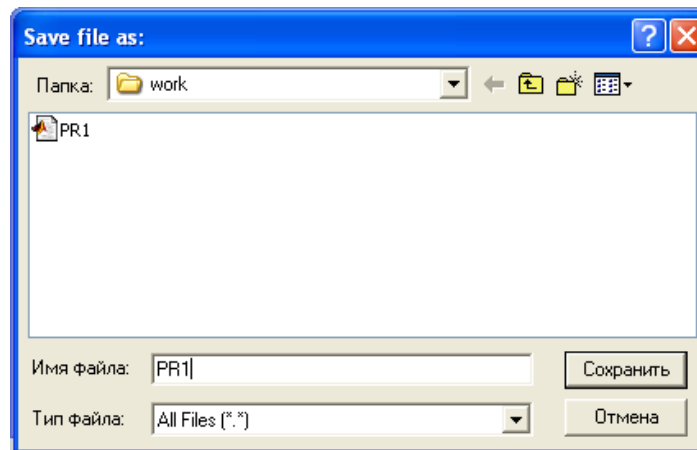


Рис. 25. Вікно для збереження файлу

Для запуску на виконання команд файлу слід вибрати пункт Run в меню Debug або просто натиснути кнопку F5, розташовану на клавіатурі (рис. 26).

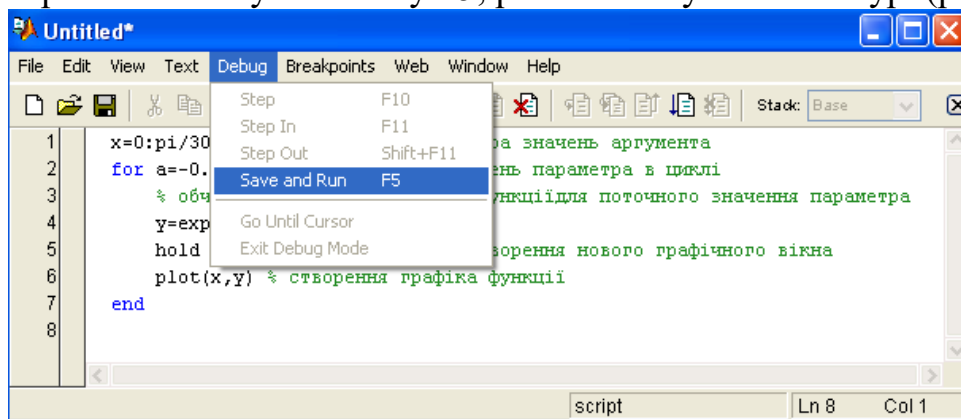


Рис. 26. Команди запуску програми на рішення

В результаті виконання програми PR1 з'явиться графічне вікно, зображене на рисунку 27, яке містить потрібне сімейство кривих.

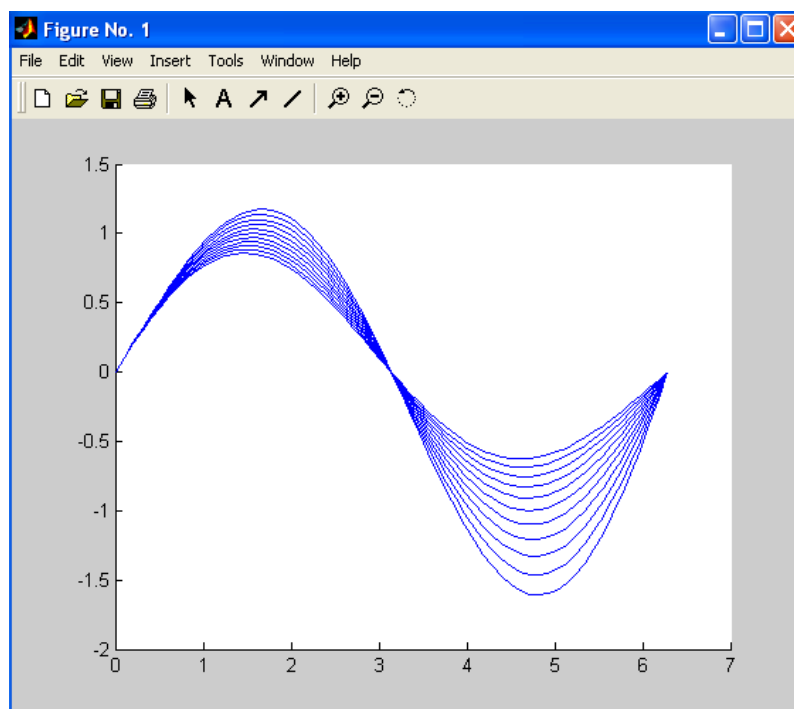


Рис. 27. Результат виконання програми PR1

Розглянемо на прикладах використання конструкції `for ...end`.

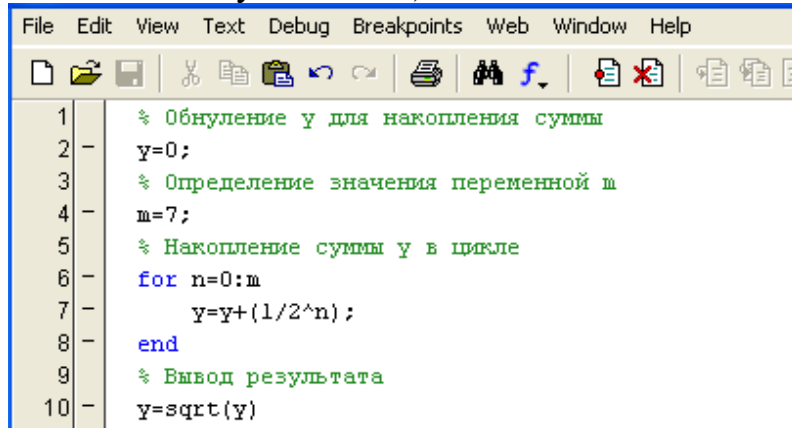


## Зразок виконання завдань

**Завдання 1.** Обчислення кореня квадратного із суми виразів:

$$Y = \sqrt{\sum_{n=0}^m \frac{1}{2^n}}$$

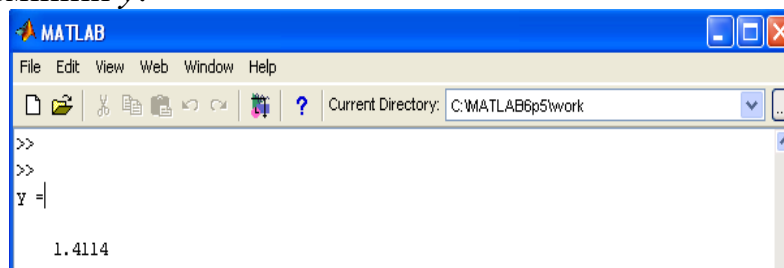
Складемо програму для реалізації даного виразу. Наберемо текст файл-програми в редакторі М-файлів (команда меню File робочого середовища MatLab → пункт New → підпункт M-file).



```

File Edit View Text Debug Breakpoints Web Window Help
1 % Обнуление y для накопления суммы
2 y=0;
3 % Определение значения переменной m
4 m=7;
5 % Накопление суммы y в цикле
6 for n=0:m
7     y=y+(1/2^n);
8 end
9 % Вывод результата
10 y=sqrt(y)
  
```

Збережемо програму вибравши в меню File редактору М-файлів команду Save as та задавши ім'я файлу. В меню команди Debug вибрати пункт Run або натиснути кнопку F5 для рішення даної програми. Обчислене значення суми знаходиться в змінній у:



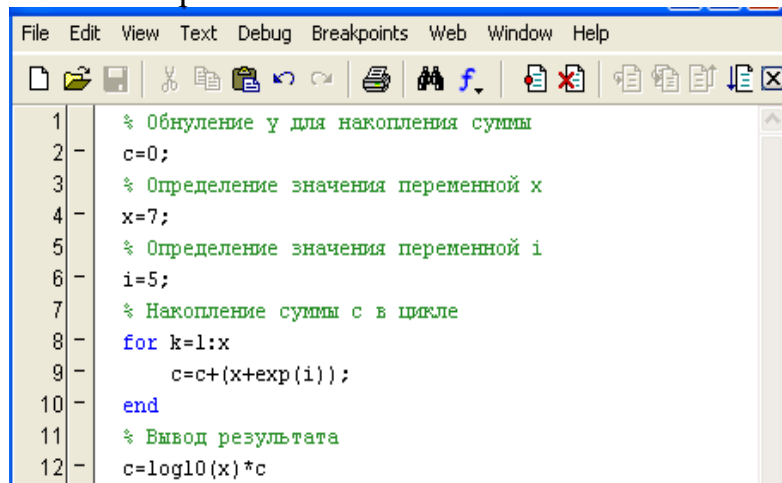
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
>>
y =
    1.4114
  
```

**Завдання 2.** Обчислення суми виразів:

$$C = Lg x \sum_{k=1}^x x + e^i$$

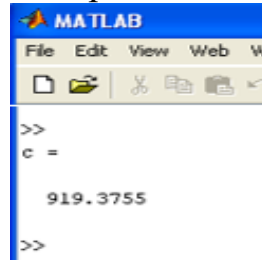
У вікні редактора зробимо набір команд програми. Також введемо в програмі текстові коментарі:



```

File Edit View Text Debug Breakpoints Web Window Help
1 % Обнуление y для накопления суммы
2 c=0;
3 % Определение значения переменной x
4 x=7;
5 % Определение значения переменной i
6 i=5;
7 % Накопление суммы c в цикле
8 for k=1:x
9     c=c+(x+exp(i));
10 end
11 % Вывод результата
12 c=log10(x)*c
  
```

Здійснимо запуск програми на рішення і отримаємо результат:

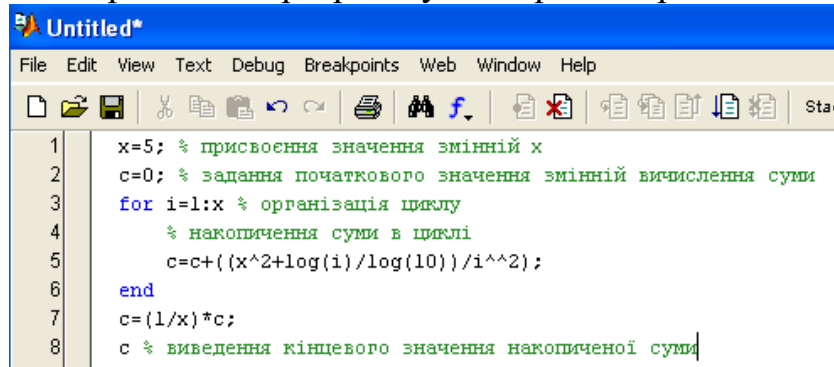


```

MATLAB
File Edit View Web W
>>
c =
    919.3755
>>
  
```

**Завдання 3.** Обчислення суми:  $C = \frac{1}{x} \sum_{i=1}^x \frac{x^2 + \lg i}{i^2}$ .

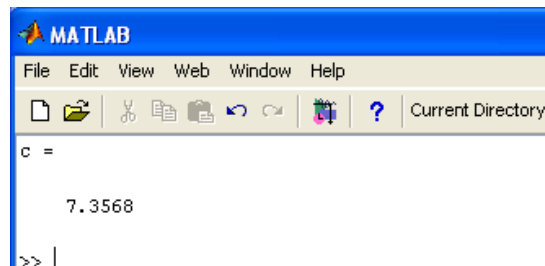
Виконаємо набір команд програми у вікні редактора:



```

Untitled*
File Edit View Text Debug Breakpoints Web Window Help
x=5; % присвоєння значення змінній x
c=0; % задання початкового значення змінній вичислення суми
for i=1:x % організація циклу
    % накопичення суми в циклі
    c=c+((x^2+log(i)/log(10))/i^2);
end
c=(1/x)*c;
c % виведення кінцевого значення накопиченої суми
  
```

Для розв'язання програми здійснимо запуск програми на рішення і отримаємо результат:

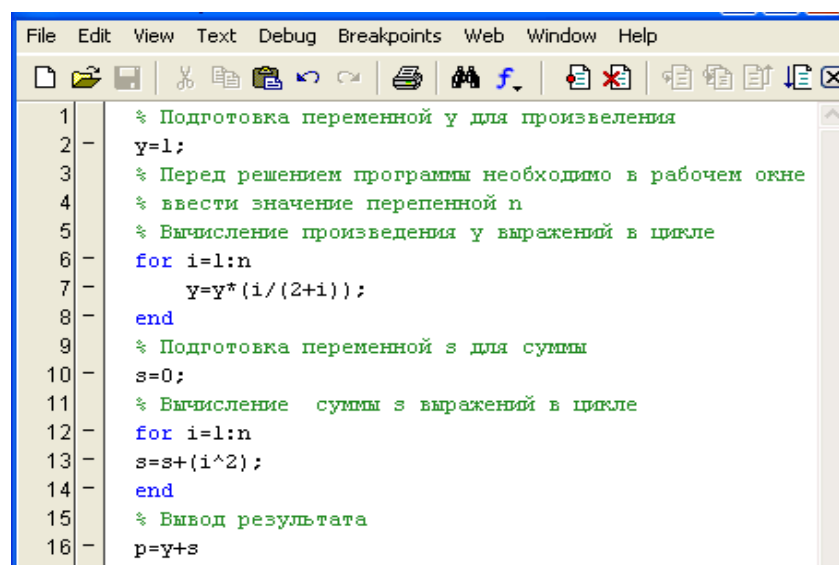


```

MATLAB
File Edit View Web Window Help
Current Directory:
c =
    7.3568
>>
  
```

**Завдання 4.** Обчислення суми виразів добутку та суми:

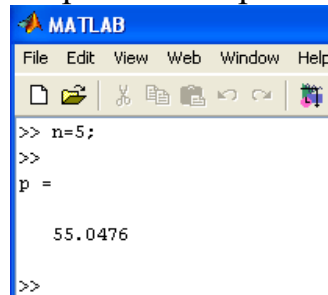
$$P = \prod_{i=1}^n \frac{i}{2+i} + \sum_{i=1}^n i^2$$



```

File Edit View Text Debug Breakpoints Web Window Help
% Подготовка переменной y для произведения
y=1;
% Перед решением программы необходимо в рабочем окне
% ввести значение переменной n
% Вычисление произведения y выражений в цикле
for i=1:n
    y=y*(i/(2+i));
end
% Подготовка переменной z для суммы
z=0;
% Вычисление суммы z выражений в цикле
for i=1:n
    z=z+(i^2);
end
% Вывод результата
p=y+z
  
```

Після запуску програми на рішення отримаємо результат:



```

MATLAB
File Edit View Web Window Help
[Icons]
>> n=5;
>>
p =
    55.0476
>>
  
```

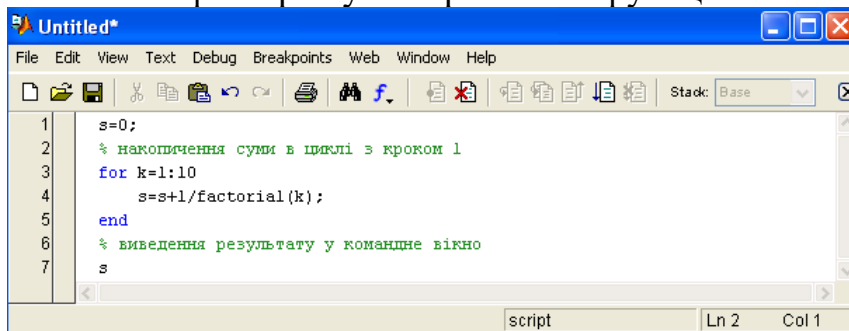
**Факторіал.** Обчислення факторіалу – це добуток ряду від 1 до заданого числа. Наприклад необхідно обчислити  $k!$ , це обчислення добутку від 1 до  $k$ , зрозуміло, що треба ввести значення  $k$  і потім здійснити обчислення. В MatLab існує функція для обчислення факторіалу. Набравши в командному рядку *help factorial* можна отримати інформацію про призначення і спосіб виклику файл-функції. Формат функції factorial(n). Використовується функція в циклі, який організовано від 1 до n. Дії функції, які виконуються в циклі, це *prod(1:n)*, тобто виконується перемноження елементів від 1 до n.

**Завдання 5.** Складемо файл-програму для обчислення суми факторіалів:

$$s = \sum_{k=1}^{10} \frac{1}{k!}$$

Алгоритм обчислення суми використовує накопичення результату, тобто спочатку сума дорівнює нулю, потім в змінну  $k$  заноситься одиниця, обчислюється  $1/k!$  (тобто  $1/1!$ ), додається до  $s$  і результат знову заноситься в  $s$ . Далі  $k$  збільшується на одиницю, і процес продовжується, поки останнім складовим не стане  $1/10!$ . В тому випадку, коли крок циклу дорівнює 1, його можна не вказувати.

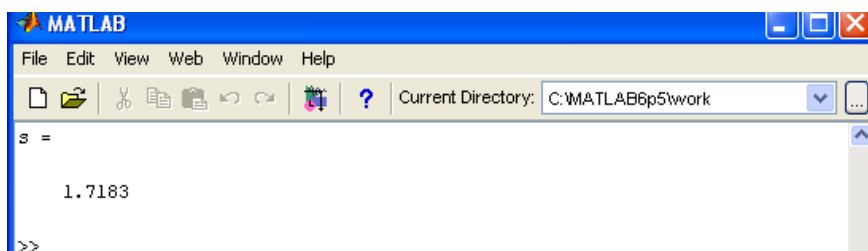
Наберемо файл-програму обчислення суми факторіалів в редакторі М-файлів. Для обчислення факторіалу використаємо функцію:



```

Untitled*
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 s=0;
2 % накопичення суми в циклі з кроком 1
3 for k=1:10
4     s=s+1/factorial(k);
5 end
6 % виведення результату у командне вікно
7 s
script Ln 2 Col 1
  
```

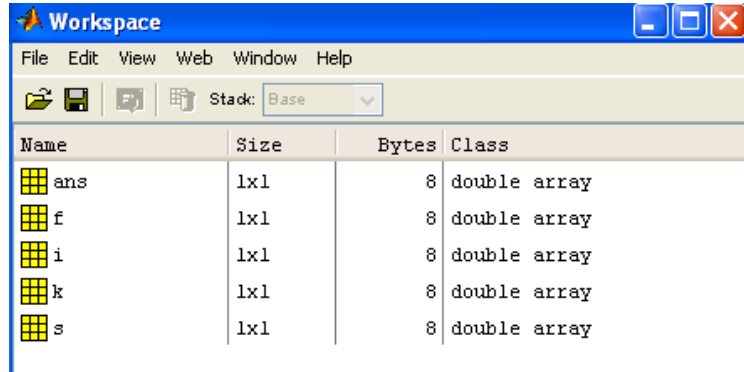
Збережемо її в файлі з іменем, наприклад PR2.m і виконаємо рішення. Результат рішення відобразиться в командному вікні:



```

MATLAB
File Edit View Web Window Help
[Icons] Current Directory: C:\MATLAB6p5\work
s =
    1.7183
>>
  
```

Усі змінні, які використовуються в файл-програмі, стають доступними в робочому середовищі (вікно Workspace). Вікно Workspace з його вмістом представлено нижче.



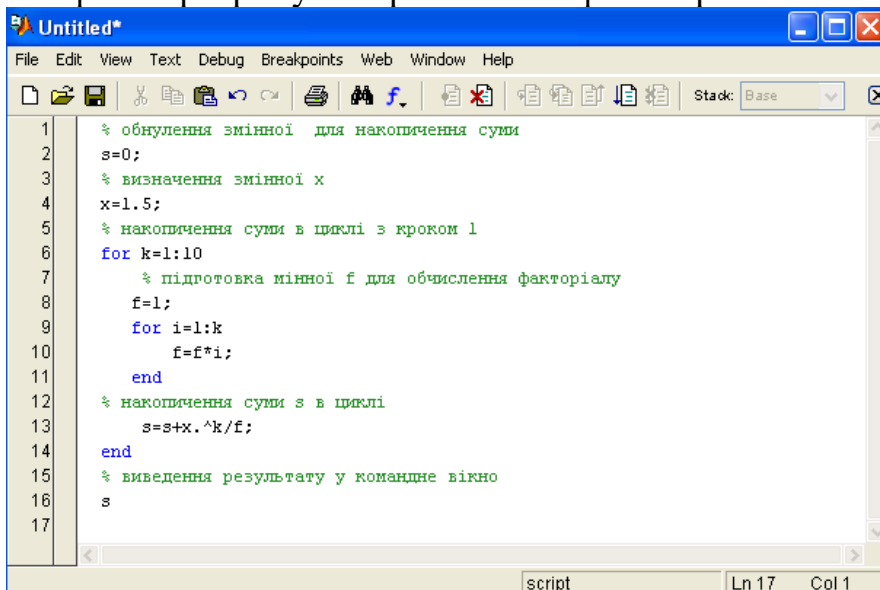
Name	Size	Bytes	Class
ans	1x1	8	double array
f	1x1	8	double array
i	1x1	8	double array
k	1x1	8	double array
s	1x1	8	double array

**Завдання 6.** Розглянемо приклад на обчислення суми, яка залежить не тільки від змінної  $k$  але і від змінної  $x$ . 
$$s(x) = \sum_{k=1}^{10} \frac{x^k}{k!}.$$

Спочатку в тексті файл-програми необхідно визначити змінну  $x$ , потім організувати цикл і в ньому організувати обчислення суми. В якості  $x$  може бути вектор або матриця, оскільки в файл-програмі при накопиченні суми використані поелементні операції.

Цикли *for* можуть бути вкладені один в одного, при цьому змінні вкладених циклів повинні бути різними.

Складемо файл-програму і збережемо її в файлі присвоївши ім'я.

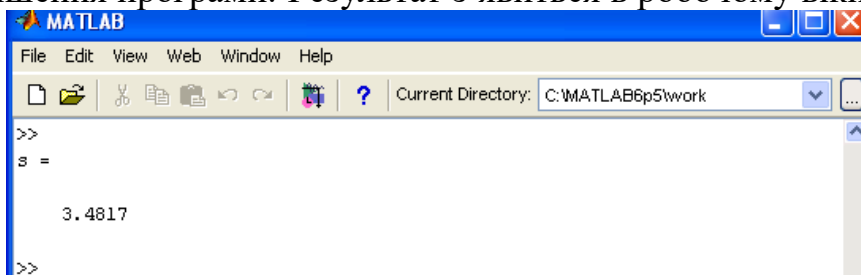


```

1  % обнулення змінної для накопичення суми
2  s=0;
3  % визначення змінної x
4  x=1.5;
5  % накопичення суми в циклі з кроком 1
6  for k=1:10
7      % підготовка мінної f для обчислення факторіалу
8      f=1;
9      for i=1:k
10         f=f*i;
11     end
12     % накопичення суми s в циклі
13     s=s+x.^k/f;
14 end
15 % виведення результату у командне вікно
16 s
17

```

Вибравши пункт Run в меню команди Debug або натиснувши F5 виконаємо рішення програми. Результат з'явиться в робочому вікні MatLab:

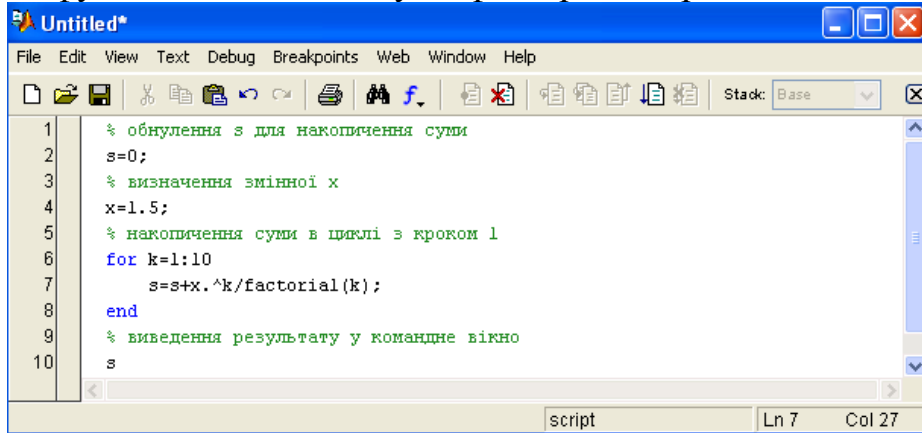


```

>>
s =
    3.4817
>>

```

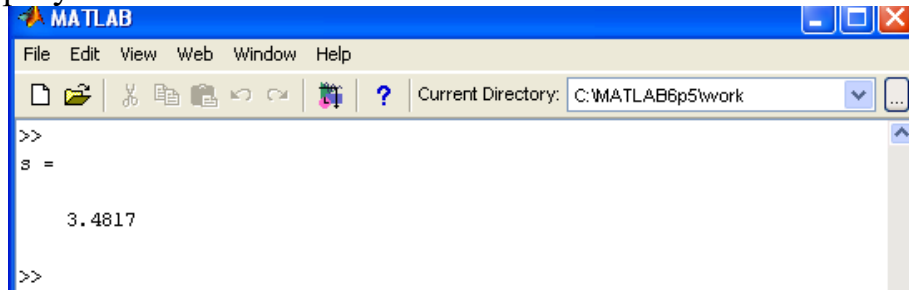
Використання конструкцій мови програмування, зокрема, циклів, в файл-функціях здійснюється так же, як і в файл-програмах. Розглянемо даний приклад, використовуючи при розв'язанні функцію для обчислення факторіалу. Лістинг файл-функції обчислення суми факторіалів представлено нижче.



```

1 % обнулення s для накопичення суми
2 s=0;
3 % визначення змінної x
4 x=1.5;
5 % накопичення суми в циклі з кроком 1
6 for k=1:10
7     s=s+x.^k/factorial(k);
8 end
9 % виведення результату у командне вікно
10 s
  
```

Збережемо файл-функцію і виконаємо її рішення в робочому вікні MatLab отримаємо результат:

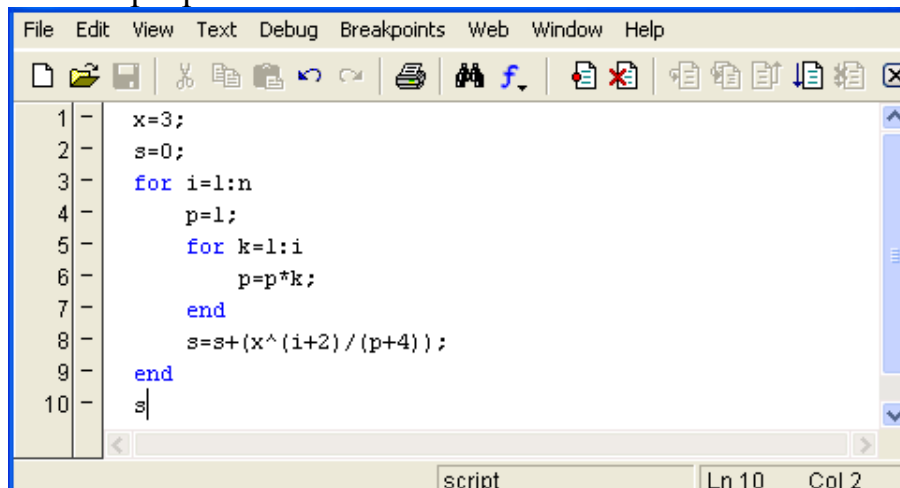


```

>>
s =
    3.4817
>>
  
```

**Завдання 7.** Приклад обчислення суми виразів, в яких обчислюються факторіали  $S = \sum_{i=1}^n \frac{x^{i+2}}{i!+4}$ . При обчисленні необхідно знайти суму факторіалів

MatLab. Під час складання програми необхідно здійснювати обчислення циклу в циклі: цикл для обчислення суми – зовнішній, і цикл для обчислення кожного проміжного факторіалу – внутрішній. Кінцеве значення змінної циклу –  $n$  краще задавати в робочому середовищі MatLab (в рядку вводу вікна Command Window). Лістинг програми:



```

1 x=3;
2 s=0;
3 for i=1:n
4     p=1;
5     for k=1:i
6         p=p*k;
7     end
8     s=s+(x^(i+2)/(p+4));
9 end
10 s
  
```

Результат рішення програми обчислення суми факторіалів у вікні Command Window:

```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLABE
>> n=3;
>>
s =
43.2000

```

Розглянуті приклади циклічних обчислювальних процесів відносяться до арифметичних циклів. Арифметичний цикл, це цикл, у якого число повторень відомо заздалегідь або його можна обчислити.

### Індивідуальні завдання за темою «Оператори циклу» Цикл *for*

#### Варіанти індивідуальних завдань

№	Прості цикли	Вложені цикли
1	$Y = \sum_{i=1}^n \frac{(-1)^{i+1}}{i(i+1)}$	$Y = \sum_{k=0}^n \frac{(-1)^k (k+1)}{k!}$
2	$A = \prod_{k=1}^N \left(1 + \frac{\sin(kx)}{k}\right)$	$Y = m! + \sum_{k=1}^m \sqrt{a + \frac{1}{k}}$
3	$Y = \sum_{k=1}^n \frac{(A)^k}{(2k+1)k}$	$P = \prod_{k=1}^n \left(1 + \frac{k^2 \cdot x}{k!}\right)$
4	$F = \sum_{k=1}^N \frac{(-1)^{k+1}}{k(k+1)}$	$P = \prod_{i=1}^n \frac{(1-x)^{i+1} + 1}{((i-1)! + 1)^2}$
5	$P = a \sum_{i=1}^N e^{x+i}$	$S = \sum_{i=1}^n \left(\frac{1}{i!} + \sqrt{ x }\right)$
6	$A = \sum_{k=1}^N \frac{(-1)^k (k+1)}{k}$	$H = 1 + \sum_{k=1}^n \frac{a^k}{k!}$
7	$Y = \sqrt{\sum_{k=1}^n x^k} + e$	$Y = \sum_{k=1}^n (k!) - 2n!$
8	$Y = \sum_{i=1}^n \frac{(-1)^i}{i(i+1)}$	$Y = 1 + \sum_{k=1}^n \frac{x^k}{k!}$
9	$S = \sum_{i=1}^n \frac{x^{i+2}}{i+4}$	$S = 5 \sqrt[n]{n! \sum_{i=1}^n \frac{1}{i^2}}$

10	$F = \sum_{k=1}^N \frac{(-1)^k}{(2k+1)^k} \cdot a$	$Y = \sum_{i=1}^n \frac{(-2)^i}{i!} \lg(b)$
11	$Y = \ln\left(b + \prod_{i=1}^3 \lg(i \cdot b)\right)$	$Y = \sum_{k=1}^n \frac{(-1)^k}{k!} \cdot \frac{b}{n}$
12	$G = \sum_{i=1}^N (N + \sqrt{ B })$	$S = \sum_{i=1}^n \frac{x}{i!} + 2 \cdot \sqrt{ x }$
13	$A = \sum_{i=1}^N \frac{x^i}{i} + e^i$	$Y = \sum_{i=1}^n \sum_{j=1}^i \frac{1}{2^{j+i}}$
14	$C = \frac{1}{x} \sum_{i=1}^x \frac{x^2 + \lg i}{i^2}$	$Y = \sum_{k=1}^n \frac{(-1)^k}{k!}$
15	$K = \frac{1}{\pi} \sum_{i=1}^6 \frac{x+i}{x}$	$Y = \prod_{k=1}^n \frac{e^{n+5}}{k!}$
16	$R = e^x \sum_{i=1}^7 (\pi + \lg X)$	$P = \prod_{i=1}^n \left(x - \frac{1}{i!}\right)^2$
17	$K = \pi x \prod_{i=1}^9 i x + 1$	$P = \prod_{k=1}^n \left(1 + \frac{\sin(kx)}{k!}\right)$
18	$C = \lg x \sum_{k=1}^x x + e^i$	$Y = \sum_{k=1}^m \prod_{i=1}^k \frac{i+1}{i^2 - 0,5}$
19	$F = \sqrt{x} + \sum_{k=1}^N x^k$	$Y = \prod_{k=li=1}^n \sum \frac{i^2}{n}$
20	$P = a \prod_{i=1}^{12} (3i^2 + a)$	$S = \sum_{i=1}^n \sum_{j=1}^i \frac{1}{j+3i}$

### Ітераційні цикли. Цикл `while`, знаходження суми ряду

Цикл `for` використовується, коли визначено кінцеве число дій. Але існують алгоритми з заздалегідь невідомою кількістю повторень. Реалізувати їх дозволяє більш гнучкий цикл `while`.

Цикл `while` використовується для організації повторів однотипових дій у випадку, коли число повторів заздалегідь невідомо і визначається виконанням деякої умови. Цикл `while` працює, доки виконується умова циклу:

```
while умова повторення циклу
    команди MatLab
end
```

Розглянемо приклад. Визначити суму і кількість ітерацій, при якій буде виконуватися рівність з точністю до 0,001.:

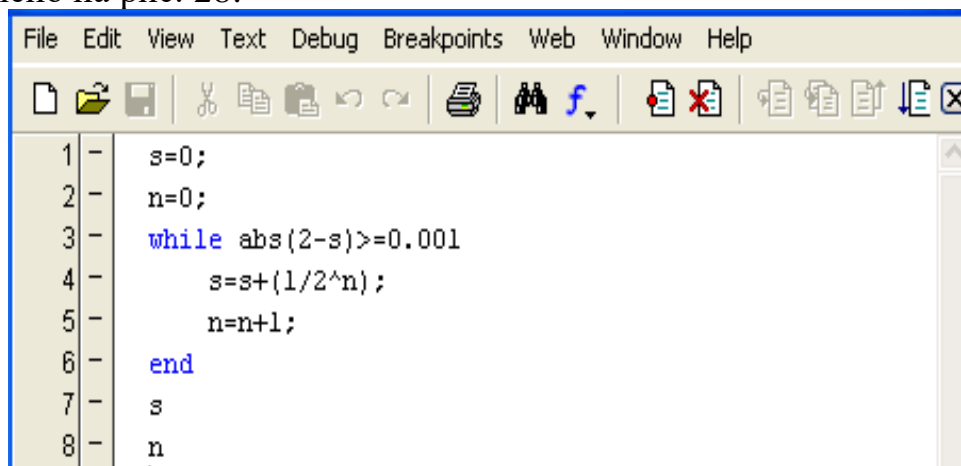
$$\sum_{n=0}^{\infty} \frac{1}{2^n} = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = 2$$

При складанні програми треба в першу чергу надати початкове значення змінній, в якій буде накопичуватися сума ( $s = 0$ ) та надати початкове значення змінній, в якій буде вестись підрахунок кількості ітерацій ( $n = 0$ ).

Слід звернути цвагу на те, що у цикла `while`, на відміну від `for`, немає змінної циклу, тому необхідно до початку циклу змінній  $n$ , яка буде вести підрахунок кількості ітерацій (повторень у циклі) присвоїти початкове значення, тобто нуль, а всередині циклу збільшувати значення  $n$  на одиницю.

У заданому прикладі межа підсумовування нескінченність. Для рішення таких прикладів необхідно накопичувати суму доки виконується умова циклу.

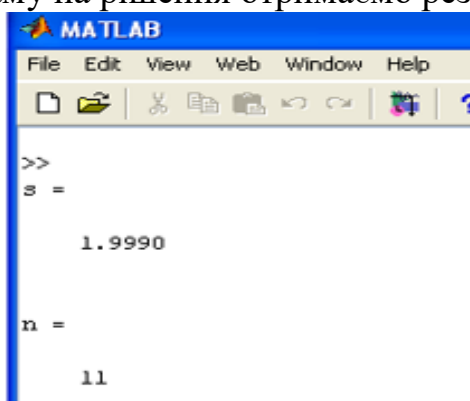
Лістинг файл-функції, яка обчислює кількість ітерацій і суму ряду представлено на рис. 28:



```

File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 - s=0;
2 - n=0;
3 - while abs(2-s)>=0.001
4 -     s=s+(1/2^n);
5 -     n=n+1;
6 - end
7 - s
8 - n
  
```

Рис. 28. Лістинг програми підрахунку суми та ітерацій  
Запустивши програму на рішення отримаємо результат (рис. 29):



```

MATLAB
File Edit View Web Window Help
[Icons]
>>
s =
    1.9990
n =
    11
  
```

Рис. 29. Результат роботи програми

З результату можна зробити висновок, що за 11 ітерацій було накопичено суму, яка дорівнює 1,999.

Умова циклу `while` може містити логічний вираз, складений з операцій відношення і логічних операцій або операторів. Для задання умови повторення циклу допустимі операції відношення.

Розглянемо на прикладах використання конструкції `while ...end`.

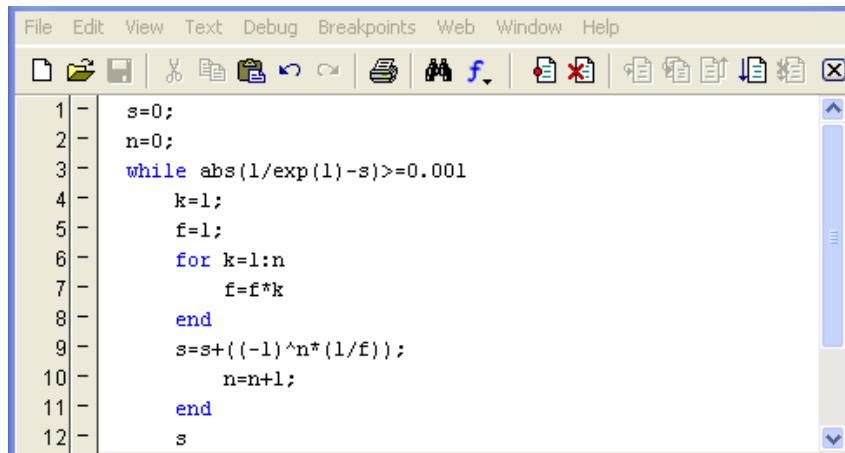


## Зразок виконання завдань

**Завдання 1.** Знайти суму ряду:

$$\sum_{n=0}^{\infty} (-1)^n \frac{1}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots = \frac{1}{e}$$

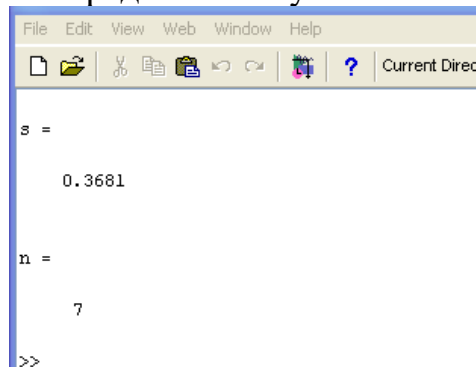
Межею підсумовування являється нескінченність. Для рішення прикладу необхідно, щоб абсолютна величина різниці між складовими і заданою точністю не перевищувала б значення 0,001. Лістинг програми представлено у вікні Untiled



```

File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 - s=0;
2 - n=0;
3 - while abs(1/exp(1)-s)>=0.001
4 -     k=1;
5 -     f=1;
6 -     for k=1:n
7 -         f=f*k
8 -     end
9 -     s=s+((-1)^n*(1/f));
10 -     n=n+1;
11 - end
12 - s
  
```

А результат її рішення представлено у вікні Command Window.

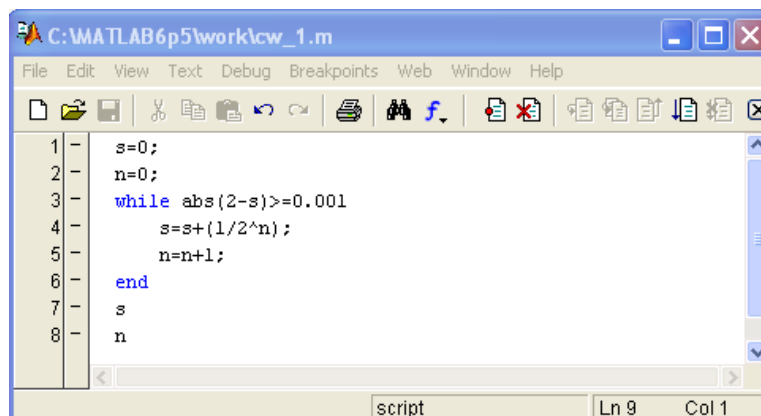


```

File Edit View Web Window Help
[Icons] Current Direc
s =
    0.3681
n =
    7
>>
  
```

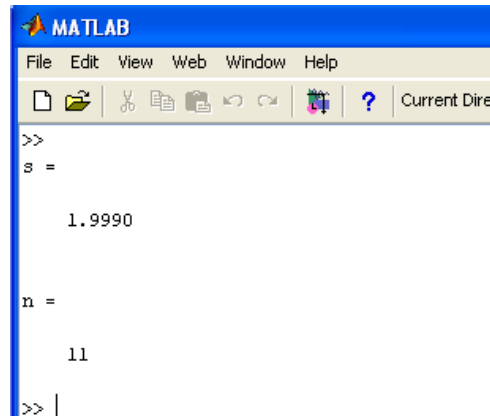
**Завдання 2.** Нехай задана сума ряду  $\sum_{n=0}^{\infty} \frac{1}{2^n} = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = 2$ .

Визначити кількість ітерацій, при яких буде виконуватися рівність з точністю до 0,001. Лістинг програми і результат рішення представлено у вікнах пакету MatLab.



```

C:\MATLAB6p5\work\cw_1.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 - s=0;
2 - n=0;
3 - while abs(2-s)>=0.001
4 -     s=s+(1/2^n);
5 -     n=n+1;
6 - end
7 - s
8 - n
  
```



```

MATLAB
File Edit View Web Window Help
[Icons] ? Current Dire
>>
s =
    1.9990
n =
    11
>> |

```

Розглянуті приклади циклічних обчислювальних процесів відносяться до ітераційних циклів. Ітераційний цикл, це цикл у якого заздалегідь не відомо кількість повторень і його не можна обчислити. В ітераційному циклі число повторень не задається, а визначається в ході виконання циклу. У цьому випадку одне повторення циклу називається ітерацією. В ітераційних циклах на кожному кроці обчислень відбувається послідовне наближення і перевірка умови досягнення бажаного результату. Вихід з ітераційного циклу здійснюється в разі виконання заданої умови, тобто обов'язкове досягнення умови виходу з циклу. В іншому випадку відбудеться "зацікнення" алгоритму.

### Індивідуальні завдання за темою «Ітераційні цикли» Цикл *while*

#### Знайти суму ряду

#### Варіанти індивідуальних завдань:

1.  $\sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots = e;$
2.  $\sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{n!} = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \dots = \ln 2;$
3.  $\sum_{n=0}^{\infty} \frac{1}{2^n} = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = 2;$
4.  $\sum_{n=0}^{\infty} (-1)^n \frac{1}{2^n} = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots = \frac{2}{3};$
5.  $\sum_{n=0}^{\infty} (-1)^n \frac{1}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4};$
6.  $\sum_{n=1}^{\infty} \frac{1}{n(n+1)} = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots = 1;$
7.  $\sum_{n=1}^{\infty} \frac{1}{(2n-1)(2n+1)} = \frac{1}{1 \cdot 3} + \frac{1}{3 \cdot 5} + \frac{1}{5 \cdot 7} + \dots = \frac{1}{2};$

$$8. \sum_{n=2}^{\infty} \frac{1}{(n-1)(n+1)} = \frac{1}{1 \cdot 3} + \frac{1}{2 \cdot 4} + \frac{1}{3 \cdot 5} + \dots = \frac{3}{4};$$

$$9. \sum_{n=1}^{\infty} \frac{1}{(4n-1)(4n+1)} = \frac{1}{3 \cdot 5} + \frac{1}{7 \cdot 9} + \frac{1}{11 \cdot 13} + \dots = \frac{1}{2} - \frac{\pi}{8};$$

$$10. \sum_{n=1}^{\infty} \frac{1}{n(n+1)(n+2)} = \frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{2 \cdot 3 \cdot 4} + \dots = \frac{1}{4};$$

$$11. \sum_{n=1}^{\infty} \frac{1}{n^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots = \frac{\pi^2}{6};$$

$$12. \sum_{n=1}^{\infty} (-1)^n \frac{1}{n^2} = 1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \dots = \frac{\pi^2}{12};$$

$$13. \sum_{n=1}^{\infty} \frac{1}{(2n-1)^2} = 1 + \frac{1}{3^2} + \frac{1}{5^2} + \dots = \frac{\pi^2}{8};$$

$$14. \sum_{n=1}^{\infty} \frac{1}{n^4} = 1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots = \frac{\pi^4}{90};$$

$$15. \sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{n^4} = 1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots = \frac{7\pi^4}{720};$$

$$16. \sum_{n=1}^{\infty} \frac{1}{(2n-1)^4} = 1 + \frac{1}{3^4} + \frac{1}{5^4} + \dots = \frac{\pi^4}{96};$$

### Поєднання розгалуженого, циклічного та лінійного обчислювальних процесів

Розглянемо приклад розгалуженого обчислювального процесу, який включає в себе циклічний та лінійний обчислювальні процеси.

$$A = \begin{cases} \prod_{i=1}^n \frac{e^{x+5}}{i!} & , \text{ якщо } 1 \leq x \leq 5 \\ \frac{1}{e^x + 5} & - \text{ в інших випадках} \end{cases}$$

Розгалужений обчислювальний процес складається з двох гілок розв'язання. Згідно умови, якщо вхідна змінна  $x$  буде мати значення в інтервалі від 1 (включаючи 1) до 5 (включаючи 5) то розв'язання здійснюватиметься по першій гілці, тобто буде виконуватись обчислення добутку виразу, який включає обчислення факторіалу. Якщо ж вхідна змінна  $x$  буде приймати значення менші ніж 1, або більші ніж 5, то розв'язання буде здійснюватися по другій гілці, тобто відбудеться обчислення виразу, який зазначений умовою «в інших випадках». Складемо рядки програми у вікні редактора. Лістинг програми представлено на рисунку 30.

```

1  % В рабочем окне надо ввести значения x,n
2  if x>=1 & x<=5
3  % Подготовка переменной a для вычисления произведения
4  a=1;
5  % Подготовка переменной f для вычисления факториала
6  f=1;
7  for i=1:n
8      for j=1:i
9          f=f*j;
10     end
11     a=a*((exp(5))/f);
12     end
13     else
14     a=1/(exp(x)+5);
15     end
16 % Вывод результата в рабочее окно
17 a

```

Рис. 30. Лістинг програми

Результат рішення програми, якщо вхідна змінна  $x$  згідно умови буде мати значення в інтервалі від 1 (включаючи 1) до 5 (включаючи 5), тобто  $x=3$  (рис. 31).

```

>> x=3;
>>
a =
    1
>> |

```

Рис. 31. Результат розв'язання, що відповідає першій умові завдання

Результат рішення програми, якщо вхідна змінна  $x$  буде приймати значення менші ніж 1, або більші ніж 5, тобто  $x=-6$  (рис. 32).

```

>> x=-6;
>>
a =
    0.1999
>> |

```

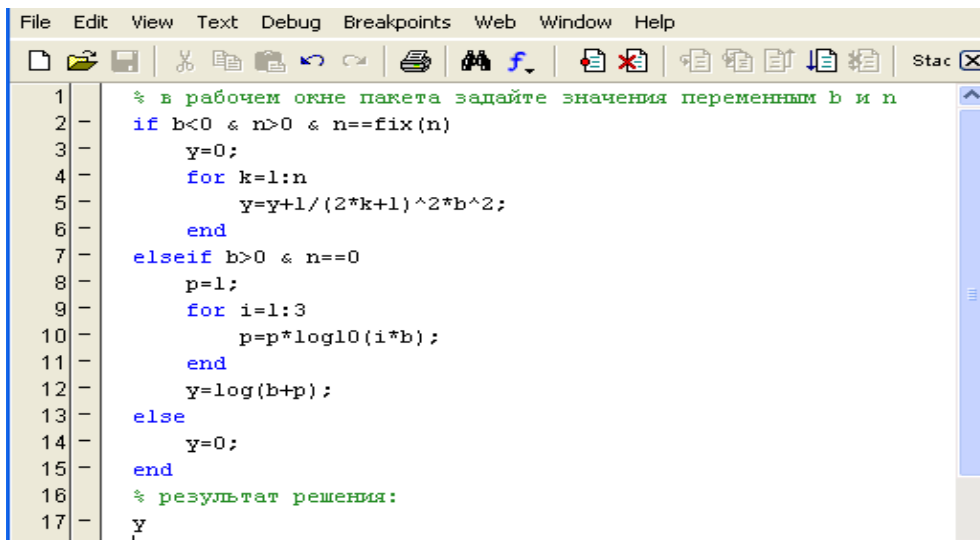
Рис. 32. Результат розв'язання, що відповідає другій умові завдання

### Зразок виконання завдань

**Завдання 1.** Обчислити результат, в залежності від заданої умови.

$$Y = \begin{cases} \sum_{k=1}^n \frac{1}{(2k+1)^2} \cdot b^2, & \text{если } b < 0, n > 0 \text{ и целое} \\ \ln\left(b + \prod_{i=1}^3 \lg(ib)\right), & \text{если } b > 0, n = 0 \\ 0 & \text{в остальных случаях} \end{cases}$$

Вхідними даними є змінні  $b$  та  $n$ . Вихідною змінною є  $y$ . Складемо лістинг програми:

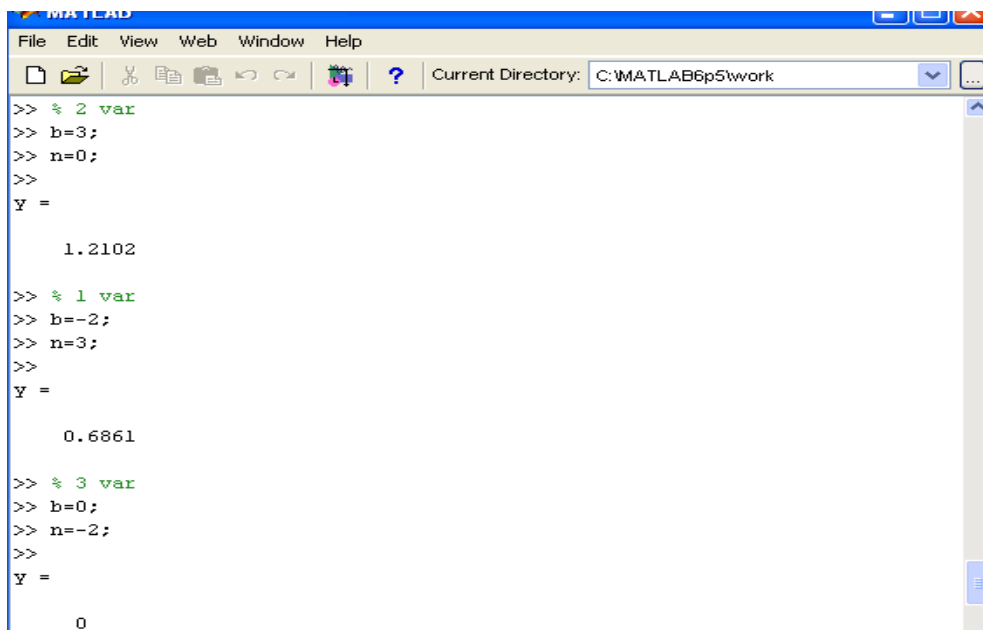


```

1  % в робочем окне пакета задайте значення переменным b и n
2  - if b<0 & n>0 & n==fix(n)
3  -     y=0;
4  -     for k=1:n
5  -         y=y+1/(2*k+1)^2*b^2;
6  -     end
7  - elseif b>0 & n==0
8  -     p=1;
9  -     for i=1:3
10 -         p=p*log10(i*b);
11 -     end
12 -     y=log(b+p);
13 - else
14 -     y=0;
15 - end
16 % результат решения:
17 y

```

Перевіримо роботу програми. Згідно умови, розв'язання може відбуватися по одній із трьох гілок, в залежності від значення вхідних даних. Здійснимо три варіанти рішення, задаючи відповідно варіанту вхідні дані і отримаємо результати.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>> % 2 var
>> b=3;
>> n=0;
>>
Y =
    1.2102

>> % 1 var
>> b=-2;
>> n=3;
>>
Y =
    0.6861

>> % 3 var
>> b=0;
>> n=-2;
>>
Y =
    0

```

Розглянуті приклади поєднують в собі розгалужені, циклічні та лінійні обчислювальних процесів і відносяться до складних обчислювальних процесів.

### **Індивідуальні завдання за темою «Поєднання розгалуженого і циклічного обчислювальних процесів»**

#### **Ускладнені завдання**

Знайти значення суми або добутку в залежності від заданої умови. Значення змінних задати самостійно.

## Варіанти індивідуальних завдань

№	Умова
1	$R = \begin{cases} \sum_{i=1}^x \lg i, & \text{якщо } x > 0 \text{ і ціле} \\ \sum_{i=0}^{10} \frac{(-1)^i \cdot (i+1)}{i!} & \text{в інших випадках} \end{cases}$
2	$Y = \begin{cases} \sum_{k=1}^n \frac{1}{k^5}, & \text{якщо } n\text{-ціле, } m > n \\ \prod_{i=1}^{10} \left(2 + \frac{1}{i!}\right) & \text{– в інших випадках} \end{cases}$
3	$L = \begin{cases} \sum_{i=1}^5 \left(e^{x-1} + \ln \frac{x}{i}\right), & \text{якщо } x > 0 \text{ і кратне } 2 \\ \sum_{i=1}^{50} \frac{1}{i^3} & \text{– в інших випадках} \end{cases}$
4	$Y = \begin{cases} \sum_{k=1}^n \frac{1}{k^5}, & \text{якщо } n\text{-ціле, } 1 < n < 15 \\ \prod_{i=1}^{10} \left(2 + \frac{1}{i!}\right) & \text{– в інших випадках} \end{cases}$
5	$Y = \begin{cases} \sum_{i=1}^n \frac{(-1)^i}{i!} \cdot \frac{b}{z}, & \text{якщо } n > 0 \text{ і ціле, } b > 4, z > 1 \\ \prod_{i=1}^{10} \left(2 + \frac{1}{i!}\right) & \text{– в інших випадках} \end{cases}$
6	$Y = \begin{cases} \sum_{k=1}^n (-1)^k (2k^2 + 1), & \text{якщо } n > 0 \text{ і ціле} \\ \sum_{i=1}^{50} \sum_{j=1}^i \frac{j-i+1}{i+j} & \text{– в інших випадках} \end{cases}$
7	$F = \begin{cases} \sum_{i=1}^n \frac{1}{i!} + \sqrt{ x }, & \text{якщо } n\text{-ціле і } 2 \leq n \leq 12 \\ \sum_{i=1}^{25} \frac{1}{(2i)^2} & \text{– в інших випадках} \end{cases}$
8	$F = \begin{cases} \sum_{k=1}^n \frac{(-1)^{k+1}}{k(k+1)} \cdot x, & \text{якщо } x < 0, n > 0 \text{ і ціле} \\ \sum_{k=1}^{10} k^3 \cdot \sum_{l=1}^{15} (k-l)^2 & \text{– в інших випадках} \end{cases}$
9	$L = \begin{cases} \frac{1}{x} \cdot \sum_{i=1}^{15} i!, & \text{якщо } x > 0 \\ \sum_{i=1}^{25} \sum_{j=1}^i \frac{1}{j+3i} & \text{– в інших випадках} \end{cases}$

10	$Y = \begin{cases} \pi \cdot \prod_{i=1}^z \lg z, & \text{якщо } 1 \leq z \leq 5 \text{ і } z - \text{ціле} \\ \sum_{i=1}^{30} \sum_{j=1}^i \frac{1}{2j+i} & - \text{ в інших випадках} \end{cases}$
11	$Y = \begin{cases} \sum_{i=1}^m \frac{(-1)^i}{i!}, & \text{якщо } m > 0 \text{ і ціле} \\ \sum_{i=2}^{50} \frac{i+1}{i+2} & - \text{ в інших випадках} \end{cases}$
12	$Y = \begin{cases} \prod_{k=1}^n \left( \frac{k}{k+1} - b^k  x  \right), & \text{якщо } n > 0 \text{ і ціле} \\ \sum_{k=1}^{10} \frac{x + \sqrt{ b }}{2^k}, & \text{якщо } x > 0 \text{ і } b \text{ кратне } 2 \\ 0 & - \text{ в інших випадках} \end{cases}$
13	$V = \begin{cases} \sum_{i=1}^n \frac{(-2)^i}{i!} \lg(b), & \text{якщо } n > 10 \text{ і ціле, } 1 < b < 25 \\ \sum_{i=1}^{10} \frac{b}{i(i+1)} & - \text{ в інших випадках} \end{cases}$
14	$Y = \begin{cases} \sum_{i=1}^n \frac{1}{i^2}, & \text{якщо } n > t, n > 0 \\ \sum_{i=1}^m \frac{1}{i!}, & \text{якщо } t > n, m > 0 \\ 0 & - \text{ в інших випадках} \end{cases}$
15	$Y = \begin{cases} \sum_{i=1}^n \frac{x^2 + \lg i}{i^2}, & \text{якщо } n > t, n > 0 \\ \sum_{i=1}^m \frac{\ln x + e^i}{xi!}, & \text{якщо } t > n, m > 0 \\ 0 & - \text{ в інших випадках} \end{cases}$
16	$P = \begin{cases} \prod_{i=1}^n \frac{i}{2+i} + \sum_{i=1}^n i^2, & \text{якщо } n > 0 \text{ і ціле} \\ 0, & \text{якщо } x = 0 \\ \pi \cdot x \sum_{i=1}^5 \frac{1}{i^2} & - \text{ в інших випадках} \end{cases}$
17	$R = \begin{cases} \sum_{i=1}^x \frac{(-1)^i}{i!}, & \text{якщо } x > 0 \text{ і ціле} \\ \prod_{i=1}^5 \frac{x^2 - e}{i} & - \text{ в інших випадках} \end{cases}$

18	$P = \begin{cases} \sum_{i=0}^{15} \frac{a}{4^i + 5^{i+2}}, & \text{якщо } 2 > a > 100 \\ \prod_{i=2}^{10} \left(1 - \frac{1}{i!}\right) & - \text{ в інших випадках} \end{cases}$
19	$F = \begin{cases} \prod_{k=1}^n \left(1 + \frac{k^2 \cdot x}{k!}\right), & \text{якщо } x > 4, n > 0 \\ x\sqrt{b^3}, & \text{якщо } 0 < x < 4, b > 0 \\ 0 & - \text{ в інших випадках} \end{cases}$
20	$Y = \begin{cases} \prod_{x=1}^n e^x, & \text{якщо } n \geq 2 \text{ і ціле} \\ \sum_{i=1}^4 \left(\lg \frac{x}{i} + \ln \frac{x}{i}\right) & - \text{ в інших випадках} \end{cases}$

### Робота з масивами

Усі дані в MATLAB представляються у вигляді масивів. Важливо правильно зрозуміти, як використовувати масиви. Без цього неможлива ефективна робота в MATLAB, зокрема: побудова графіків, рішення задач лінійної алгебри, статистики, обробки даних і багато іншого. Масив – це упорядкована, пронумерована сукупність однорідних даних. Масив повинен мати ім'я. Масиви відрізняють за числом розмірностей: одномірні, двомірні, багатомірні. Розміром масиву називають число елементів вздовж кожного із вимірів. Доступ до елементів здійснюється за допомогою індексу. В MATLAB нумерація елементів масивів починається з одиниці. Це означає, що індекси повинні бути більші або дорівнювати одиниці.

Вектор, вектор-рядок, матриця являються математичними об'єктами, а одномірні, двомірні або багатомірні масиви – способи зберігання цих об'єктів в комп'ютері. Вектор може бути записаний в стовбець (вектор-стовбець) і в рядок (вектор-рядок). Вектор-стовбці і вектор-рядки часто називають просто векторами.

### Формування векторів і матриць

#### Особливості задання векторів і матриць

MatLab – система, спеціально призначена для проведення складних обчислень з векторами, матрицями і масивами. При цьому вона за замовчуванням передбачає, що кожна задана змінна – це вектор, матриця або масив. Все визначається конкретним значенням змінної. Наприклад, якщо задано  $x=1$ , то це означає, що  $x$  – це вектор з одним елементом, який має значення 1. Якщо потрібно задати вектор із трьох елементів, то їх значення необхідно перелічити в квадратних скобках, розділюючи пробілами. Наприклад, присвоєння:

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

```
v =
```

```
1 2 3
```



задає вектор  $v$ , який має три елемента із значеннями 1, 2 і 3. Після введення вектора система виводить його на екран дисплею.

### Використання оператора двокрапка (:)

Часто виникає необхідність здійснити формування упорядкованих числових послідовностей. Такі послідовності потрібні для створення векторів або значень абсциси при побудові графіків. Для цього в MatLab використовується оператор : (двокрапка). Синтаксис оператора:

Початкове\_значення : Крок : Кінцеве\_значення

Дана конструкція формує зростаючу послідовність чисел, яка починається з початкового значення, змінюється із заданим кроком і завершується кінцевим значенням. Якщо Крок не задано, то він приймає значення 1. Якщо кінцеве значення вказано меншим, ніж початкове значення, то видається повідомлення про помилку. Розглянемо приклади застосування оператора.

```
>> 1:5
ans =
    1    2    3    4    5
>> i=0:2:10
i =
    0    2    4    6    8   10
>> j=10:-2:2
j =
   10    8    6    4    2
>> v=0:pi/2:2*pi
v =
    0  1.5708  3.1416  4.7124  6.2832
>> x=1:-0.2:0
x =
  1.0000  0.8000  0.6000  0.4000  0.2000    0
>> x=0:5
x =
    0    1    2    3    4    5
>> cos(x)
ans =
  1.0000  0.5403 -0.4161 -0.9900 -0.6536  0.2837
>> sin(x)/x
ans =
 -0.0862
>> sin(x)./x
Warning: Divide by zero.
ans =
   NaN  0.8415  0.4546  0.0470 -0.1892 -0.1918
```

### Вектор-стовбці і вектор-рядки

Числа в MatLab представляються у вигляді двомірного масиву один на один. Тому при додаванні векторів використовується той же знак плюс, що і для додавання чисел. Для знаходження різниці векторів слід застосовувати знак мінус.

### Введення, додавання і віднімання векторів

Роботу з масивами почнемо з прикладу – обчислення суми векторів  $a$ , який містить числа 1.3, 5.4, 6.9 і  $b$ , який містить числа 7.1, 3.5, 8.2.

$$a = \begin{pmatrix} 1.3 \\ 5.4 \\ 6.9 \end{pmatrix} \quad b = \begin{pmatrix} 7.1 \\ 3.5 \\ 8.2 \end{pmatrix}$$

Для збереження векторів використовуються масиви  $a$  і  $b$ . В командному рядку введемо масив  $a$ , використовуючи квадратні дужки і розділяючи елементи вектор-стовбця крапкою з комою. Крапка з комою в кінці виразу використовується для скасування виведення результату на екран.

```
>> a = [1.3; 5.4; 6.9]
```

```
a =
```

```
1.3000
```

```
5.4000
```

```
6.9000
```

Так як введений вираз не завершено крапкою з комою, то MATLAB автоматично виведе значення змінної  $a$ . Тепер в командному рядку введемо другий вектор – масив  $b$  скасувавши виведення на екран.

```
>> b = [7.1; 3.5; 8.2];
```

Так як введений вираз завершено крапкою з комою, то виведення на екран значень змінної  $b$  скасовується. Обчислимо суму векторів  $a$  і  $b$ :

```
>> c = a+b
```

```
c =
```

```
8.4000
```

```
8.9000
```

```
15.1000
```

Звернемо увагу на інформацію, яка з'явилася у вікні *Workspace* (рис. 33):

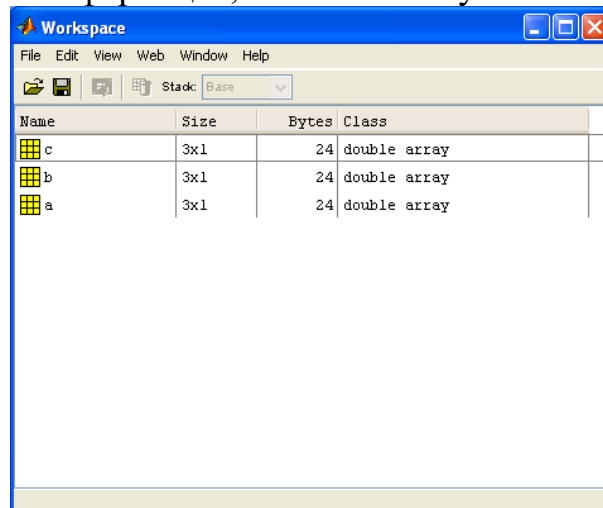


Рис. 33. Вікні Workspace з інформацією про масиви

Вектори  $a$ ,  $b$  і  $c$  зберігаються в двовірних масивах. В стовбці *Size* вказано  $3 \times 1$ , тобто масиви мають по три рядки і одному стовбцю, кожен з них займає 24 байта пам'яті (стовбець *Bytes*).

Аналогічну інформацію про розмір масиву  $c$  можна отримати за допомогою вбудованої функції *size*:

```
>> sa = size(a)
```

```
sa =
```

```
3 1
```

```
>> % Отримана інформація – рядків 3, стовбець 1.
```

Функція *size* дозволяють не тільки вивести характеристики масивів в командне вікно, але й записати їх в змінні для подальшого використання.

Особливість MATLAB представляти усі дані у вигляді масивів являється досить зручною. Нехай, наприклад, потрібно обчислити значення функції *sin* одразу для всіх елементів вектора  $c$  (який зберігається в масиві  $c$ ) і записати результат у вектор  $d$ :

```
>> d = sin(c)
```

```
d =
```

```
0.8546
```

```
0.5010
```

```
0.5712
```

Вбудовані в MATLAB елементарні функції адаптуються під вид аргументів. Якщо аргумент являється масивом, то результат функції буде масивом того ж розміру, але з елементами, які дорівнюють значенню функції від відповідних елементів вхідного масиву.

Розглянемо приклад: необхідно знайти квадратний корінь з елементів вектора  $d$  із знаком мінус. Введемо команди:

```
>> sgrt(-d)
```

```
ans =
```

```
0 + 0.9244i
```

```
0 + 0.7078i
```

```
0 + 0.7558i
```

Оператор присвоєння не використовувався, тому MATLAB записав відповідь в стандартну змінну *ans*.

Введення вектор-рядку здійснюється в квадратних дужках, однак на відміну від вектор-стовбця елементи вектор-рядку при введенні слід розділяти пробілами або комами. Операції додавання, віднімання і обчислення елементарних функцій від вектор-рядків виконується так же, як і з вектор-стовбцями. В результаті отримаємо вектор-рядок такого ж розміру, як і вхідний.

```
>> s1 = [3 4 9 2]
```

```
s1 =
```

```
3 4 9 2
```

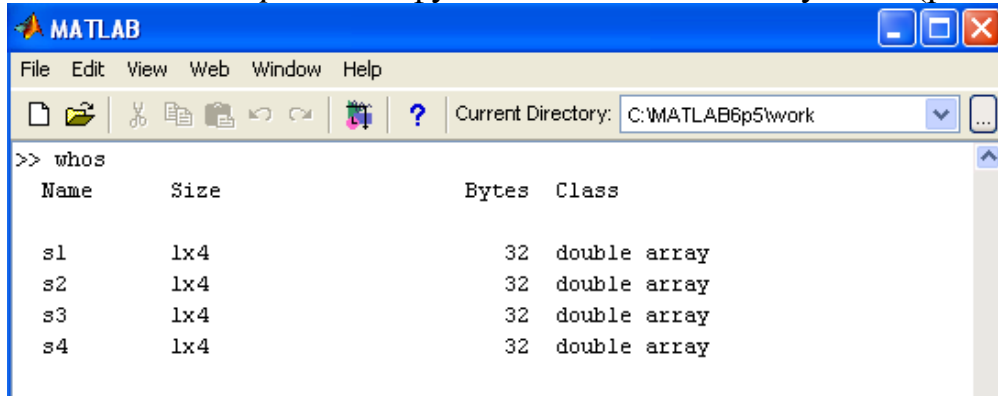
```
>> s2 = [5 3 3 2]
```

```
s2 =
```

```
5 3 3 2
```

```
>> s3 = s1 + s2
s3 =
    8 7 12 4
>> s4 = log(s3)
s4 =
2.0794 1.9459 2.4849 1.38634
```

Щоб з'ясувати, в яких масивах зберігаються вектор-рядки можна використати вікно *Workspace* або функцію *size* або команду *whos* (рис. 31):



```
>> whos
Name      Size      Bytes  Class
-----
s1        1x4        32    double array
s2        1x4        32    double array
s3        1x4        32    double array
s4        1x4        32    double array
```

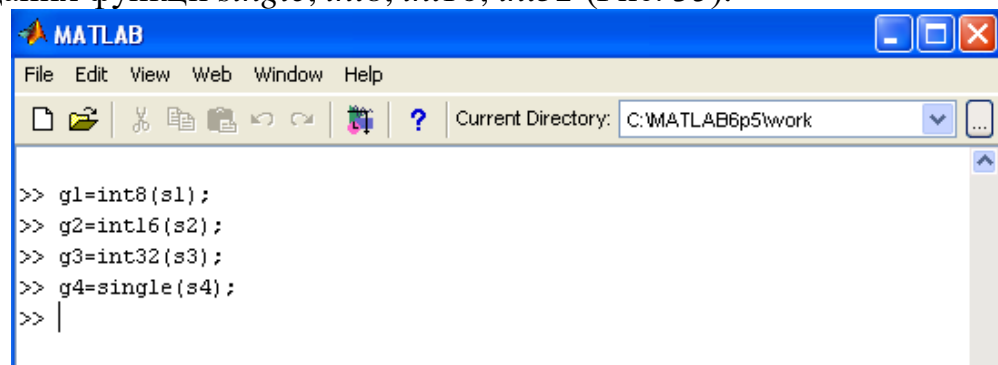
Рис. 34. Результат роботи команди *whos*

За допомогою команди *whos* виводиться інформація, що вектор-рядки *s1*, *s2*, *s3* і *s4* розміщуються в двомірних масивах розміром один на чотири. Для визначення довжини вектор-стовбців або вектор-рядків слугує вбудована функція *length*:

```
>> L = length(s1)
L =
    4
```

За замовчуванням усі числа (елементи масивів) зберігаються у форматі з подвійною точністю (*double*) і займають 8 байтів. Великі масиви потребують для зберігання значних об'ємів пам'яті. Для зменшення об'єму пам'яті, що займається масивами можна застосовувати інші способи зберігання елементів масиву: *single* для дійсних чисел, які потребують для розміщення 4 байта, і *int8*, *int16*, *int32* – для цілих чисел, які займають 1, 2 або 4 байта відповідно. Використання таких даних значно економить пам'ять і не впливає на функціональні можливості MATLAB.

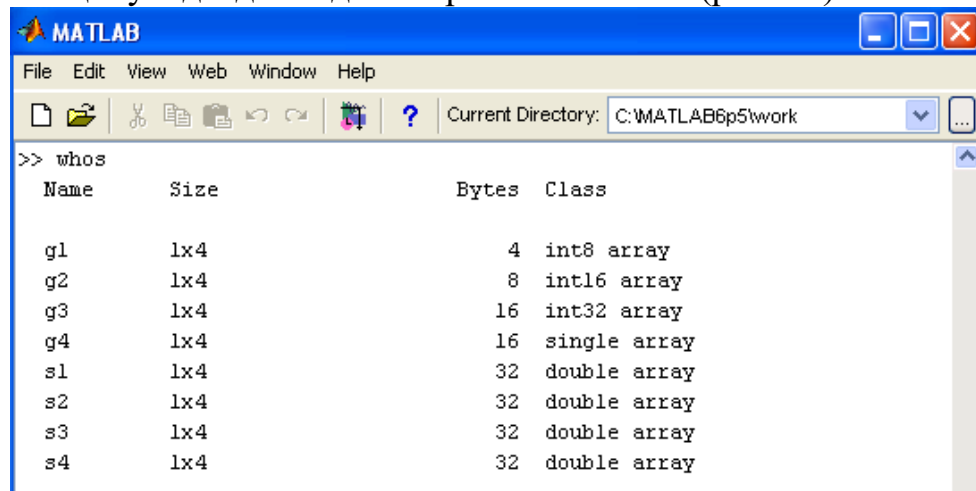
Для зміни точності представлення чисел призначені однойменні з типом (*Class*) даних функції *single*, *int8*, *int16*, *int32* (Рис. 35):



```
>> g1=int8(s1);
>> g2=int16(s2);
>> g3=int32(s3);
>> g4=single(s4);
>> |
```

Рис. 35. Використання функцій *single*, *int8*, *int16*, *int32*

Якщо тепер звернути увагу на розподіл пам'яті під масив, то видно суттєву різницю у відведеній для зберігання пам'яті (рис. 36):



```

>> whos

```

Name	Size	Bytes	Class
g1	1x4	4	int8 array
g2	1x4	8	int16 array
g3	1x4	16	int32 array
g4	1x4	16	single array
s1	1x4	32	double array
s2	1x4	32	double array
s3	1x4	32	double array
s4	1x4	32	double array

Рис. 36. Розподіл пам'яті під масиви отриманий командою whos

Для масивів великої розмірності економія пам'яті може бути суттєвою. Виконання арифметичних операцій з дійсними числами різного типу допустимо і дає результат з найменшою точністю – *single*:

```

>> qs44 = q4 + s4;
>> whos qs44

```

Name	Size	Bytes	Class
qs44	1x4	16	single array

Спроба виконати арифметичні операції над цілими числами різних типів призводить до помилки:

```

>> q31 = q3 + q1
??? Error using ==> plus

```

Integers can only be combined with integers of the same class, or scalar doubles.

Більш того, при перетворенні цілого числа, коли для точного представлення не вистачає пам'яті, що відводиться, отримуємо максимальне число для даного типу (тобто невірний результат) без попередження про це:

```

>> w1 = 34567;
>> r1 = int8(w1)
r1 =
    127
>> t1 = int16(w1)
t1 =
    32767

```

Зважаючи на це, застосування цілих чисел, які мають меншу точність представлення, обмежено. Продовжимо роботу з векторами і матрицями, використовуючи прийняте за замовчуванням представлення чисел *double*.

Із декількох вектор-стовбців можна скласти один, використовуючи квадратні дужки і розділяючи вхідні вектор-стовбці крапкою з комою:

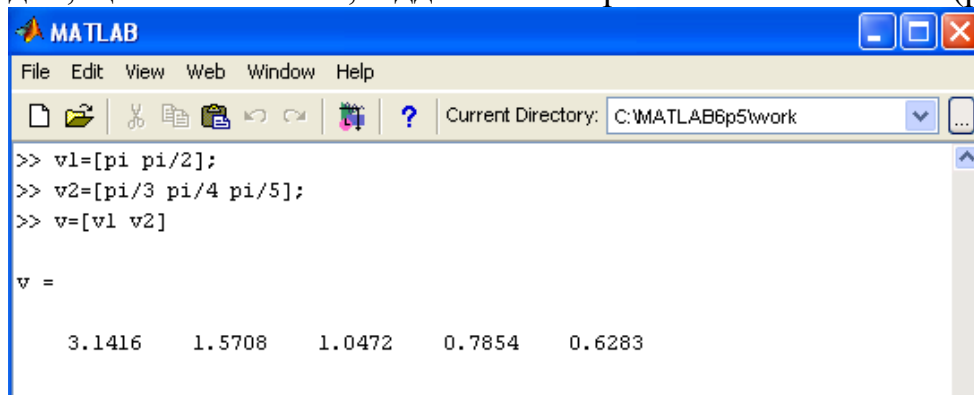
```

>> v1 = [1; 2];
>> v2 = [3; 4; 5];

```

```
>> v = [v1; v2]
v =
     1
     2
     3
     4
     5
```

Для зчеплення вектор-рядків також застосовуються квадратні дужки, але вектор-рядки, що зчіплюються, відділяються пробілами або комами (рис. 37):



```
MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>> v1=[pi pi/2];
>> v2=[pi/3 pi/4 pi/5];
>> v=[v1 v2]

v =

     3.1416     1.5708     1.0472     0.7854     0.6283
```

Рис. 37. Введення вектор-рядків та їх зчеплення

Для перегляду і змінювання значень елементів масивів зручно використовувати редактор масивів (*Array Editor*). Двічі клацнемо мишею по імені масиву  $v$  у вікні *Workspace* або натиснемо кнопку *Open Selection* на панелі інструментів вікна *Workspace* при положенні курсору на імені масиву – відкриється вікно редактора масивів з вмістимим вектор-рядку  $v$  (рис. 38).

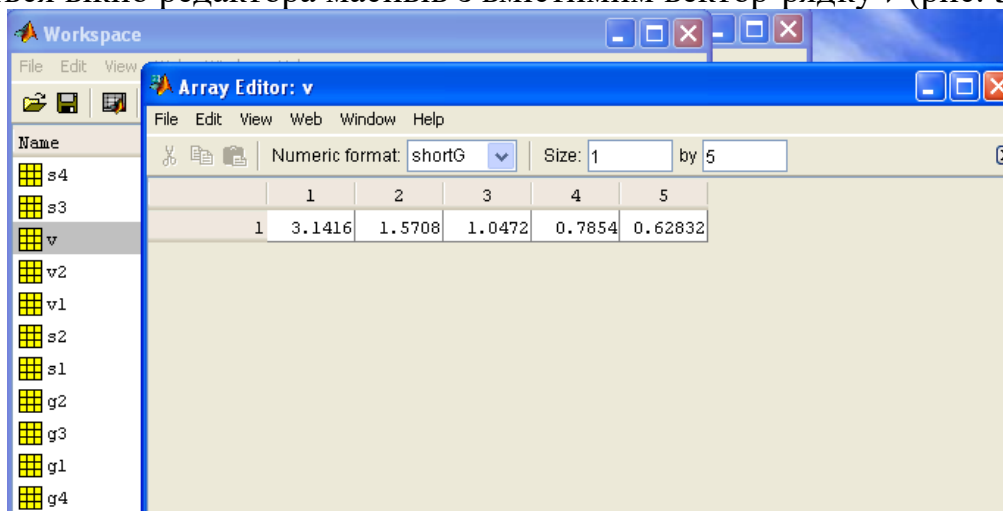


Рис. 38. Вектор-рядок  $v$  у вікні редактора масивів *Array Editor*

Ім'я масиву, що переглядається, вказано в рядку заголовка вікна. Подвійне клацання мишею по осередку або натиск на клавішу F2 (як і в таблицях MS Excel) дозволяє редагувати вміст певного елемента масиву. При заповненні пустих клітинок змінюються розміри масиву так, що він залишається прямокутним найменшого розміру и містить усі визначені елементи. При цьому не визначені користувачем елементи набувають нульового значення. Для зменшення розмірів необхідно використовувати

елемент управління *Delete* (в контекстному або віконному меню або на панелі інструментів), а не *Cut*, який призводить лише до обнуління елементів масиву.

Для роботи з даними редактор масивів MATLAB надає можливості, аналогічні MS Excel. Збільшимо довжину вектор-рядку  $v$  до 10, для чого виділимо перші п'ять її елементів, скопіюємо їх за допомогою спливаючого меню (або кнопки *Copy* панелі інструментів, або однойменного пункту меню *Edit*), зробимо поточним шостий осередок і використаємо вставку (*Paste*) для дублювання перших п'яти елементів вектор-рядку (рис.39).

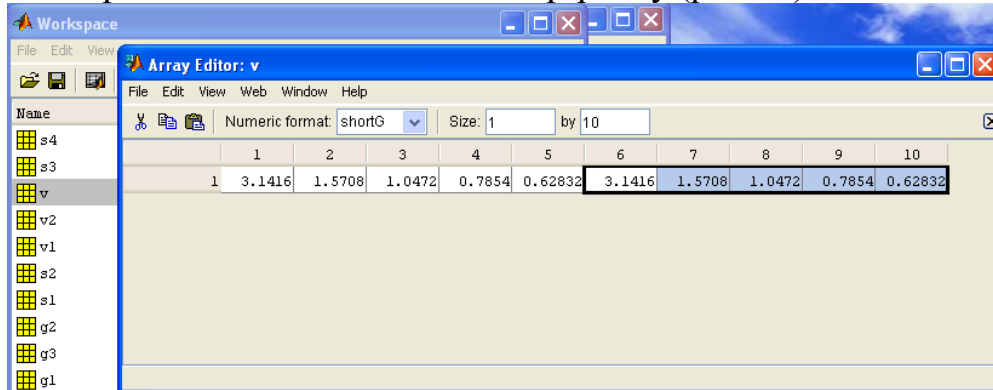


Рис. 39. Результат збільшення довжини вектор-рядку  $v$

За допомогою командного вікна перевіримо чи змінився вектор  $v$  (рис.40).

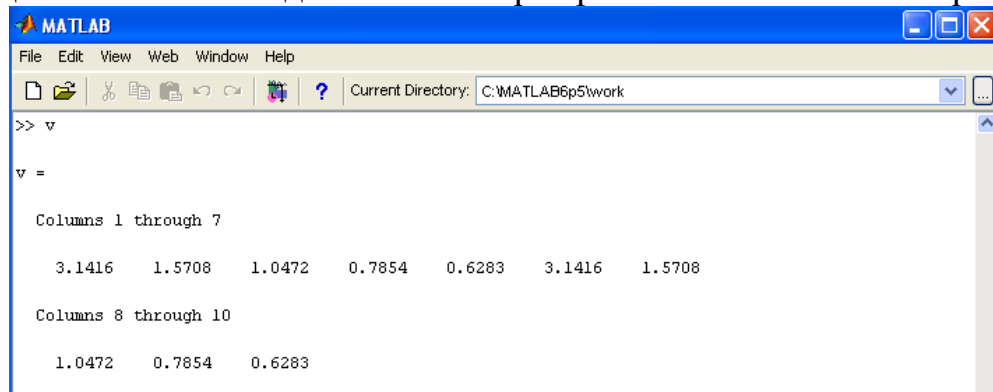


Рис. 40. Вікно Command Window зі збільшеною довжиною вектору  $v$

Редактор масивів дозволяє переглядати значення декількох змінних. Подвійне клацання по рядку з  $v1$  у вікні *Workspace* призводить до появи нового вікна в редакторі масивів. Кожне з вікон можна зробити активним за допомогою відповідної вкладки внизу вікна редактора або вибравши його назву в меню *Window* (рис. 41).

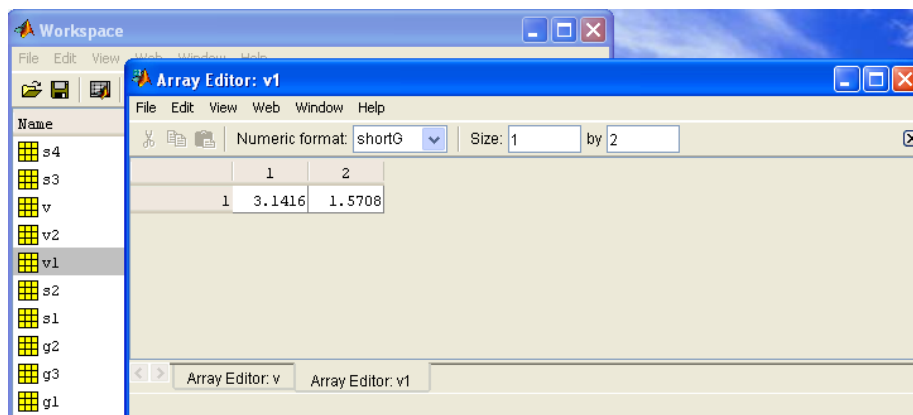


Рис. 41. Виклик нового вікна в редакторі масивів з вектор-рядком  $v1$

Кнопки в правій частині панелі інструментів відповідають різним способам розташування таблиць з наявним масивів у вікні редактора *Array Editor*. Також редактор масивів може бути корисним при налагодженні власних програм.

### Звернення до елементів вектора

Доступ до елементів вектору або вектор-рядку здійснюється за допомогою індексу, який береться в круглі дужки після імені масиву, в якому зберігається вектор. Якщо серед змінних робочого середовища є масив *v*, визначений вектор рядком:

```
>> v = [1.3 3.6 7.4 8.2 0.9];
```

то для звернення, наприклад, до четвертого елемента використовується *індексація*:

```
>> h = v(4)
```

```
h =
```

```
6.2000
```

До зміни масиву приводить вказівка на конкретний елемент масиву лівої частини оператора присвоювання:

```
>> v(2) = 555
```

```
v =
```

```
1.3000 555.0000 7.4000 8.2000 0.9000
```

Із елементів масиву можна формувати нові масиви, наприклад:

```
>> u = [v(3); v(2); v(1)]
```

```
u =
```

```
7.4000
```

```
555.0000
```

```
1.3000
```

Для розміщення відповідних елементів вектору в другий вектор в заданому порядку служить *індексація за допомогою вектора*. Наприклад, запис в вектор-рядку *w* четвертого, другого і п'ятого елементів *v* здійснюється наступним чином:

```
>> v = [1.3 555.0 7.4 8.2 0.9];
```

```
>> ind = [4 2 5];
```

```
>> w = v(ind)
```

```
w =
```

```
8.2000 555.000 0.9000
```

MATLAB надає зручний спосіб звернення до блоків послідовно розташованих елементів вектора або вектор-рядку. Для цього служить *індексація за допомогою знаку двокрапка*. Припустимо, що в заданому вектор-рядку *w* із семи елементів треба замінити нулями елементи з другого по шостий. Індксація за допомогою двокрапки дозволяє просто и наглядно вирішити цю задачу:

```
>> w = [7 9 2 5 1 8 3];
```

```
>> w(2:6) = 0
```

```
w =
```

```
7 0 0 0 0 0 3
```



Присвоювання  $w(2:6) = 0$  еквівалентно послідовності команд  $w(2) = 0$ ;  $w(3) = 0$ ;  $w(4) = 0$ ;  $w(5) = 0$ ;  $w(6) = 0$ ;

Індексацію за допомогою двокрапки зручно використовувати при виділенні частини даних з більшого об'єму даних в новий масив:

```
>> w = [7 9 2 5 1 8 3];
>> w1 = w(3:5)
w1 =
     2     5     1
```

Складемо вектор-рядок  $w2$ , який містить елементи  $w$  крім четвертого. Використаємо двокрапку і зчеплення рядків:

```
>> w2 = [w(1:3) w(5:7)]
w2 =
     7     9     2     1     8     3
```

Замість  $w(5:7)$  можна написати  $w(5:end)$ . Такий запис означає, що вибиратимуться елементи, починаючи з п'ятого і закінчуючи останнім. Якщо ж ввести просто  $w(end)$ , то отримаємо останній елемент масиву:

```
>> w(end)
ans =
     9.8000
```

Елементи масиву можуть входити у вирази. Обчислення, наприклад, середнього геометричного із елементів вектора  $u$  можна виконати наступним чином:

```
>> qn = (u(1)*u(2)*u(3))^(1/3)
qn =
    17.4779
```

Цей спосіб дуже зручний для довгих масивів.

### Застосування функцій обробки даних до векторів

Перемноження елементів вектор-стовбця або вектор-рядку здійснюється за допомогою функції *prod*:

```
>> z = [3; 2; 1; 4; 6; 5];
>> p = prod(z)
p = 720
```

Використовуючи цю функцію нескладно знайти середнє геометричне елементів вектору  $z$ :

```
>> gm = prod(z) ^ (1/length(z))
gm =
     2.9938
```

Функція *sum* призначена для знаходження суми елементів вектора:

```
>> s = sum(z)
s = 21
```

Обчислимо середнє арифметичне елементів вектора  $z$ :

```
>> sum(z) / length(z)
ans =
     3.5000
```

Перевіримо результат, обчислимо середнє арифметичне, використавши вбудовану функцію *mean*:

```
>> mean(z)
```

```
ans =
```

```
3.5000
```

Для знаходження мінімуму із елементів вектора служить функція *min*:

```
>> n = min(z)
```

```
n = 1
```

Для знаходження максимуму із елементів вектора служить функція *max*:

```
>> m = max(z)
```

```
m = 6
```

Часто виникає необхідність знати не тільки значення мінімального або максимального елемента в масиві, але і його індекс (порядковий номер). Інтерфейс функцій MATLAB достатньо універсальний. Більшість з функцій допускають звернення до них зі змінним числом вхідних і вихідних аргументів. Ознайомимося з роботою функції *min* з двома вихідними аргументами:

```
>> [m, k] = min(z)
```

```
m =
```

```
1
```

```
k =
```

```
3
```

В результаті змінній *m* буде присвоєно значення мінімального елемента масиву *z*, а номер мінімального елемента буде занесено в змінну *k*.

В число основних функцій для роботи з векторами входить функція упорядкування вектора по зростанню його елементів – функція *sort*:

```
>> z = [9.4 -2.3 -5.2 7.1 0.8 1.3];
```

```
>> R = sort(z)
```

```
R = -5.2000 -2.3000 0.8000 1.3000 7.1000 9.4000
```

Виконаємо упорядкування вектора по зменшенню його елементів використовуючи функцію *sort*:

```
>> R1 = -sort(-z)
```

```
R1 = 9.4000 7.1000 1.3000 0.8000 -2.3000 -5.2000
```

В число вбудованих функцій входять: дискретне перетворення Фур'є – *fft*, згортка – *conv*, робота із звуком – *sound*. Інформацію про функції обробки даних можна узнати, набравши в командному рядку команду *help datafun* або звернувшись до пункту *Data Analysis and Fourier Transforms* підрозділу *Mathematics* розділу *Functions – Categorical List* інтерактивної довідкової системи MATLAB.

### Поелементні операції з векторами

В розглянутому раніше матеріалі вектор використовувався в якості аргументу математичних функцій, результатом яких був вектор з елементами, які дорівнюють значенням функції від відповідних елементів вхідного вектора. Таким чином, відбувалося *поелементне* обчислення функції яка викликається. Далі розглянемо можливості поелементної роботи з векторами, які

знадобляться в подальшому для визначення власних функцій і побудови їх графіків.

Поелементні операції з векторами:

Введемо два вектор-рядки

```
>> v1 = [2 -3 4 1];
```

```
>> v2 = [7 5 -6 9];
```

Операція `.*` (перед знаком множення повинна бути крапка) приводить до поелементного множення векторів однакової довжини. В результаті отримуємо вектор з елементами, які дорівнюють добутку відповідних елементів вхідних векторів:

```
>> u = v1.*v2
```

```
u = 14 -15 -24 9
```

За допомогою `.^` здійснюється поелементне піднесення до степеню:

```
>> p = v1.^2
```

```
p = 4 9 16 1
```

Показником степеня може бути вектор тієї ж довжини, що і зведений до степеня. При цьому кожен елемент першого вектора підноситься до степеню, який дорівнює відповідному елементу другого вектора:

```
>> p = v1.^v2
```

```
P = 128.0000 -243.0000 0.0002 1.0000
```

Ділення відповідних елементів векторів однакової довжини виконується з використанням `./`

```
>> d = v1./v2
```

```
d = 0.2857 -0.6000 -0.6667 0.1111
```

Зворотнє поелементне ділення (ділення елементів 2-го вектора на відповідні елементи 1-го) здійснюється за допомогою `.\`

```
>> dinv = v1.\v2
```

```
dinv = 3.5000 -1.6667 -1.5000 9.0000
```

Таким чином, крапка в MATLAB використовується не тільки для введення десяткових дробів, але і для вказівки на те, що ділення або множення або зведення до степеню масивів однакового розміру повинно бути виконано поелементно.

До поелементних операцій відносяться і операції з вектором і числом. Складання вектора і числа не приводить до повідомлення про помилку. MATLAB прибавляє число до кожного елемента вектора. Таке ж твердження відноситься і для віднімання.

Додавання числа до кожного елемента вектора:

```
>> v = [4 6 8 10];
```

```
>> s = v+1.2
```

```
s = 5.2000 7.2000 9.2000 11.2000
```

Додавання до кожного елемента вектора числа:

```
>> v = [4 6 8 10];
```

```
>> s1 = 1.2+v
```

```
s1 = 5.2000 7.2000 9.2000 11.2000
```

Віднімання числа з кожного елемента вектора

```
>> v = [4 6 8 10];
>> r1 = v-1.2
r1 = 2.8000 4.8000 6.8000 8.8000
Віднімання із числа кожного елемента вектора
>> r2 = 1.2-v
r2 = -2.8000 -4.8000 -6.8000 -8.8000
```

Множити вектор на число можна як справа, так і зліва:

```
>> v = [4 6 8 10];
>> p1 = v*2
p1 = 8 12 16 20
>> p2 = 2*v
p2 = 8 12 16 20
```

Ділити за допомогою знака / можна вектор на число:

```
>> d = v/2
d = 2 3 4 5
```

Спроба ділення числа на вектор-рядок приводить до повідомлення про помилку:

```
>> p = 2/v
??? Error using == > /
Matrix dimensions must agree.
```

Якщо потрібно розділити число на кожен елемент вектора і записати результат в новий вектор, то слід використати операцію ./

```
>> w = [4 2 6];
>> d1 = 12./w
d1 = 3 6 2
```

Усі розглянуті операції можуть застосовуватися як до вектор-рядків, так в до вектор-стовбців.

### Добуток векторів

Вектор можна множити на другий вектор скалярно (це множення ще називають внутрішнім), векторно, або здійснити зовнішнє множення. Результатом скалярного добутку являється число, векторного – вектор, а зовнішнього – матриця.

### Скалярний добуток

Скалярний добуток векторів  $a$  і  $b$  довжини  $N$ , які складаються із дійсних чисел, визначається формулою:  $a \cdot b = \sum_{k=1}^N a_k b_k$ . Отже, для обчислення скалярного добутку необхідно скласти в суму компоненти вектора, отриманого в результаті поелементного множення  $a$  на  $b$ , тобто необхідно використати функцію *sum* і поелементне множення. Знайдемо скалярний добуток векторів:

```
a = [ 1.2
     -3.2
      0.7 ]; b = [ 4.1
                  6.5
                 -2.9 ]
Введемо послідовність необхідних команд:
>> a = [1.2; -3.2; 0.7];
```

```
>> b = [4.1; 6.5; -2.9];
>> s = sum(a.*b)
s =
-17.9100
```

Скалярний добуток векторів можна також обчислити, використавши функцію MATLAB *dot*:

```
>> s = dot(a, b);
s =
-17.9100
```

Знайти довжину (або модуль) вектора  $a$ :  $|a| = \sqrt{a \cdot a}$ .

Введемо команди для рішення:

```
>> d = sqrt(dot(a, b))
d =
3.4886
```

### Векторний добуток

Векторний добуток  $a \times b$  визначений тільки для векторів із трьохвимірного простору, тобто які складаються із трьох елементів. Результатом також являється вектор із трьохвимірного простору. Для обчислення векторного добутку в MATLAB існує функція *cross*:

```
>> a = [1.2; -3.2; 0.7];
>> b = [4.1; 6.5; -2.9];
>> c = cross(a, b)
c =
4.7300
6.3500
20.9200
```

Змішаний добуток векторів  $a, b, c$  визначається формулою  $abc = a \cdot (b \times c)$ . Модуль змішаного добутку векторів дорівнює об'єму паралелепіпеда. Знайдемо об'єм паралелепіпеда, якщо

$$a = \begin{bmatrix} 3.5 \\ 0 \\ 0 \end{bmatrix}; \quad b = \begin{bmatrix} 0.5 \\ 2.1 \\ 0 \end{bmatrix}; \quad c = \begin{bmatrix} -0.2 \\ -1.9 \\ 2.8 \end{bmatrix}.$$

Для виконання завдання необхідно ввести команди:

```
>> a = [3.5; 0; 0];
>> b = [0.5; 2.1; 0];
>> c = [-0.2; -1.9; 2.8];
>> v = abs(dot(a, cross(b, c)))
v =
20.5800
```

### Зовнішній добуток

Зовнішнім добутком векторів  $a = (a_j)_{j=1, \dots, N}$ ,  $b = (b_k)_{k=1, \dots, M}$  називається матриця  $C = (c_{jk})_{j=1, \dots, N, k=1, \dots, M}$  розміром  $N \times M$ , елементи якої обчислюються за формулою  $c_{jk} = a_j b_k$ . Вектор-стовбець  $a$  в MATLAB представляється у вигляді двовимірного

масиву розміру  $N$  на один. Вектор-стовбець  $b$  при транспонуванні переходить у вектор-рядок розміру один на  $M$ . Вектор-стовбець і вектор-рядок є матриці, у яких один із розмірів дорівнює одиниці. Фактично  $C = ab^{\circ}$ , де множення відбувається по правилу матричного добутку. Для обчислення матричного добутку в MATLAB використовується оператор «зірочка»:

```
>> a = [1; 2; 3];
>> b = [5; 6; 7];
>> c = a*b'
c =
    5    6    7
   10   12   14
   15   18   21
```

MATLAB вивела в командне вікно матрицю у звичайному вигляді – по рядкам. Використаємо команду *whos* для перегляду змінних робочого середовища (рис. 42), зверніть увагу на змінну *c*.

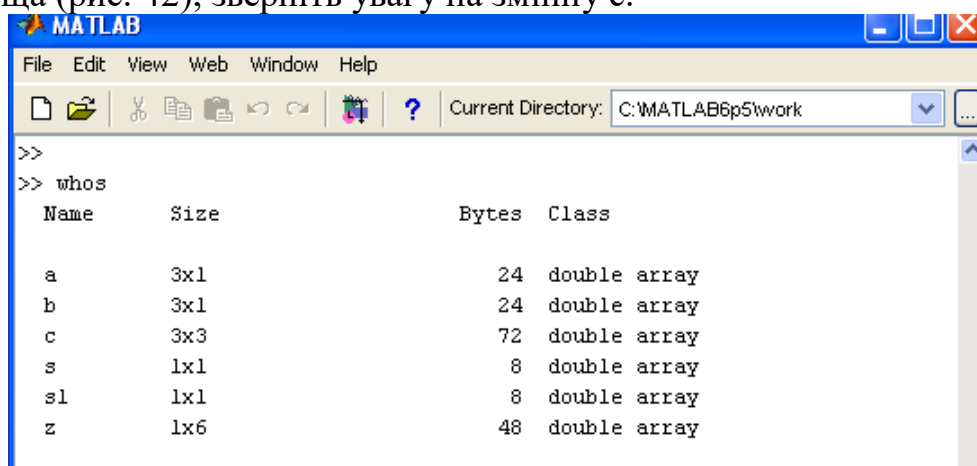


Рис. 42. Вікно Command Window з інформацією про змінні робочого середовища

Командами View – Workspace відкриємо вікно Workspace. Установивши курсор на змінній *c* виконаємо подвійне клацання лівою кнопкою миші. Таким чином викличемо вікно редактора масивів Array Editor (рис. 43), в якому представлена інформація про змінну *c*.

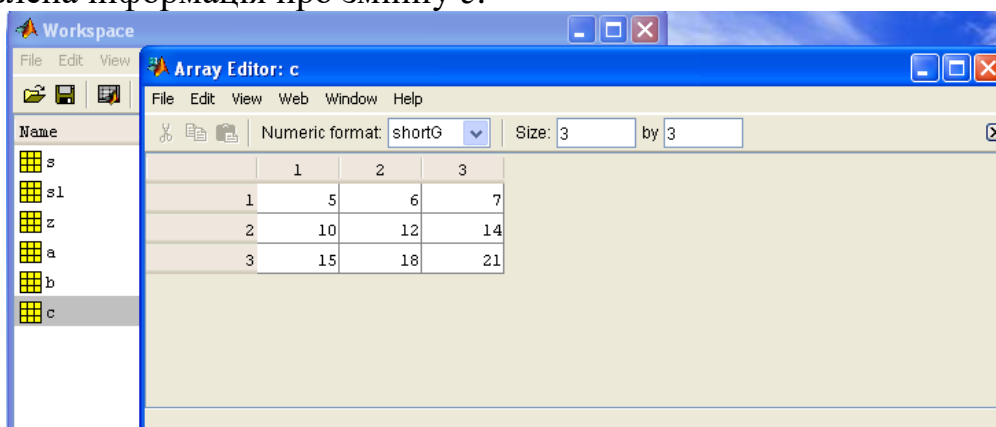


Рис. 43. Вікно редактора масивів Array Editor

Числа, вектори і матриці зберігаються в двовірних масивах: числа – в масивах, розмірністю один на один, вектор-стовбці і вектор-рядки містяться в

масивах, у яких один із вимірів дорівнює одиниці, а для матриць виділяються двовірні масиви підходящих розмірів. Завдяки цьому операції і вбудовані функції в MATLAB пристосовуються до виду аргументів, видаючи результат у відповідному вигляді.

Приклад: Задано довільний вектор. Переставити елементи вектора у порядку зростання. Знайти суму і добуток додатних елементів вектора. Відсортований масив повернути на  $90^\circ$  проти часової стрілки.

Зразок виконання завдання:

1) Ввести довільний вектор:

```
>> v = [-1 3 -5 0 -2 0 7 0 5 -7];
```

2) Відсортувати його у порядку зростання:

```
>> vn = sort(v)
```

```
vn =
```

```
    -7    -5    -2    -1     0     0     0     3     5     7
```

3) Обчислити суму додатних елементів вектора:

```
>> vs = sum(vn(vn>0))
```

```
vs =
```

```
    15
```

4) Обчислити добуток додатних елементів вектора:

```
>> vp = prod(vn(vn>0))
```

```
    105
```

5) Застосувати функцію *rot90*, яка повертає масив на  $90$  градусів проти часової стрілки.

```
>> pov = rot90(vn)
```

```
     7
```

```
     5
```

```
     3
```

```
     0
```

```
     0
```

```
     0
```

```
    -1
```

```
    -2
```

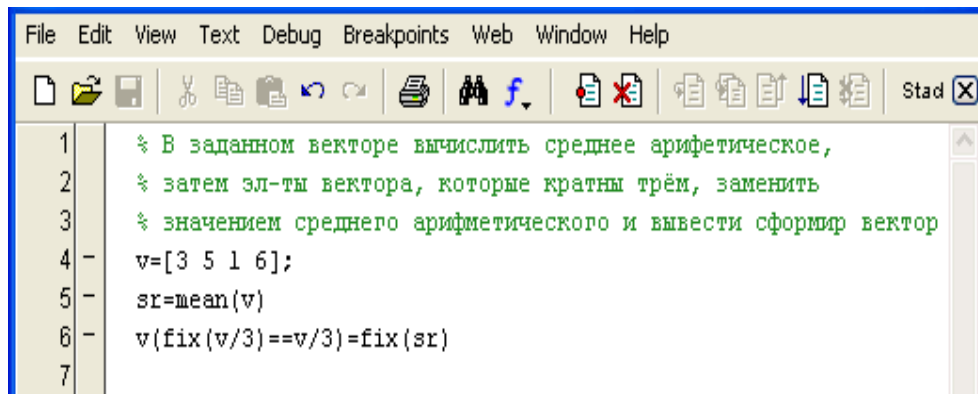
```
    -5
```

```
    -7
```

### Зразок виконання завдань

**Завдання 1.** У довільно заданому векторі обчислити середнє арифметичне, потім елементи вектору, які кратні 3 замінити значенням середнього арифметичного елементів вектору. Вивести значення середнього арифметичного елементів вектору і вивести сформований вектор.

Складемо програму для реалізації завдання, наберемо текст файл-програми в редакторі М-файлів.

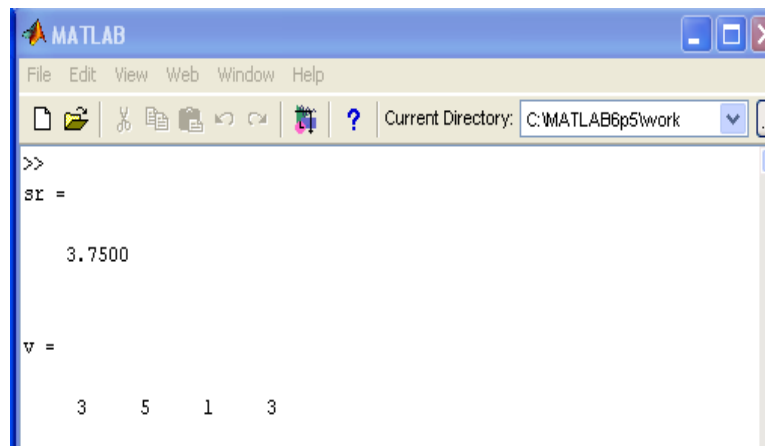


```

File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stad X
1 % В заданном векторе вычислить среднее арифметическое,
2 % затем эл-ты вектора, которые кратны трём, заменить
3 % значением среднего арифметического и вывести сформир вектор
4 - v=[3 5 1 6];
5 - sr=mean(v)
6 - v(fix(v/3)==v/3)=fix(sr)
7

```

На екрані з'явиться результат. Обчислене значення середнього арифметичного знаходиться у змінній *sr*, у змінній *v* знаходиться сформований вектор.



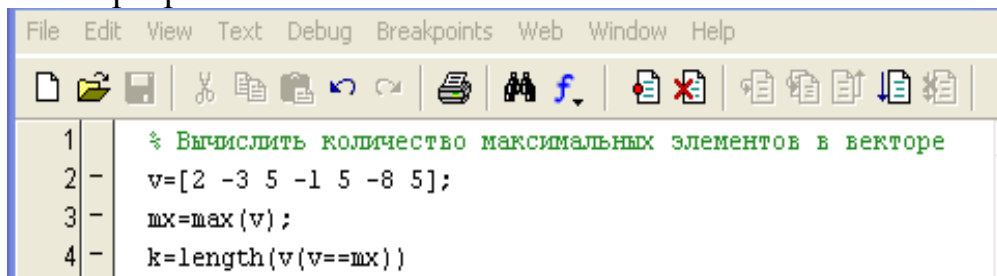
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
sr =
    3.7500
v =
    3     5     1     3

```

**Завдання 2.** У довільно заданому векторі обчислити кількість максимальних елементів у векторі.

Лістинг програми:

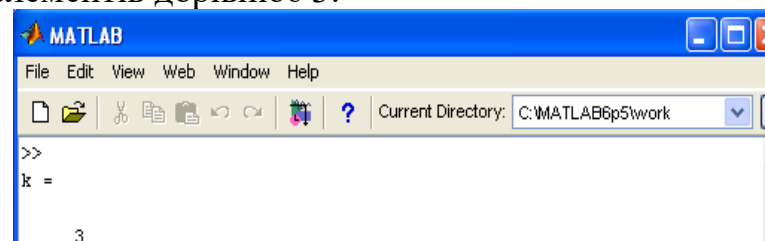


```

File Edit View Text Debug Breakpoints Web Window Help
[Icons]
1 % Вычислить количество максимальных элементов в векторе
2 - v=[2 -3 5 -1 5 -8 5];
3 - mx=max(v);
4 - k=length(v(v==mx))

```

Результат обчислення. У довільно заданому векторі кількість максимальних елементів дорівнює 3.



```

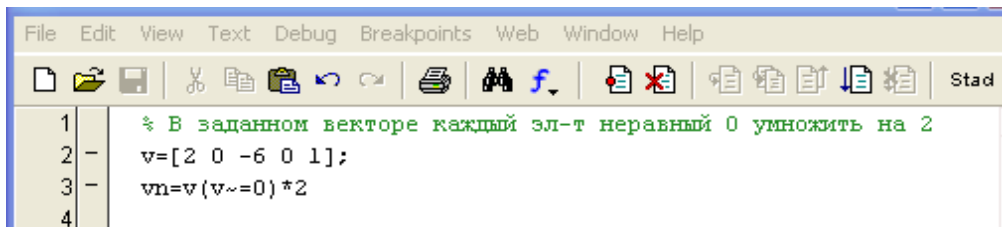
MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
k =
    3

```

**Завдання 3.** У довільно заданому векторі кожен елемент, значення якого не дорівнює нулю, помножити на 2. Сформований вектор вивести.

Лістинг програми:



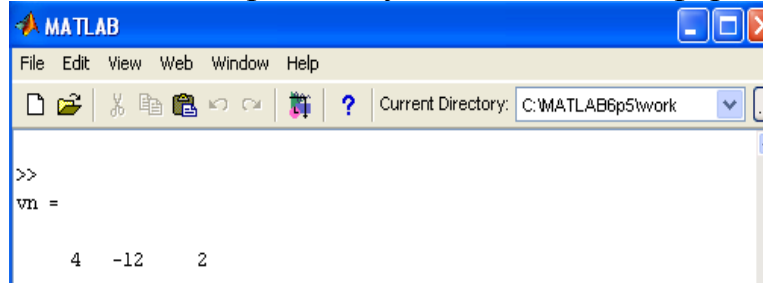


```

1   % В заданном векторе каждый эл-т неравный 0 умножить на 2
2   v=[2 0 -6 0 1];
3   vn=v(v~=0)*2
4

```

Результат обчислення. У робочому вікні виведено сформований вектор.



```

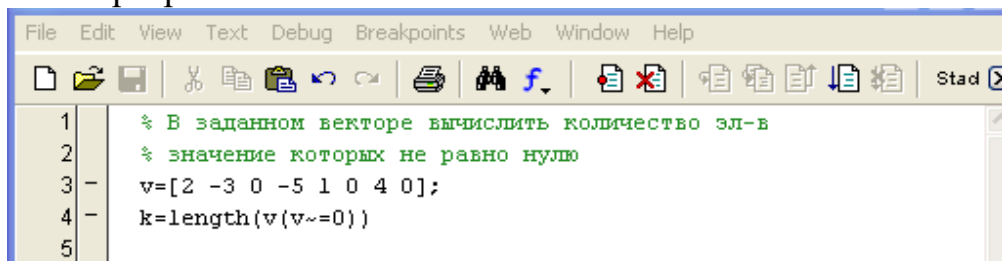
>>
vn =

     4    -12     2

```

**Завдання 4.** У довільно заданому векторі обчислити кількість елементів, значення яких не дорівнює нулю.

Лістинг програми:

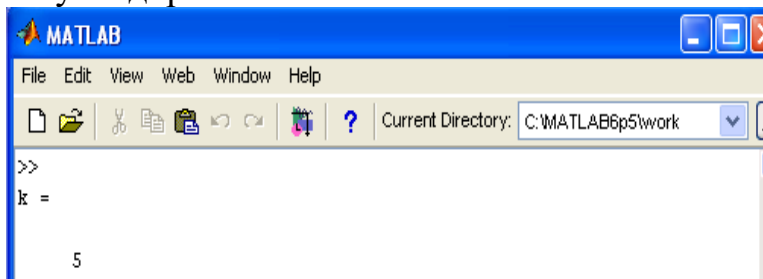


```

1   % В заданном векторе вычислить количество эл-в
2   % значение которых не равно нулю
3   v=[2 -3 0 -5 1 0 4 0];
4   k=length(v(v~=0))
5

```

Результат обчислення. У заданому векторі кількість елементів, значення яких не дорівнює нулю дорівнює 5.



```

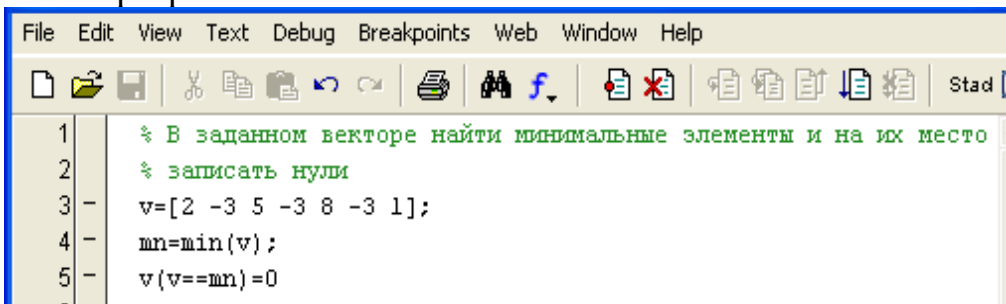
>>
k =

     5

```

**Завдання 5.** У довільно заданому векторі знайти максимальні елементи і на їх місце записати нулі. Вивести сформований вектор.

Лістинг програми:

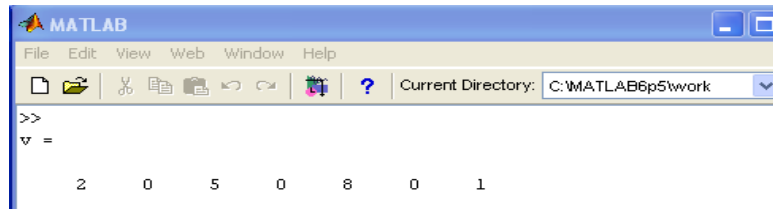


```

1   % В заданном векторе найти минимальные элементы и на их место
2   % записать нули
3   v=[2 -3 5 -3 8 -3 1];
4   mn=min(v);
5   v(v==mn)=0
6

```

Результат обчислення. У робочому вікні виведено вектор, сформований згідно умови.



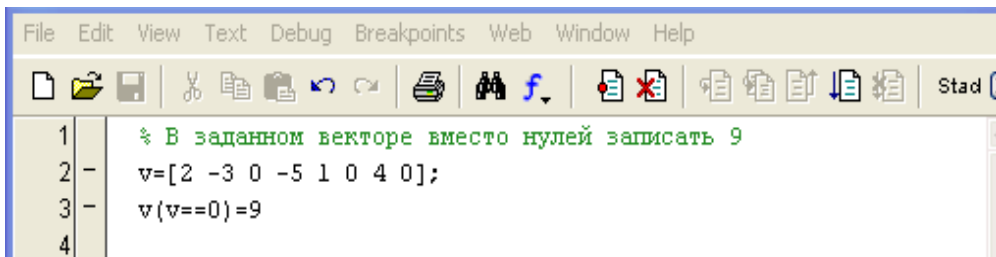
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
v =
     2     0     5     0     8     0     1

```

**Завдання 6.** У довільно заданому векторі замість нулів записати число 9. Сформований вектор вивести.

Лістинг програми:

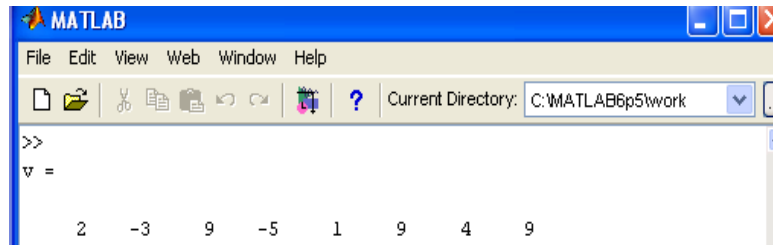


```

File Edit View Text Debug Breakpoints Web Window Help
1 % В заданном векторе вместо нулей записать 9
2 - v=[2 -3 0 -5 1 0 4 0];
3 - v(v==0)=9
4

```

Результат обчислення. У робочому вікні виведено вектор, сформований згідно умови.



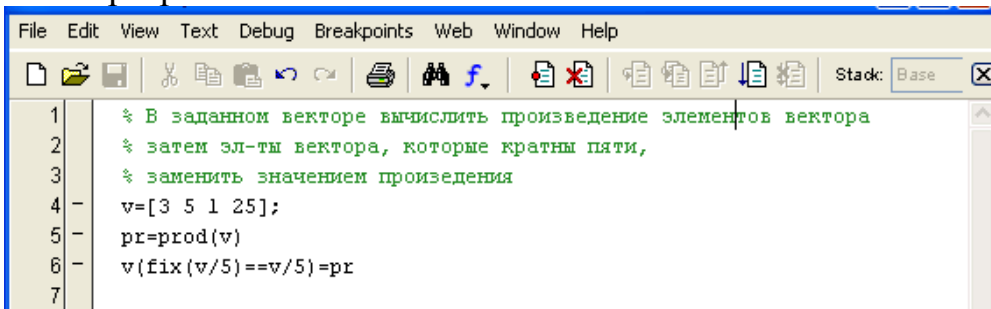
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
v =
     2    -3     9    -5     1     9     4     9

```

**Завдання 7.** У довільно заданому векторі обчислити добуток елементів вектору. Потім елементи вектору, які кратні 5, замінити значенням добутку елементів вектору. Вивести добуток елементів вектору та новий сформований вектор.

Лістинг програми:

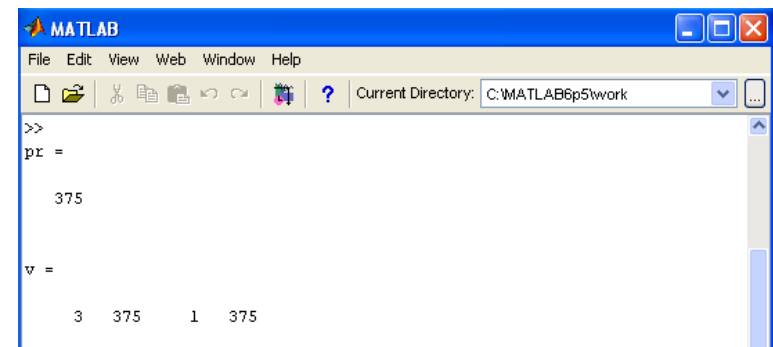


```

File Edit View Text Debug Breakpoints Web Window Help
1 % В заданном векторе вычислить произведение элементов вектора
2 % затем эл-ты вектора, которые кратны пяти,
3 % заменить значением произведения
4 - v=[3 5 1 25];
5 - pr=prod(v)
6 - v(fix(v/5)==v/5)=pr
7

```

Результат обчислення. У робочому вікні виведено вектор, сформований згідно умови.



```

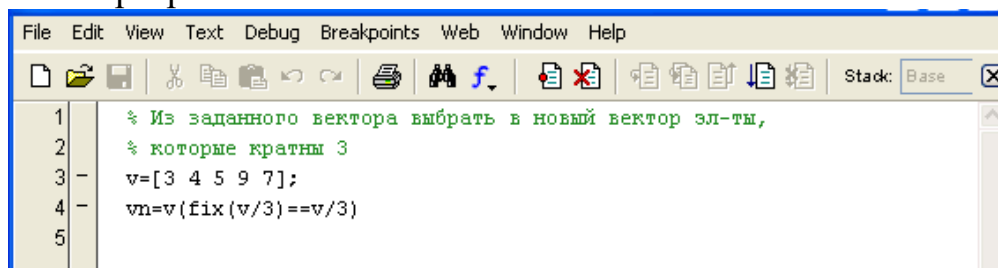
MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
pr =
    375

v =
     3   375     1   375

```

**Завдання 8.** З довільно заданих елементів вектору сформувати вектор, в який вибрати елементи, значення яких кратне 3.

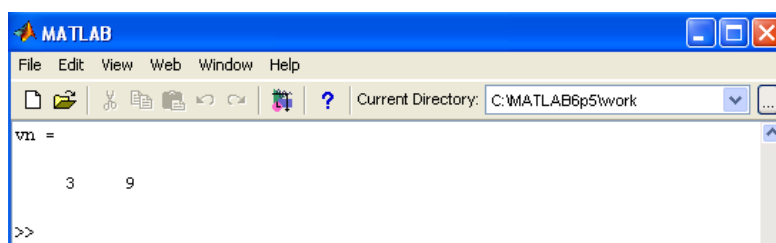
Лістинг програми:



```

File Edit View Text Debug Breakpoints Web Window Help
% Из заданного вектора выбрать в новый вектор эл-ты,
% которые кратны 3
v=[3 4 5 9 7];
vn=v(fix(v/3)==v/3)
  
```

Результат обчислення. У робочому вікні виведено вектор, сформований згідно умови.



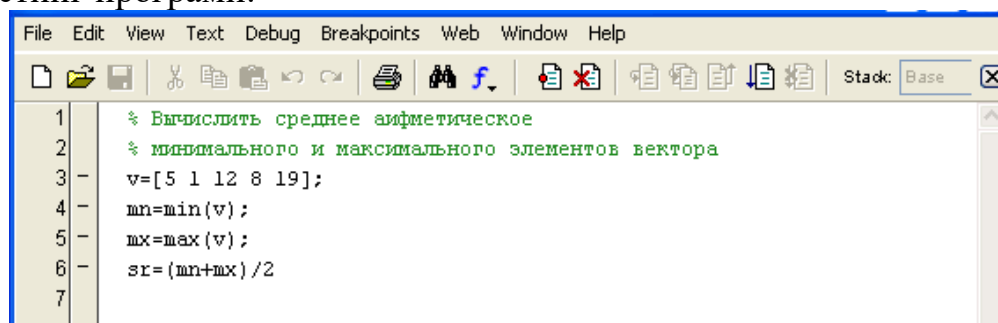
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
vn =

     3     9
>>
  
```

**Завдання 9.** У довільно заданому векторі визначити максимальний і мінімальний елементи. Знайти середнє арифметичне цих елементів і вивести цей результат.

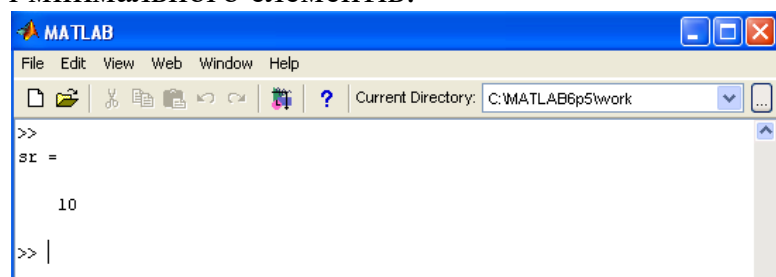
Лістинг програми:



```

File Edit View Text Debug Breakpoints Web Window Help
% Вычислить среднее арифметическое
% минимального и максимального элементов вектора
v=[5 1 12 8 19];
mn=min(v);
mx=max(v);
sr=(mn+mx)/2
  
```

Результат обчислення. У робочому вікні виведено середнє арифметичне максимального і мінімального елементів.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
>>
sr =

     10
>> |
  
```

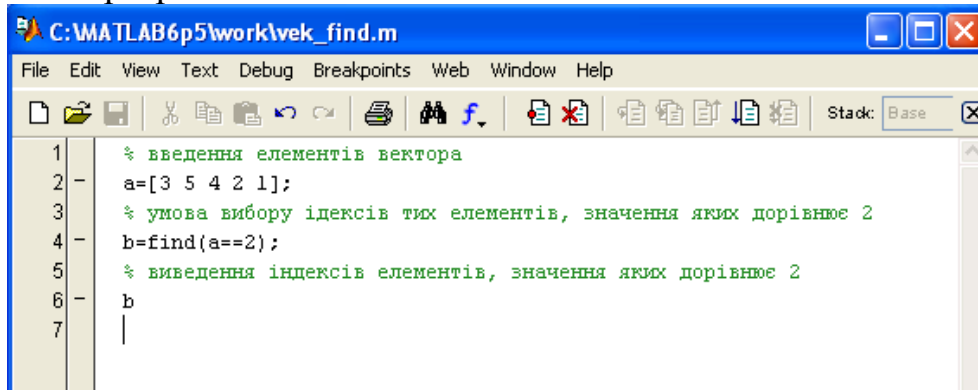
### Функція find()

У завданнях інколи бувають випадки, коли потрібно знайти певний елемент у векторі чи матриці. Це можна виконати за допомогою стандартної функції `find()`, яка в якості аргументу приймає умову, відповідно до якої знаходяться необхідні елементи.

Розглянемо роботу функції `find()` де як аргумент використовується вектор.

**Завдання 10.** У довільно заданому векторі вибрати і вивести індекси тих елементів, значення яких дорівнює 2.

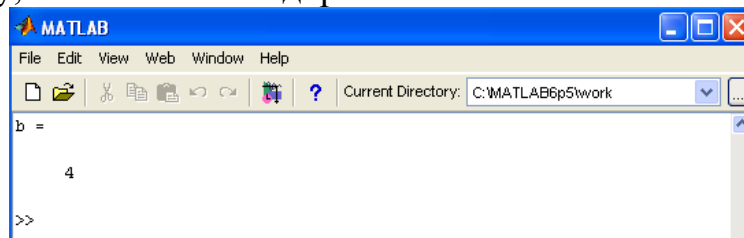
Лістинг програми:



```

C:\MATLAB6p5\work\vek_find.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
1 % введення елементів вектора
2 a=[3 5 4 2 1];
3 % умова вибору ідексів тих елементів, значення яких дорівнює 2
4 b=find(a==2);
5 % виведення ідексів елементів, значення яких дорівнює 2
6 b
7 |
  
```

Результат обчислення. У робочому вікні виведено індекс елемента із заданого вектору, значення якого дорівнює 2.



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work
b =
    4
>>
  
```

## Індивідуальні завдання за темою «Робота з векторами. Обробка даних в векторах»

### Варіанти індивідуальних завдань

Задано довільний вектор.

1. Переставити елементи вектора в зворотному порядку, використовуючи індксацію вихідного вектора, і записати результат в новий вектор.
2. Виділити в новий вектор елементи із заданого вектора з парними номерами. Обчислити добуток трьох останніх елементів нового вектора.
3. У заданому векторі знайти суму тільки позитивних елементів вектора.
4. Замінити всі мінімальні елементи вектора максимальним значенням його елементів.
5. Знайти кількість позитивних і кількість негативних елементів вектора.
6. Обчислити суму всіх елементів вектора з непарними ідексами.
7. У заданому векторі знайти максимальний елемент вектора і його індекс.
8. Обчислити добуток елементів вектора, що стоять на непарних позиціях.
9. Впорядкувати елементи вектора по спадаючій і в отриманому векторі обчислити суму елементів більших за 2.
10. Обчислити кількість елементів вектора, значення яких дорівнює 5.
11. Замінити всі негативні елементи вектора мінімальним значенням його елементів.

12. Обчислити середнє арифметичне елементів вектора.
13. Знайти суму тільки негативних елементів вектора.
14. Обчислити суму всіх елементів вектора з парними індексами.
15. У вихідному векторі обчислити суму позитивних елементів, потім обчислити корінь квадратний з суми
16. Обчислити добуток елементів вектора, що стоять на парних позиціях. Визначити середнє арифметичне знайденого добутку і довжини вектора.
17. Впорядкувати елементи вектора в порядку зростання їх модулів.
18. Замінити всі позитивні елементи вектора максимальним значенням його елементів.
19. Впорядкувати елементи вектора за зростанням і в отриманому векторі обчислити кількість негативних елементів.
20. У вихідному векторі з усього обсягу даних обчислити середнє геометричне першого і останнього елементів вектора.
21. Задано вектор, який містить кілька елементів рівних 1. Обчислити кількість цих елементів і на їх місце поставити число 7.
22. Замінити всі максимальні елементи вектора мінімальним значенням його елементів.
23. Обчислити середнє арифметичне елементів вектора. Замінити всі негативні елементи значенням середнього арифметичного.
24. Обчислити середнє геометричне елементів вектора. Замінити елементи, значення яких дорівнює нулю, значенням середнього арифметичного.
25. Замінити елементи вектора, що відрізняються від середнього геометричного його елементів більш ніж на 10%, на середнє геометричне.

### **Двомірні масиви, матриці**

MATLAB зберігає елементи матриці в пам'яті по стовбцях.

При додаванні і відніманні матриці повинні бути одного розміру.

При перемноженні матриць число стовбців першої матриці повинно дорівнювати числу рядків другої матриці.

Додавання і віднімання матриць так же як чисел і векторів, здійснюється за допомогою знаків плюс і мінус.

Множення матриць здійснюється за допомогою знака \*.

### **Введення матриць, прості операції**

Існують різні способи введення. Вводити невеликі за розміром матриці зручно прямо з командного рядка. Введемо матрицю  $A$  розміром два на три:

$A = \begin{pmatrix} 3 & 1 & -1 \\ 2 & 4 & 3 \end{pmatrix}$ . Для зберігання матриці використовується двомірний масив з

іменем  $A$ . Введемо масив  $A$  у командному рядку:

```
>> A = [3 1 -1; 2 4 3]
```

```
A =
     3     1    -1
     2     4     3
```

Введемо квадратну матрицю  $B$  розміру три на три:

$B = \begin{pmatrix} 4 & 3 & -1 \\ 2 & 7 & 0 \\ -5 & 1 & 2 \end{pmatrix}$ . В командному рядку наберемо команди:

```
>> B = [4 3 -1
        2 7 0
        -5 1 2]
```

Набравши перший рядок натиснемо клавішу *Enter*. Курсор мигає на наступному рядку без символу  $>>$ . Продовжимо введення матриці по рядкам, натискаючи у кінці кожного рядку *Enter*. Останній рядок завершимо закриваючою квадратною дужкою. Отримаємо:

```
B =
     4     3    -1
     2     7     0
    -5     1     2
```

Ще один спосіб введення матриць полягає в тому, що матрицю можна трактувати як вектор-рядок, кожен елемент якого являється вектор-стовбцем.

Введемо матрицю два на три:  $C = \begin{pmatrix} 3 & -1 & 7 \\ 4 & 2 & 0 \end{pmatrix}$ . Використаємо команди:

```
>> C = [[3; 4] [-1; 2] [7; 0]]
C =
     3    -1     7
     4     2     0
```

Переглянемо змінні робочого середовища у вікні *Workspace* (рис. 44)

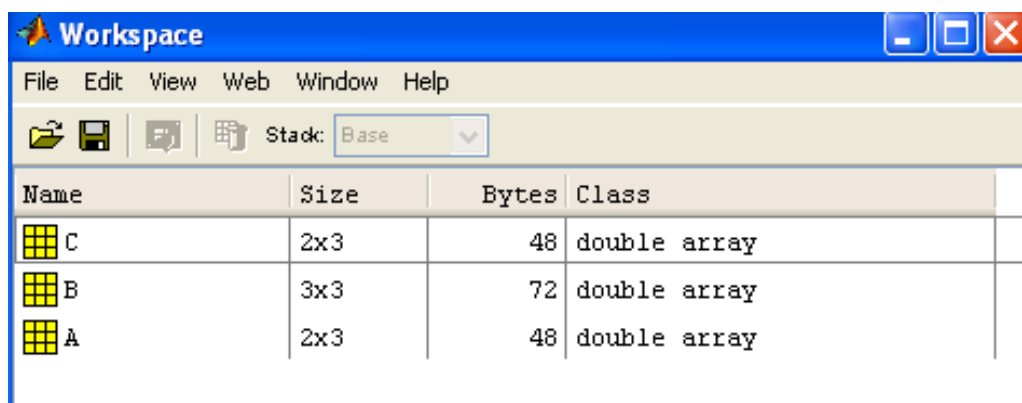


Рис. 44. Інформація про змінні у вікні *Workspace*

Або набравши у командному рядку *whos* (рис. 44)

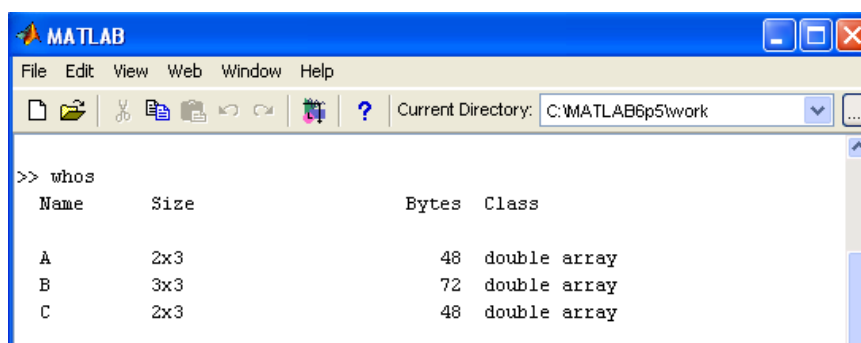


Рис. 44. Інформація про змінні у робочому вікні

### Звернення до елементів матриці

Доступ до елементів матриці здійснюється за допомогою двох індексів – номеру рядку і номеру стовпця, записаних в круглих дужках, і розділених між собою комою, наприклад:

```
>> C(2, 3)
```

```
ans =  
0
```

Елементи матриць можуть входити в склад виразів:

```
>> C(1, 1)+C(2, 2)+C(2, 3)
```

```
ans =  
5
```

В якості індексів можуть виступати вектори, які містять номери потрібних рядків і стовпців. Наприклад, для виокремлення елементів першого і другого рядків другого і третього стовпців введеної вище матриці  $B$  достатньо ввести команди:

```
>> i = [1 2];
```

```
>> j = [2 3];
```

```
>> B1 = B(i, j)
```

```
B1 =  
3 -1  
7 0
```

Розташування елементів матриці в пам'яті комп'ютера визначає ще один спосіб звернення до них. MATLAB зберігає елементи матриці в пам'яті по стовбцям. Елементи  $q_{ij}$  матриці  $Q$  розміру  $m$  на  $n$  містяться в пам'яті в послідовності:  $q_{11}, q_{21}, \dots, q_{m1}, q_{12}, q_{22}, \dots, q_{m2}, \dots, q_{1n}, q_{2n}, \dots, q_{mn}$ . Таким чином, для доступу до елементів матриці можна використовувати *один індекс*, який задає порядковий номер елемента матриці у векторі. Наприклад, елементи матриці  $C$ , яка введена вище, записані у такому порядку:  $C(1, 1), C(2, 1), C(1, 2), C(2, 2), C(1, 3), C(2, 3)$ .

Тому звернення до елементів матриці як до елементів вектора за допомогою одного індексу (індексація за допомогою порядкового номера) призводить до передбачуваного результату:

```
>> C(5)
```

```
ans =  
7
```

але може бути і джерелом помилок у обчисленнях, якщо один індекс випадково вказаний замість двох – ніякого попередження не виводиться. Краще використовувати два індекси, за винятком деяких спеціальних випадків, наприклад, проходження елементів матриці по стовпцям.

### Додавання, віднімання, множення, транспонування і зведення до ступеню

При виконанні матричних операцій слід пам'ятати, що для додавання або віднімання матриці повинні бути одного розміру, а при перемноженні число стовбців першої матриці обов'язково повинні дорівнювати числу рядків другої матриці. Додавання і віднімання матриць, так же як чисел і векторів, здійснюється за допомогою знаків плюс і мінус. Для множення матриць призначена зірочка. Виконаємо дії з матрицями, введеними вище.

Додавання матриць: Обчислити суму матриць  $A$  і  $C$ .

```
>> S = A+C          >> S = C+A
S =                  S =
  6  0  6            6  0  6
  6  6  3            6  6  3
```

Віднімання матриць: Обчислити різницю матриць  $A$  і  $C$ :

```
>> R = A-C          >> R = C-A
R =                  R =
  0  2  -8           0  -2  8
 -2  2  3            2  -2  -3
```

Для множення матриць призначена зірочка:

Множення матриць: Обчислити добуток матриць  $C$  і  $B$ :

```
>> P = C*B
P =
 -25  9  11
  20 26  -4
```

Множення матриць на число також здійснюється за допомогою зірочки, причому помножити на число можна як справа, так і зліва:

```
>> P = A*3          >> P = 3*A
P =                  P =
  9  3  -3           9  3  -3
  6 12  9            6 12  9
```

Транспонування матриць, так же як і вектора, здійснюється за допомогою '.', а символ ' означає комплексне сполучення. Для дійсних матриць ці операції призводять до однакових результатів:

```
>> B'              >> B.'
ans =               ans =
  4  2  -5           4  2  -5
  3  7  1            3  7  1
 -1  0  2           -1  0  2
```

Зведення матриць до ступеню:

Квадратна матриця в цілу ступінь зводиться за допомогою оператора ^



```
>> B2 = V^2          >> B2 = V*B
B2 =                 B2 =
 27 32 -6            27 32 -6
 22 55 -2            22 55 -2
-28 -6 9             -28 -6 9
```

**Завдання 1.** Знайти значення наступного виразу:  $(A+C)V^3(A-C)^0$ .

Необхідно враховувати *пріоритет* операцій: спочатку виконується транспонування, потім зведення до ступеня, потім множення, а додавання і віднімання виконується в останню чергу:

```
>> (A+C)*V^3*(A-C)
ans =
 1848 1914
10290 3612
```

### Перемноження матриці і вектору

Оскільки вектор-стовбець або вектор-рядок в MATLAB являються матрицями, у яких один із розмірів дорівнює одиниці, то усі вищеописані операції застосовуються і для множення матриці на вектор, або вектор-рядку на матрицю. Наприклад, обчислимо вираз:

$$[1 \ 3 \ -2], \begin{pmatrix} 2 & 0 & 1 \\ -4 & 8 & -1 \\ 0 & 9 & 2 \end{pmatrix}, \begin{bmatrix} -8 \\ 3 \\ 4 \end{bmatrix}$$

Обчислити його можна наступним чином:

```
>> a = [1 3 -2];
>> b = [2 0 1; -4 8 -1; 0 9 2];
>> c = [-8; 3; 4];
>> a*b*c
ans =
 74
```

У математиці не визначена операція ділення для матриць і векторів, однак в MATLAB символ \ використовується для рішення систем лінійних рівнянь.

**Завдання 2:** Задані матриці  $A = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}$  і  $C = \begin{pmatrix} 3 & -1 \\ 4 & 2 \end{pmatrix}$ . Виконаємо операції додавання віднімання, множення, ділення, зведення до ступеня, транспонування матриць. Завдання виконується в два етапи: обчислення виконується з допомогою можливостей пакету MATLAB, потім обчислення виконується без використання комп'ютера відповідно прийнятим правилам або використовуючи додаток Excel також згідно прийнятих правил. Отримані результати порівнюються. Збіг результатів підтверджує правильність виконання завдання.

Зразок виконання завдання

1) Введемо вхідний масив A в командному рядку:

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

```
3 1
2 4
```

2) Введемо вхідний масив  $C$  використавши команду:

```
>> C = [3 -1; 4 2]
```

```
C =
```

```
3 -1
4 2
```

3) Обчислимо суму матриць  $A$  і  $C$  (з допомогою пакета MATLAB)

```
>> S = A+C
```

```
S =
```

```
6 0
6 6
```

Виконати додавання по правилу  $\begin{pmatrix} a_{11} + c_{11} & a_{12} + c_{12} \\ a_{21} + c_{21} & a_{22} + c_{22} \end{pmatrix}$  (з допомогою додатку

Excel) (рис.45):

	A	B	C	D	E	F	G	H	I	J
1	матр А			матр С				результат		
2		3	1		3	-1		6	0	
3		2	4		4	2		6	6	
4										
5										

Рис. 45. Обчислення суми матриць за допомогою додатку Excel

Порівнявши результати можна зробити висновок про правильність виконання завдання.

4) Обчислити різницю матриць  $A$  і  $C$ :

```
>> R = A-C
```

```
R =
```

```
0 2
-2 2
```

Виконати віднімання по правилу  $\begin{pmatrix} a_{11} - c_{11} & a_{12} - c_{12} \\ a_{21} - c_{21} & a_{22} - c_{22} \end{pmatrix}$  (з допомогою додатку

Excel) (рис.46):

	A	B	C	D	E	F	G	H	I	J
1	матр А			матр С				результат		
2		3	1		3	-1		0	2	
3		2	4		4	2		-2	2	
4										
5										

Рис. 46. Обчислення різниці матриць за допомогою додатку Excel

Порівняти результати.

5) Обчислити добуток матриць  $A$  і  $C$ . Множення кожного елемента однієї матриці на відповідний елемент другої відбувається за допомогою оператора  $*$

```
>> P = A.*C
```

```
P =
```

```
 9 -1
```

```
 8  8
```

Добуток виконати по правилу  $\begin{pmatrix} a_{11} * c_{11} & a_{12} * c_{12} \\ a_{21} * c_{21} & a_{22} * c_{22} \end{pmatrix}$  (з допомогою додатку

Excel) (рис.47):

	A	B	C	D	E	F	G	H	I	J
1	матр А				матр С			результат		
2	3	1		3	-1		9	-1		
3	2	4		4	2		8	8		
4										
5										

Рис. 47. Обчислення добуток матриць за допомогою додатку Excel  
Порівняти результати.

б) Для ділення елементів першої матриці на відповідні елементи другої матриці використовується ./

```
>> D = A./C
```

```
D =
```

```
1.0000 -1.0000
```

```
0.5000  2.0000
```

Ділення виконати по правилу  $\begin{pmatrix} a_{11} / c_{11} & a_{12} / c_{12} \\ a_{21} / c_{21} & a_{22} / c_{22} \end{pmatrix}$  (з допомогою додатку

Excel) (рис.48):

	A	B	C	D	E	F	G	H	I	J
1	матр А				матр С			результат		
2	3	1		3	-1		1	-1		
3	2	4		4	2		0,5	2		
4										
5										

Рис. 48. Обчислення ділення матриць за допомогою додатку Excel  
Результати порівняти.

7) Для ділення елементів другої матриці на відповідні елементи першої служить ./

```
>> D = A.\C
```

```
D =
```

```
1.0000 -1.0000
```

```
2.0000  0.5000
```

Ділення виконати по правилу  $\begin{pmatrix} c_{11} / a_{11} & c_{12} / a_{12} \\ c_{21} / a_{21} & c_{22} / a_{22} \end{pmatrix}$  (з допомогою додатку

Excel) (рис.49):

	A	B	C	D	E	F	G	H	I	J
1	матр А				матр С			результат		
2	3	1		3	-1		1	-1		
3	2	4		4	2		2	0,5		
4										
5										

Рис. 49. Обчислення ділення елементів другої матриці на відповідні елементи першої за допомогою додатку Excel

Результати порівняти.

8) Поелементне зведення до степеню здійснюється за допомогою .^

>> PS=A.^2

PS =

```
9  1
4 16
```

Зведення до степеню виконується по правилу  $\begin{pmatrix} a_{11}^2 & a_{12}^2 \\ a_{21}^2 & a_{22}^2 \end{pmatrix}$  (3)

допомогою додатку Excel) (рис.50):

	A	B	C	D	E	F	G
1	матр А				результат		
2	3	1		9	1		
3	2	4		4	16		
4							
5							

Рис. 50. Зведення до степеню елементів матриці за допомогою додатку Excel

Результати порівняти.

9) Транспонування матриці А, виконується за допомогою .'.

>> A.'

ans =

```
3  2
1  4
```

Транспонування матриці А виконується по правилу  $\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$  (3)

допомогою додатку Excel) (рис.51):

	A	B	C	D	E	F	G
1	матр А				результат		
2	3	1		3	2		
3	2	4		1	4		
4							
5							

Рис. 51. Транспонування матриці А за допомогою додатку Excel  
Порівняти результати.

## Індивідуальні завдання за темою «Робота з матрицями. Операції з матрицями»

### Варіанти індивідуальних завдань

Виконати операції додавання, віднімання, множення, ділення, зведення до степеню, транспонування матриць. Завдання виконати в два етапи: виконати обчислення за допомогою можливостей пакета MATLAB, потім виконати обчислення без використання комп'ютера відповідно до прийнятих правил або використовуючи додаток Excel і також застосовуючи прийняті правила. Результати порівняти.

$$1. \begin{pmatrix} 3 & -2 \\ 5 & -4 \end{pmatrix} \cdot \begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix}$$

$$2. \begin{pmatrix} 1 & -3 & 2 \\ 3 & -4 & 1 \\ 2 & -5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 2 & 5 & 6 \\ 1 & 2 & 5 \\ 1 & 3 & 2 \end{pmatrix}$$

$$3. \begin{pmatrix} 5 & 8 & -4 \\ 6 & 9 & -5 \\ 4 & 7 & -3 \end{pmatrix} \cdot \begin{pmatrix} 3 & 2 & 5 \\ 4 & -1 & 3 \\ 9 & 6 & 5 \end{pmatrix}$$

$$4. \begin{pmatrix} 2 & -1 & 3 & -4 \\ 3 & -2 & 4 & -3 \\ 5 & -3 & -2 & 1 \\ 3 & -3 & -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 7 & 8 & 6 & 9 \\ 5 & 7 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 2 & 1 & 1 & 2 \end{pmatrix}$$

$$5. \begin{pmatrix} 5 & 7 & -3 & -4 \\ 7 & 6 & -4 & -5 \\ 6 & 4 & -3 & -2 \\ 8 & 5 & -6 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

$$6. \begin{pmatrix} 2 & -3 \\ 4 & -6 \end{pmatrix} \cdot \begin{pmatrix} 9 & -6 \\ 6 & -4 \end{pmatrix}$$

$$7. \begin{pmatrix} 5 & 2 & -2 & -3 \\ 6 & 4 & -3 & 5 \\ 9 & 2 & -3 & 4 \\ 7 & 6 & -4 & 7 \end{pmatrix} \cdot \begin{pmatrix} 2 & 2 & 2 & 2 \\ -1 & -5 & 3 & 11 \\ 16 & 24 & 8 & -8 \\ 8 & 16 & 0 & -16 \end{pmatrix}$$

$$8. \begin{pmatrix} 4 & 3 \\ 7 & 5 \end{pmatrix} \cdot \begin{pmatrix} -28 & 93 \\ 38 & -126 \end{pmatrix}$$

$$9. \begin{pmatrix} 0 & 2 & -1 \\ -2 & -1 & 2 \\ 3 & -2 & -1 \end{pmatrix} \cdot \begin{pmatrix} 70 & 34 & -107 \\ 52 & 26 & -68 \\ 101 & 50 & -140 \end{pmatrix}$$

$$10. \begin{pmatrix} 1 & 1 & 1 & -1 \\ -5 & -3 & -4 & 4 \\ 5 & 1 & 4 & -3 \\ -16 & -11 & -15 & 14 \end{pmatrix} \cdot \begin{pmatrix} 7 & -2 & 3 & 4 \\ 11 & 0 & 3 & 4 \\ 5 & 4 & 3 & 0 \\ 22 & 2 & 9 & 8 \end{pmatrix}$$

$$11. \begin{pmatrix} -28 & 93 \\ 38 & -126 \end{pmatrix} \cdot \begin{pmatrix} 7 & 3 \\ 2 & 1 \end{pmatrix}$$

$$12. \begin{pmatrix} 70 & 34 & -107 \\ 52 & 26 & -68 \\ 101 & 50 & -140 \end{pmatrix} \cdot \begin{pmatrix} 27 & -18 & 10 \\ -46 & 31 & -17 \\ 3 & 2 & 1 \end{pmatrix}$$

$$13. \begin{pmatrix} 1 & 2 & -1 \\ -2 & -1 & 2 \\ 3 & -2 & -1 \end{pmatrix} \cdot \begin{pmatrix} 27 & -18 & 10 \\ -46 & 31 & -17 \\ 3 & 2 & 1 \end{pmatrix}$$

$$14. \begin{pmatrix} 4 & 3 & -1 \\ 2 & 7 & 0 \\ -5 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 & -1 & 7 \\ 4 & 2 & 1 \\ 5 & 3 & 4 \end{pmatrix}$$

$$15. \begin{pmatrix} 2 & -1 & 3 & -4 \\ 3 & -2 & 4 & -3 \\ 5 & -3 & -2 & 1 \\ 3 & -3 & -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}$$

$$16. \begin{pmatrix} 5 & 7 & -3 & -4 \\ 7 & 6 & -4 & -5 \\ 6 & 4 & -3 & -2 \\ 8 & 5 & -6 & -1 \end{pmatrix} \cdot \begin{pmatrix} 7 & 8 & 6 & 9 \\ 5 & 7 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 2 & 1 & 1 & 2 \end{pmatrix}$$

$$17. \begin{pmatrix} 1 & 3 & -1 \\ 8 & 5 & -3 \\ -5 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} -2 & -1 & 7 \\ 5 & 6 & 1 \\ 3 & 3 & 4 \end{pmatrix}$$

$$18. \begin{pmatrix} -8 & 3 \\ 8 & 6 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 \\ 5 & 2 \end{pmatrix}$$

$$19. \begin{pmatrix} -2 & 9 \\ 3 & -16 \end{pmatrix} \cdot \begin{pmatrix} 9 & 8 \\ 7 & 6 \end{pmatrix}$$

$$20. \begin{pmatrix} 9 & 3 & -2 \\ 1 & 2 & 4 \\ 6 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} -8 & 5 & 3 \\ 7 & 8 & 1 \\ 2 & 6 & 4 \end{pmatrix}$$

$$21. \begin{pmatrix} 1 & 3 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 5 & -3 & -2 \\ 3 & -2 & 4 \\ 2 & -1 & 3 \end{pmatrix}$$

$$22. \begin{pmatrix} 8 & 16 & 2 \\ 16 & 24 & 8 \\ -1 & -5 & 3 \end{pmatrix} \cdot \begin{pmatrix} 7 & 6 & -4 \\ 9 & 2 & -3 \\ 6 & 4 & -3 \end{pmatrix}$$

$$23. \begin{pmatrix} 2 & -2 & -3 \\ 4 & -3 & 5 \\ 2 & -3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 2 & 2 & 2 \\ -5 & 3 & 11 \\ 24 & 8 & -8 \end{pmatrix}$$

$$24. \begin{pmatrix} -2 & 3 & 4 \\ 1 & 3 & 4 \\ 4 & 3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & -1 \\ -3 & -4 & 4 \\ 1 & 4 & -3 \end{pmatrix}$$

$$25. \begin{pmatrix} 7 & 6 & -4 \\ 6 & 4 & -3 \\ 8 & 5 & -6 \end{pmatrix} \cdot \begin{pmatrix} 5 & 7 & 4 \\ 3 & 4 & 5 \\ 2 & 1 & 1 \end{pmatrix}$$

$$26. \begin{pmatrix} 5 & 2 & -1 \\ 6 & 4 & -2 \\ 9 & 2 & -2 \end{pmatrix} \cdot \begin{pmatrix} 2 & 2 & 1 \\ -1 & -5 & 3 \\ 8 & 16 & 1 \end{pmatrix}$$

$$27. \begin{pmatrix} 6 & -4 \\ 4 & -3 \end{pmatrix} \cdot \begin{pmatrix} 7 & 4 \\ 4 & 5 \end{pmatrix}$$

$$28. \begin{pmatrix} 4 & -3 \\ 2 & -3 \end{pmatrix} \cdot \begin{pmatrix} 3 & 11 \\ 8 & -5 \end{pmatrix}$$

$$29. \begin{pmatrix} 6 & 4 \\ 9 & 2 \end{pmatrix} \cdot \begin{pmatrix} -5 & 3 \\ 4 & 8 \end{pmatrix}$$

$$30. \begin{pmatrix} 3 & 2 & 1 \\ 4 & 5 & 4 \\ 1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 & 1 \\ -2 & 7 & 3 \\ 3 & 6 & 1 \end{pmatrix}$$

### Застосування функцій обробки даних до матриць

Функція *sum* обчислює суму елементів вектору. Вектори в MATLAB, так же як і матриці, зберігаються в двомірних масивах. В якості аргументу *sum* можна використовувати не вектор, а матрицю. MATLAB обчислює вектор-рядок, довжина якого дорівнює числу стовбців матриці, а кожен елемент являється сумою відповідного стовпця матриці, наприклад: введемо масив *M*:

```
>> M = [1 -2 -4; 3 -6 4; 2 -2 0]
```

```
M =
```

```

1  -2  -4
3  -6   4
2  -2   0

```

Функція *sum* за замовчуванням обчислює суму по стовбцям, змінюючи перший індекс масиву при фіксованому другому. Для того, щоб виконувати додавання по рядкам, необхідно визвати *sum* з двома аргументами, вказавши місце індексу, по якому слід виконувати додавання. Значення другого аргументу дорівнює 1, що вказує місце індексу, по якому слід виконувати додавання – по стовбцям:

```

>> s = sum(M)                >> s1 = sum(M, 1)
s =                            s1 =
 6 -10  0                      6 -10  0

```

Вочевидь, використання функцій *sum(M)* і *sum(M, 1)* призводить до однакових результатів.

Щоб здійснити знаходження суми по рядках, необхідно значення другого аргументу зробити рівним 2, що вказує місце індексу, по якому слід виконувати додавання – по рядкам:

```

>> s2 = sum(M, 2)
s2 =
-5
 1
 0

```

Таким чином, функція *sum* виконує додавання або по рядкам, або по стовбцям, видаючи результат у вигляді вектора або вектор-рядку.

Аналогічно працює і функція *prod*:

Знаходження добутку по стовбцях:

```

>> p = prod(M)
p =
 6 -24  0

```

Знаходження добутку по рядкам:

```

>> p2 = prod(M, 2)
p2 =
 8
-72
 0

```

Функція *sort* впорядковує елементи кожного із стовбців матриці в порядку зростання:

```

>> MS = sort(M)
MS =
 1  -6  -4
 2  -2   0
 3  -2   4

```

Виклик *sort* з другим аргументом, рівним 2, призводить до впорядкування елементів рядків:

```

>> MR = sort(M, 2)

```



```
MR =
  -4 -2  1
  -6  3  4
  -2  0  2
```

Як і для векторів, функція *sort* дозволяє отримати матрицю індексів відповідності елементів вхідної і упорядкованої матриць. Для цього необхідно визвати *sort* з двома вихідними аргументами:

```
>> [MC, Ind] = sort(M)
MC =
  1 -6 -4
  2 -2  0
  3 -2  4
Ind =
  1 2 1
  3 1 3
  2 3 2
```

Матриці *M*, *MC* і *Ind* зв'язані між собою так:  $MC(i, j) = M(Ind(I, j), j)$ , де *i* та *j* змінюються від одного до трьох.

Функції *max* і *min* визначають вектор-рядок, який містить максимальні або мінімальні елементи у відповідних стовбцях матриці:

```
>> mx = max(M)           >> mn = min(M)
mx =                      mn =
  3 -2  4                 1 -6 -4
```

Для того, щоб знати не тільки значення максимальних або мінімальних елементів, але і їх номера в стовбцях, слід визвати *max* або *min* таким чином:

```
>> [mx, k] = max(M)      >> [mn, n] = min(M)
mx =                      mn =
  3 -2  4                 1 -6 -4
k =                        n =
  2  1  2                 1  2  1
```

Як видно, в другому стовбці матриці *M* два рівних максимальних елемента – перший і третій. Завжди повертається номер першого максимального елемента (другий елемент у вектор-рядку дорівнює одиниці). Таким же чином працює і *min* у випадку двох рівних мінімальних елементів у стовбці матриці. Функції *max* і *min* дозволяють виділити максимальні або мінімальні елементи із двох матриць однакових розмірів і записати результат в нову матрицю того ж розміру, що і вхідні:

```
>> P = [4 3 -1
  2 0 71]
P =
  4 3 -1
  2 0 71
>> Q = [10 0 11
  -5 3 22]
G =
  10 0 11
```

```

-5 3 22
>> R = max(P, Q)
R =
10 3 11
2 3 71

```

Одним із аргументів може бути число. В результуючу матрицю записується максимум із цього числа і відповідного елемента вхідної матриці:

```

>> S = max(P, 1)
S =
4 3 1
2 1 71

```

Якщо обидва аргументи функцій *min* або *max* являються числами, то повертається мінімальне або максимальне із цих чисел.

Для знаходження максимуму або мінімуму не по стовбцям матриці, а по рядкам, передбачена наступна форма виклику з другим аргументом – пустим масивом:

```

>> mx = max(S, [], 2)
mx =
4
71

```

Для того щоб додатково отримати номери мінімальних елементів в рядках, використовується виклик *max* з двома вихідними аргументами:

```

>> [mx, j] = max(S, [], 2)
mx =
4
71
j =
1
3

```

Розглянемо функцію *rot90*, яка повертає масив на 90 градусів проти часової стрілки. Застосуємо цю функцію до введеного вище масиву *P*:

```

>> q1 = rot90(P)
q1 =
-1 71
3 0
4 2

```

Застосуємо функцію повторно, тепер уже до масиву *q1*

```

>> q2 = rot90(q1)
q2 =
71 0 2
-1 3 4

```

Функція *fliplr* забезпечує дзеркальне відображення від умовної вертикалі, яка проходить через середину масиву:

```

>> fliplr(q2)
ans =
2 0 71

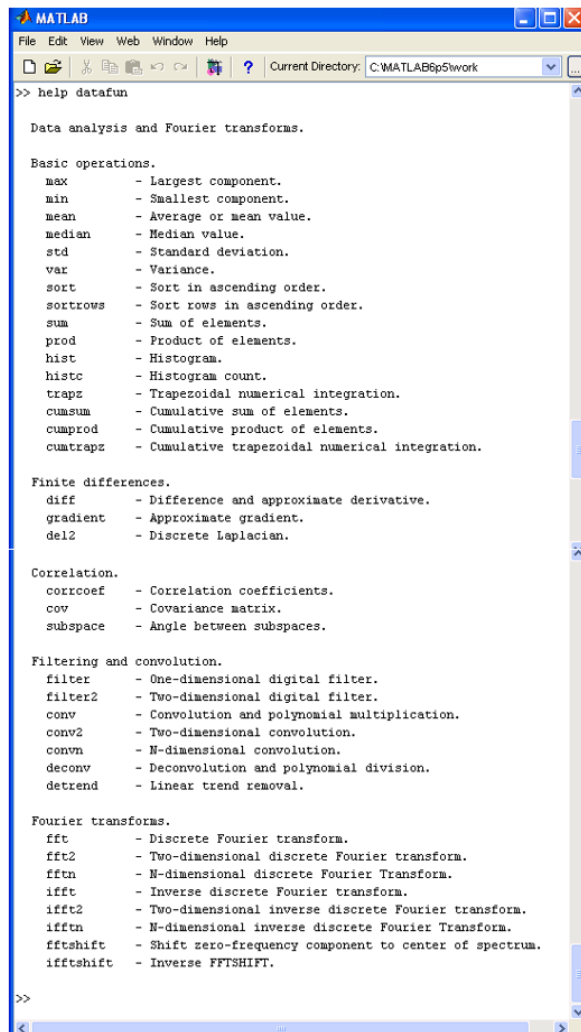
```

4 3 -1

Звернемо увагу на те, що вектор-стовбець не змінюється під впливом функції *fliplr*:

```
>> q3 = [1; 2; 3; 4; 5]
q3 =
     1
     2
     3
     4
     5
>> fliplr(q3)
ans =
     1
     2
     3
     4
     5
```

Більш детальну інформацію про обробку матричних даних можна отримати за допомогою команди *help datafun*, яка виводить список усіх вбудованих функцій обробки даних (рис. 52).



```
>> help datafun

Data analysis and Fourier transforms.

Basic operations.
max - Largest component.
min - Smallest component.
mean - Average or mean value.
median - Median value.
std - Standard deviation.
var - Variance.
sort - Sort in ascending order.
sortrows - Sort rows in ascending order.
sum - Sum of elements.
prod - Product of elements.
hist - Histogram.
histc - Histogram count.
trapz - Trapezoidal numerical integration.
cumsum - Cumulative sum of elements.
cumprod - Cumulative product of elements.
cumtrapz - Cumulative trapezoidal numerical integration.

Finite differences.
diff - Difference and approximate derivative.
gradient - Approximate gradient.
del2 - Discrete Laplacian.

Correlation.
corrcoef - Correlation coefficients.
cov - Covariance matrix.
subspace - Angle between subspaces.

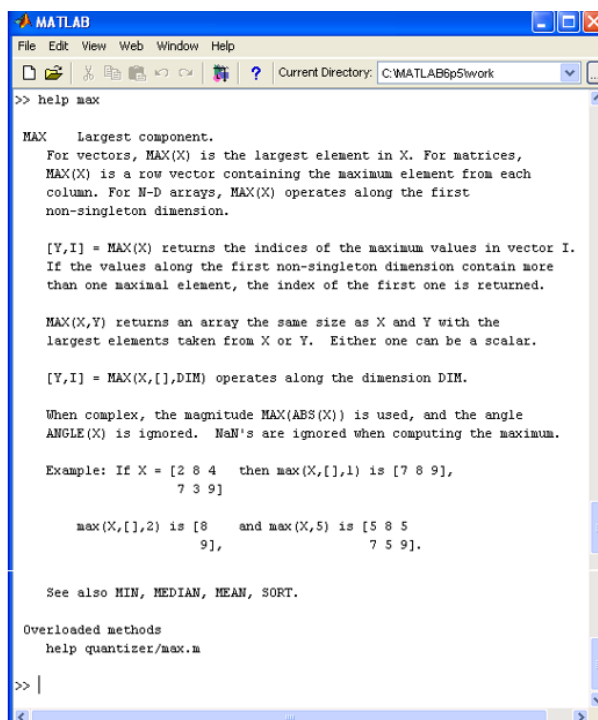
Filtering and convolution.
filter - One-dimensional digital filter.
filter2 - Two-dimensional digital filter.
conv - Convolution and polynomial multiplication.
conv2 - Two-dimensional convolution.
convn - N-dimensional convolution.
deconv - Deconvolution and polynomial division.
detrend - Linear trend removal.

Fourier transforms.
fft - Discrete Fourier transform.
fft2 - Two-dimensional discrete Fourier transform.
fftn - N-dimensional discrete Fourier Transform.
ifft - Inverse discrete Fourier transform.
ifft2 - Two-dimensional inverse discrete Fourier transform.
ifftn - N-dimensional inverse discrete Fourier Transform.
fftshift - Shift zero-frequency component to center of spectrum.
ifftshift - Inverse FFTSHIFT.

>>
```

Рис. 52. Результат роботи команди *help datafun*, яка виводить список усіх вбудованих функцій обробки даних

Потім переглянути інформацію про потрібну функцію, наприклад, *help max* (рис. 53).



```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p5\work

>> help max

MAX Largest component.
For vectors, MAX(X) is the largest element in X. For matrices,
MAX(X) is a row vector containing the maximum element from each
column. For N-D arrays, MAX(X) operates along the first
non-singleton dimension.

[Y,I] = MAX(X) returns the indices of the maximum values in vector I.
If the values along the first non-singleton dimension contain more
than one maximal element, the index of the first one is returned.

MAX(X,Y) returns an array the same size as X and Y with the
largest elements taken from X or Y. Either one can be a scalar.

[Y,I] = MAX(X,[],DIM) operates along the dimension DIM.

When complex, the magnitude MAX(ABS(X)) is used, and the angle
ANGLE(X) is ignored. NaN's are ignored when computing the maximum.

Example: If X = [2 8 4
                 7 3 9] then max(X,[],1) is [7 8 9],

max(X,[],2) is [8
                9], and max(X,5) is [5 8 5
                                   7 5 9].

See also MIN, MEDIAN, MEAN, SORT.

Overloaded methods
help quantizer/max.m

>> |

```

Рис. 53. Результат роботи команди *help max* з інформацією про функцію

### Індивідуальне завдання за темою: «Застосування функцій обробки даних до матриць»

Самостійно скласти матрицю розміром 3 x 3 або 4 x 4 із довільних елементів. Застосувати до матриці усі розглянуті функції обробки даних до матриць.

### Графіка MATLAB

Однією з переваг системи MATLAB є чисельність засобів графіки, починаючи від команд побудови графіків функцій однієї змінної в декартовій системі координат і закінчуючи комбінованими і презентаційними графіками з елементами анімації, а також засобами проектування графічного користувацького інтерфейсу. Особливу увагу в системі приділено трьохмірній графіці з функціональним забарвленням фігур, що відображаються і імітацією різних світлових ефектів.

#### Побудова графіків відрізками прямих. Графіки функцій однієї змінної

Функції однієї змінної  $y(x)$  широко застосовуються в практиці математичних і інших розрахунків, а також у техніці комп'ютерного математичного моделювання. Для відображення функцій однієї змінної  $y(x)$  використовуються графіки в декартовій (прямокутній) системі координат. При цьому будуються дві осі – горизонтальна  $X$  і вертикальна  $Y$ , і задаються координати  $x$  і  $y$ , які визначають вузлові точки функції  $y(x)$ . Ці точки об'єднуються одна з одною відрізками прямих, тобто при побудові графіку здійснюється лінійна інтерполяція для проміжних точок. Оскільки MATLAB – матрична система, сукупність точок  $y(x)$  задається векторами  $X$  і  $Y$  однакового розміру.

Команда `plot (X, Y)` служить для побудови графіків функцій в декартовій системі координат. Синтаксис команди:

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

Розглянемо приклад, який ілюструє побудову графіків двох функцій  $\sin(x)$  і  $\cos(x)$ , значення яких містяться в матриці  $Y$ , а значення аргументу  $x$  зберігається у векторі  $x$  (рис. 54, рис. 55). В робочому вікні створимо вектор  $x$ , обчислимо значення функції  $\sin(x)$ , введемо команду `plot` для побудови графіка:

```
>> x=[0 1 2 3 4 5];
>> Y=sin(x);
>> plot(x,Y)
>>
```

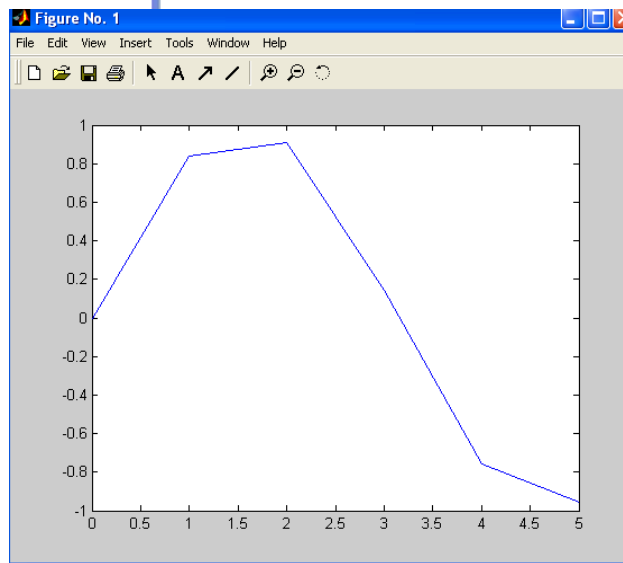


Рис. 54. Графік функції  $\sin(x)$

Тепер створимо вектор  $x$  і обчислимо значення функції  $\cos(x)$ , введемо команду `plot` для побудови графіка:

```
>> x=[0 1 2 3 4 5];
>> Y=cos(x);
>> plot(x,Y)
>>
```

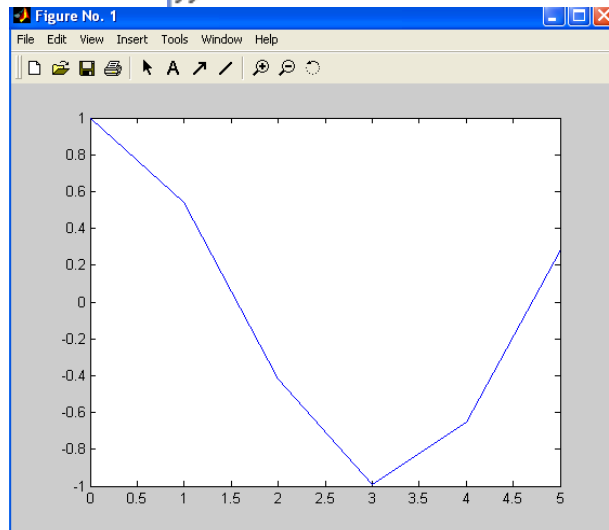


Рис. 55. Графік функції  $\cos(x)$

Далі графіки функцій  $\sin(x)$  і  $\cos(x)$  розташуємо в одному графічному вікні. Виклик функції *plot* з числовими парами  $x$  і  $y$  створює числові графіки. При відсутності вказівки на колір ліній і точок MATLAB автоматично присвоює кожному графіку свій колір, тип, що дозволяє розрізняти задані набори даних (рис. 56):

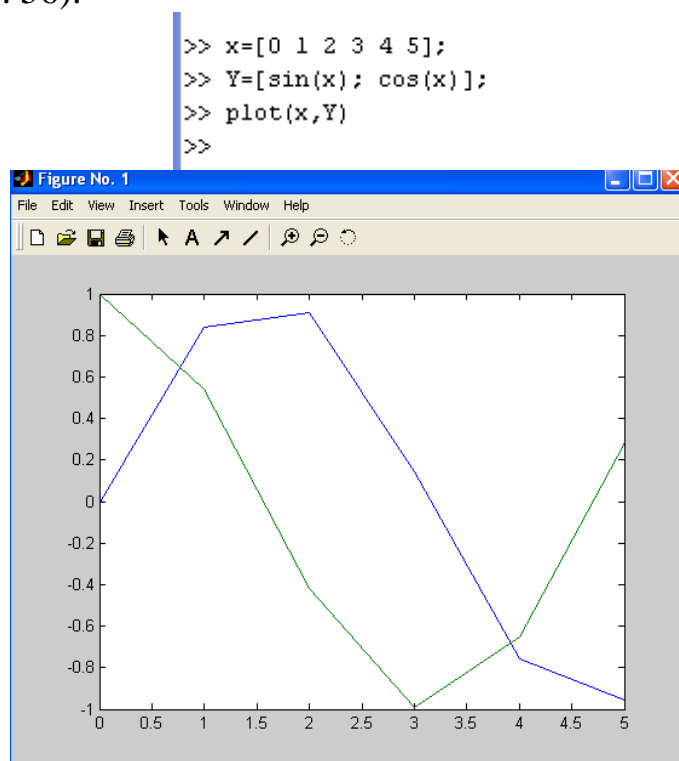


Рис. 56. Графіки функцій  $\sin(x)$  і  $\cos(x)$

Для того, щоб відобразити графіки функцій в одному графічному вікні і кожен графік щоб мав свій колір необхідно вказати це в команді *plot*.

Синтаксис: `plot(X,Y,S)`

Тип лінії графіка можна задавати з допомогою строкової константи  $S$ . Значеннями константи  $S$  можуть бути наступні символи:

Тип лінії	символ	Тип маркера	символ	Колір лінії	символ
суцільна	-	плюс	+	блакитний	c
розривна штрихова	-	кружок	o	малиновий	t
пунктирна	:	зірочка	*	жовтий	y
штрихпунктирна	-.	хрест	x	червоний	r
відсутність лінії	none	крапка	.	зелений	g
		квадрат	s	синій	b
		ромб	d	білий	w
		п'ятигранник	p	чорний	k
		шестигранник	h	фіолетовий	m
		стрілка вниз	v		
		стрілка вгору	^		
		стрілка вліво	<		
		стрілка вправо	>		

Таким чином, за допомогою строкової константи S можна змінювати колір лінії, представляти вузлові точки різними відмітками (крапка, коло, хрест, трикутник з різною орієнтацією вершин) и змінювати тип лінії графіка.

Розглянемо приклад побудови графіків трьох функцій з різним стилем представлення кожної із них (рис. 57):

```
>> x=-pi:0.1:pi;
>> y1=sin(x);
>> y2=exp(-x.^2);
>> y3=atan(x);
>> plot(x,y1,'-m',x,y2,'-.+r',x,y3,'--ok')
>>
```

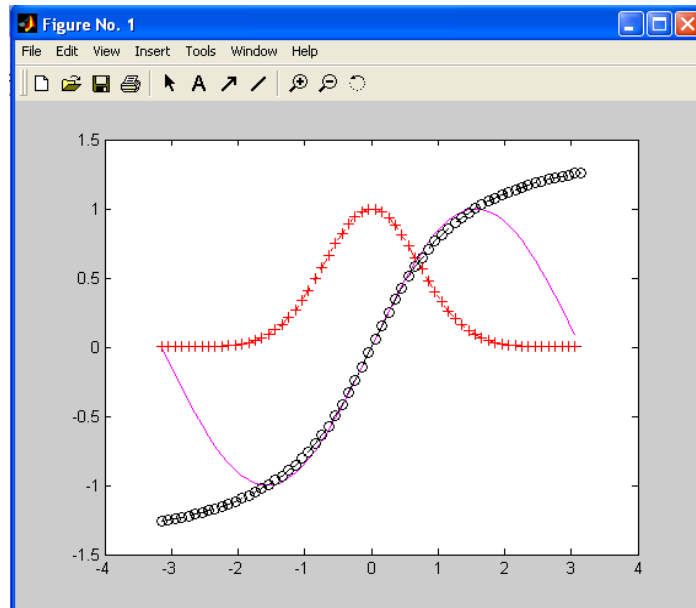


Рис. 57. Графіки трьох функцій

Графік функції  $y_1$  будується суцільною фіолетовою лінією, графік  $y_2$  будується штрих пунктирною лінією з точками у вигляді знака «плюс» червоного кольору, а графік  $y_3$  будується штриховою лінією з кружками чорного кольору.

### Оформлення графіків

Додаткові прикраси графіка полягають у можливості забезпечити його заголовком, підписати осі, нанести координатну сітку і розмістити легенду:

- title – виводить заголовок графіка
- xlabel, ylabel – підписує осі графіка
- grid on – наносить координатну сітку
- legend – розміщує легенду

Синтаксис legend (опис 1-го графіка, опис 2-го графіка, місце розміщення легенди в графічному вікні). Параметр місце розміщення легенди в графічному вікні може приймати наступні значення:

- 1 – легенда розміщується поза полем графіка, вгорі праворуч;
- 0 – система вибирає краще місце в полі графіка, що не перекривається даними;
- 1 – легенда розміщується в правому верхньому куті (за замовчуванням – там же);
- 2 – легенда розміщується в лівому верхньому кутку поля графіка;

- 3 – легенда розміщується в лівому нижньому кутку поля графіка;  
4 – легенда розміщується в правому нижньому кутку поля графіка.

Приклад оформлення графіка підписами (рис. 58):

```
>> x=0:0.2:6.28;
>> y1=sin(x);
>> y2=cos(x);
>> plot(x,y1,'-',x,y2,'*')
>> legend('sin','cos',4)
>> ylabel('y-axis')
>> xlabel('x-axis')
>> grid on
>> title('Function sin & cos')
>> |
```

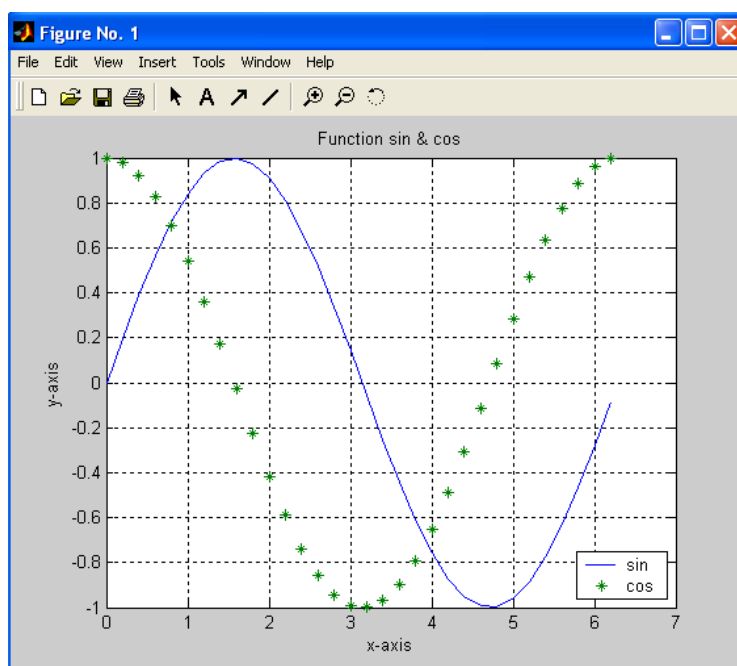


Рис. 58. Графік, оформлений підписами

## Індивідуальні завдання за темою: «Графіки функцій однієї змінної»

### Варіанти індивідуальних завдань

**Задача А.** Розрахувати і вивести на екран значення функції  $f(x)$  при заданих значеннях параметрів  $a, b$ . Аргумент  $x$  змінюється в діапазоні від  $x_H$  до  $x_K$  з кроком  $\Delta x$ . Побудувати графік функції на заданому діапазоні.

**Задача Б.** Розрахувати і вивести на екран значення функції  $f(x)$  при заданих значеннях аргументу  $x$ . Побудувати графік функції для заданих точок.

Дані для розрахунків

№ з/п	Функція $y_i = f(x_i)$	Задача А				Задача Б					
		a	b	$x_H$	$x_K$	$\Delta x$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
1.	$y = \frac{1 + \sin^2(b^3 + x^3)}{\sqrt[3]{b^3 + x^3}}$		2,5	1,28	3,28	0,4	1,1	2,4	3,6	1,7	3,9



2.	$y = \frac{\sqrt[3]{ax+b}}{\lg^2 x}$	1,35	0,98	1,14	4,24	0,62	0,35	1,28	3,51	5,21	4,16
3.	$y = \frac{1 + \lg^2 \frac{x}{a}}{b - e^{\frac{x}{a}}}$	2,0	0,95	1,25	2,75	0,3	2,2	3,78	4,51	6,58	1,2
4.	$y = \sqrt[4]{ x^2 - 2,5 } + \sqrt[3]{\lg x^2}$			1,25	3,25	0,4	1,84	2,71	3,81	4,56	5,62
5.	$y = \frac{a^x - b^x}{\lg \frac{a}{b}} \sqrt[3]{ab}$	0,4	0,8	3,2	6,2	0,6	4,48	3,56	2,78	5,28	3,21
6.	$y = \frac{b^3 + \sin^2 ax}{\arccos(xbx) + e^{-x/2}}$	1,2	0,48	0,7	2,2	0,3	0,25	0,36	0,56	0,94	1,28
7.	$y = \frac{\lg(x^2 - 1)}{\log_5(ax^2 - b)}$	1,1	0,09	1,2	2,2	0,2	1,21	1,76	2,53	3,48	4,52
8.	$y = a^{x^2-1} - \lg(x^2 - 1) + \sqrt[3]{x^2 - 1}$	1,6		1,2	3,7	0,5	1,28	1,36	2,47	3,68	4,56
9.	$y = \frac{a\sqrt{x} - b \log_5 x}{\lg x-1 }$	4,1	2,7	1,2	5,2	0,8	1,9	2,15	2,34	2,73	3,16
10.	$y = \sqrt{\frac{ a-bx }{\lg^3 x}}$	7,2	4,2	1,81	5,31	0,7	2,4	2,8	3,9	4,7	3,16
11.	$y = \frac{\log_a  b^2 - x^2 }{\sqrt[5]{ x^2 - a^2 }}$	2,0	1,1	0,08	1,08	0,2	0,1	0,3	0,4	0,45	0,62
12.	$y = \frac{a + tg^2 bx}{b + ctg^2 ax}$	0,1	0,5	0,15	1,37	0,25	0,2	0,3	0,44	0,6	0,56
13.	$y = \frac{(a+bx)^{2,5}}{1 + \lg(a+bx)}$	2,5	4,6	1,1	3,6	0,5	1,2	1,28	1,36	1,46	2,35
14.	$y = \frac{\lg^2(a+x)}{(a+x)^2}$	2,0		1,2	4,2	0,6	1,16	1,32	1,47	1,65	1,93
15.	$y = a^{x^2-1} - \lg(x^2 - 1) + \sqrt[3]{x^2 - 1}$	2,25		1,2	2,7	0,3	1,31	1,39	1,44	1,56	1,92
16.	$y = \frac{a\sqrt[3]{x} - b \log_5 x}{\lg^3(x-1)}$	4,1	2,7	1,5	2,5	0,4	1,9	2,15	2,34	2,74	3,16
17.	$y = \frac{\sqrt[3]{(x-a)^2} + \sqrt[5]{ x+b }}{\sqrt[9]{x^2 - (a+b)^2}}$	0,8	0,4	1,23	7,23	1,2	1,88	2,26	3,84	4,55	6,21
18.	$y = \sqrt[5]{\frac{a+bx}{\lg^3 x}}$	7,2	1,3	1,56	4,71	0,63	2,4	2,8	3,9	4,7	3,16
19.	$y = (\sin^3 x + \cos^3 x) \ln x$			0,11	0,36	0,05	0,2	0,3	0,38	0,43	0,57
20.	$y = (\arcsin^2 x + \arccos^4 x)^3$			0,26	0,66	0,08	0,1	0,35	0,4	0,55	0,6

## Кольорові плоскі кругові діаграми

Зафарбовані сектори часто використовуються для побудови кругових діаграм. Для цього в MATLAB існує команда `pie`.

Команда `pie(X)` – будує кругову діаграму по даним нормалізованого вектора. Розглянемо приклад побудови кольорової кругової діаграми з п'ятьма секторами, причому останній сектор відділений від інших (рис. 59):

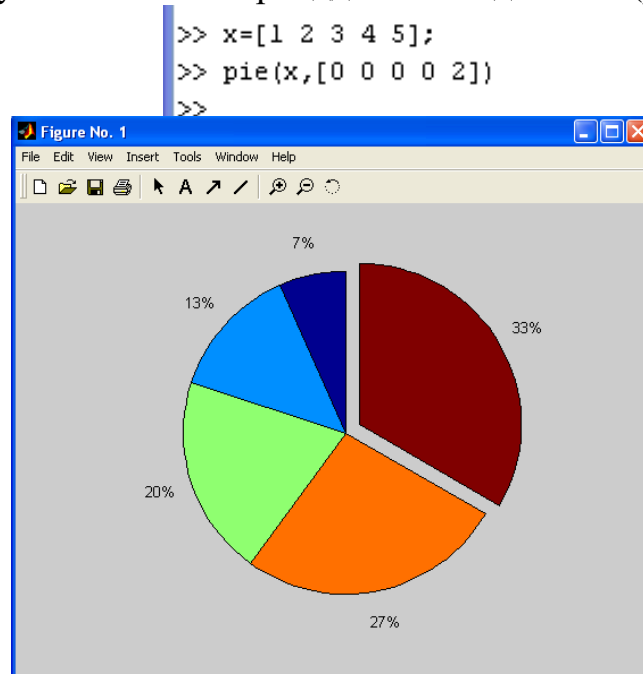


Рис. 59. Кругова діаграма з п'ятьма секторами і п'ятим відділеним від інших. Розглянемо приклад. Необхідно оцінити вклад кожного із елементів вектору в загальну суму його елементів. За допомогою функції `pie` побудуємо кругову діаграму (рис. 60).

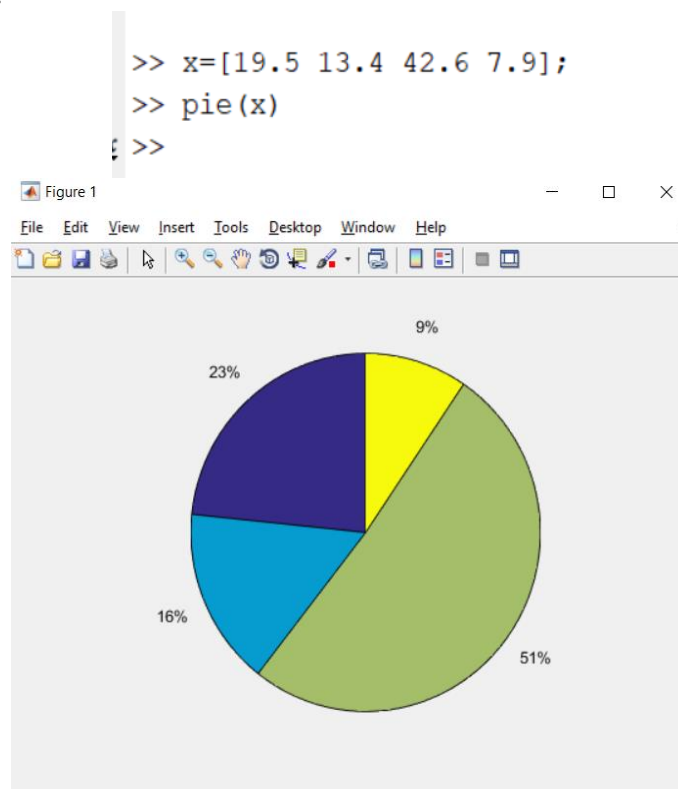


Рис. 60. Плоска кругова діаграма

На діаграмі зображені площі секторів, які відповідають процентному вкладу кожного з елементів вектору в загальну суму, тобто MatLab нормує значення, обчислюючи  $x/\text{sum}(x)$

```
>> % Перевірка
>> x/sum(x)

ans =

    0.2338    0.1607    0.5108    0.0947
```

Розглянемо приклади, коли необхідно відсунути від кола діаграми сектор, відповідний деякому елементу. Це можна зробити, задаючи другим елементом функції *pie* вектор, що складається з одиниць і нулів, причому одиниця стоїть у позиції, що відповідає номеру частини, що відокремлюється (рис. 61, рис. 62).

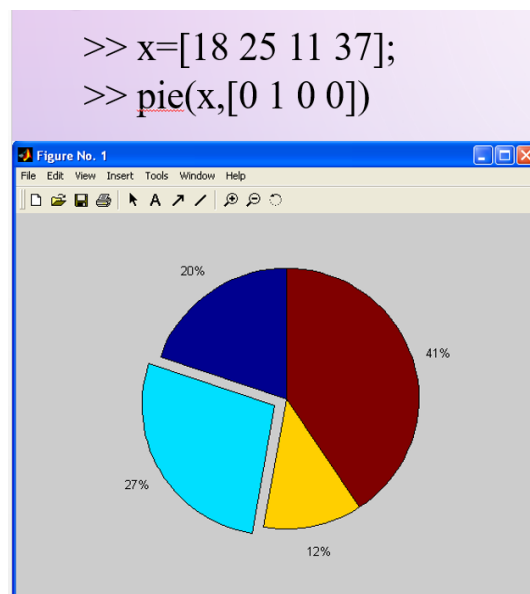


Рис. 61. Кругова діаграма з чотирма секторами і другим сектором відділеним від інших (1 стоїть в другій позиції)

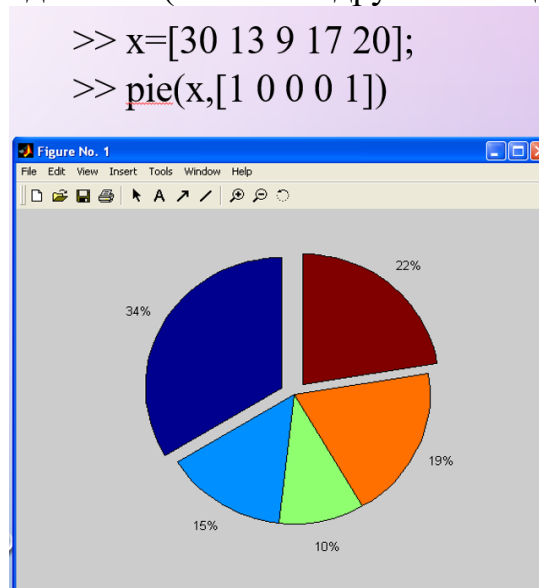


Рис. 62. Кругова діаграма з п'ятьма секторами і першим та п'ятим відділеним від інших (1 стоїть в першій та останній позиції)

### Кольорові об'ємні кругові діаграми

Іноді використовуються об'ємні кругові діаграми. Для їх побудови служить команда `pie3`. Синтаксис команди:

Команда `pie3 (...)` аналогічна команді `pie (...)`, але дає побудову об'ємних секторів. Розглянемо на прикладі побудову об'ємної діаграми з відділенням двох секторів (рис. 63):

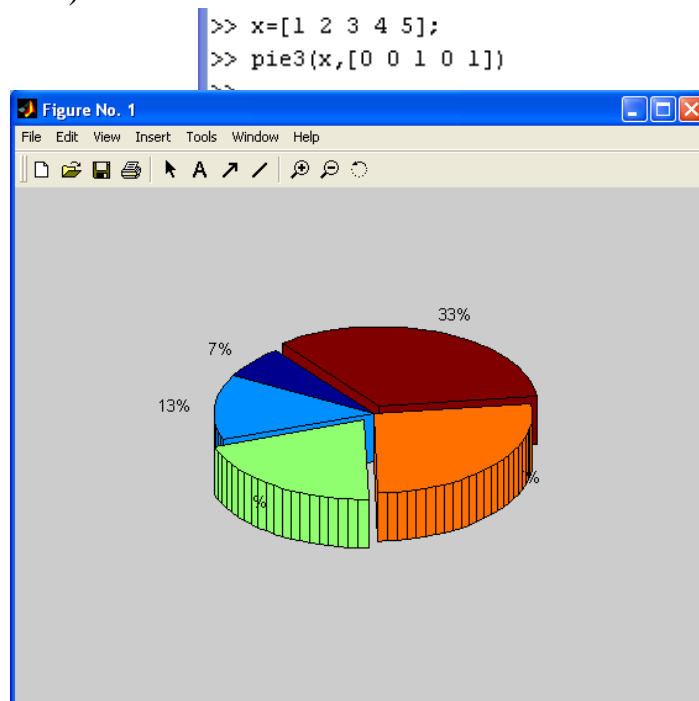


Рис. 63. Об'ємна кругова діаграма з п'ятьма секторами, серед яких відділені сектори третій та п'ятий

Розглянемо приклад побудови об'ємної діаграми з відділенням трьох секторів. На номери секторів, які необхідно відділити вказують позиції одиниць у векторі (рис. 64):

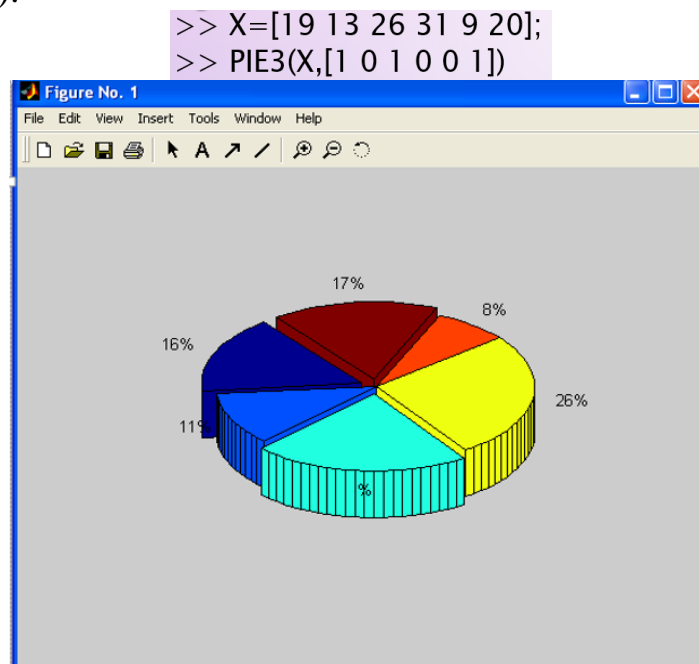


Рис. 64. Об'ємна кругова діаграма з шістьма секторами, серед яких відділені сектори перший, третій та шостий

## Стовбцеві діаграми

Розглядаючи приклади, у розрахунках часто зустрічаються графіки, які називаються стовбцевими діаграмами, які відображають зміст деякого вектору  $V$ . При цьому кожен елемент вектору представляється стовбцем, висота якого пропорційна значенню елемента. Стовбці нумеруються і масштабуються по відношенню до максимального значення найбільш високого стовпця.

Функція `bar` – будує вертикальну діаграму.

Функція `barh` – будує горизонтальну діаграму.

Функція `bar3` – будує об'ємну діаграму.

Синтаксис команди `bar(t,x,0.3)`

Перший аргумент, це вектор або матриця з координатами точок розмітки. Другий аргумент, це вектор значень, які потрібно відобразити на діаграмі. Розмірність аргументів повинна збігатися.

Третій аргумент, це ширина стовпців (за замовчуванням дорівнює 0,8).

Номер елемента вектору відкладається по горизонтальній осі, а його значення – по вертикальній. Аргументом функції `bar` може бути як вектор-рядок так і вектор-стовбець.

Команда `bar(V)` – будує графік значень елементів матриці  $V$ . Для побудови графіка використовується вектор  $V = 1:m$ .

Наприклад, задано вектор  $x$

```
>> x=[100 150; 240 240; 150 120; 216 288; 30 60]
x =
    100    150
    240    240
    150    120
    216    288
     30     60
>> bar(x)
```

Використовуючи команду `bar` побудуємо графік стовбцевої діаграми (рис. 65).

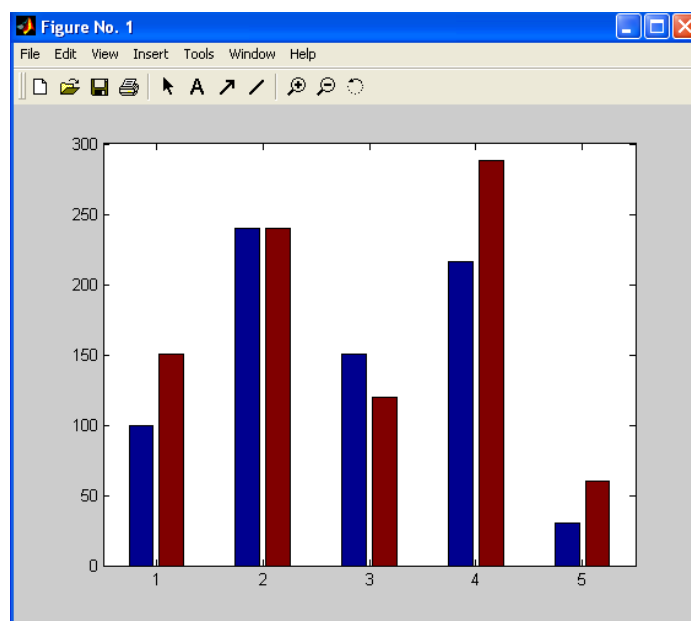


Рис. 65. Графік стовбцевої діаграми

Розглянемо приклад побудови діаграми векторних даних. Відобразити на відрізку [1.2, 2.5] функцію  $x(t) = \sin t * e^t$  у вигляді стовбцевої діаграми, значення третього аргумента за замовчуванням дорівнює нулю (рис. 66):

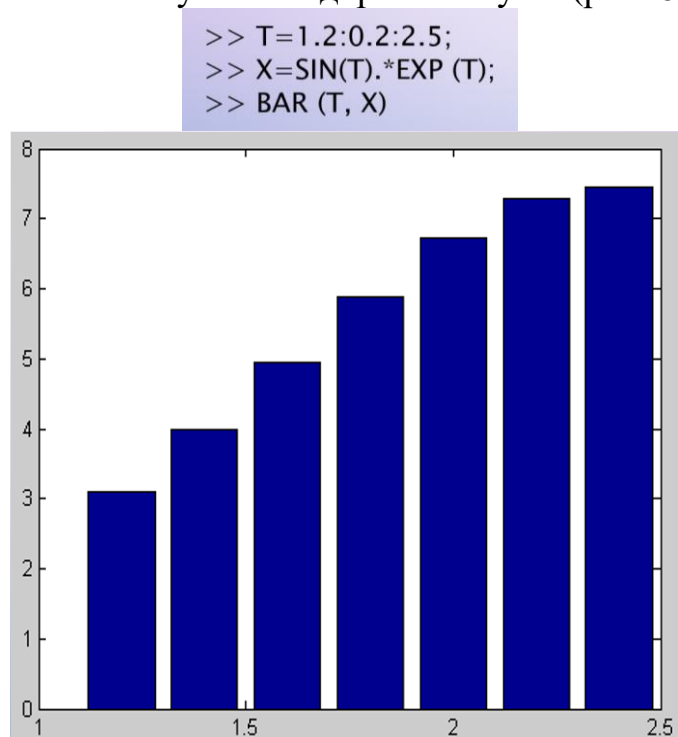


Рис. 66. Стовбцева діаграма функції  $x(t) = \sin t * e^t$ . Значенням аргумента ширини стовбця за замовчуванням дорівнює нулю

Розглянемо приклади використання третього аргумента в синтаксисі команди `bar`, значенням якого можна задавати ширину стовбця. При значенні аргумента, яке дорівнює 0.3 діаграма має такий вигляд (рис. 67):

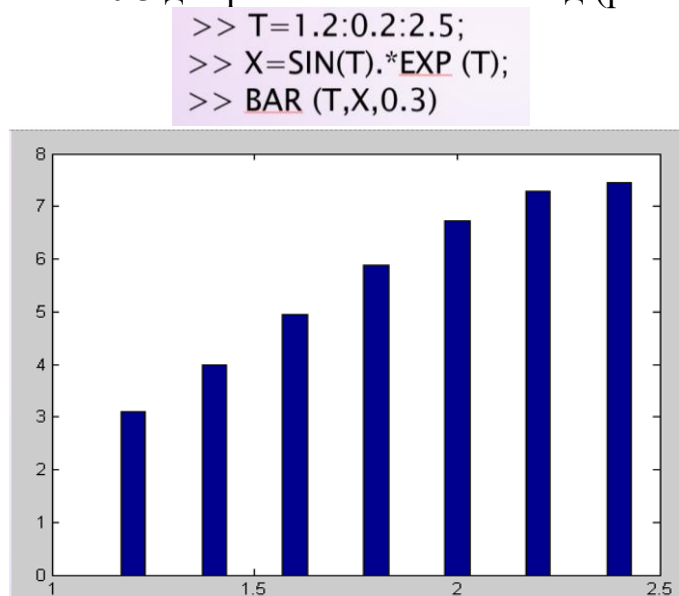


Рис. 67. Стовбцева діаграма, зі значенням аргумента ширини стовбця рівним 0.3

У випадку, якщо необхідно побудувати діаграму без проміжків між стовбцями треба значення аргументу ширини стовбця зробити рівним 1 (одиниці) (рис. 68):

```
>> BAR(T,X,1)
```

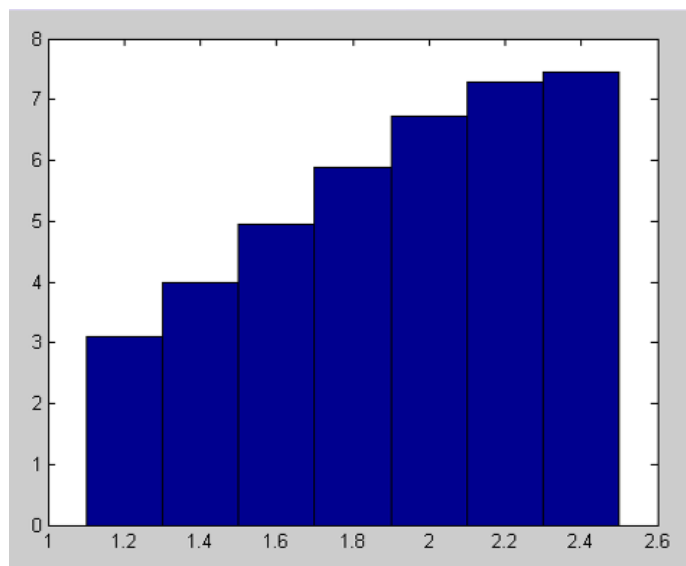


Рис. 68. Стовбцева діаграма, зі значенням аргумента ширини стовбця рівним 1

Розглянемо приклади побудови об'ємних діаграм, для побудови яких застосовується функція `bar3`. На рисунку 69 представлено об'ємні діаграми із значенням аргументу ширини стовбця рівним 0, рівним 0.3 та рівним 1:

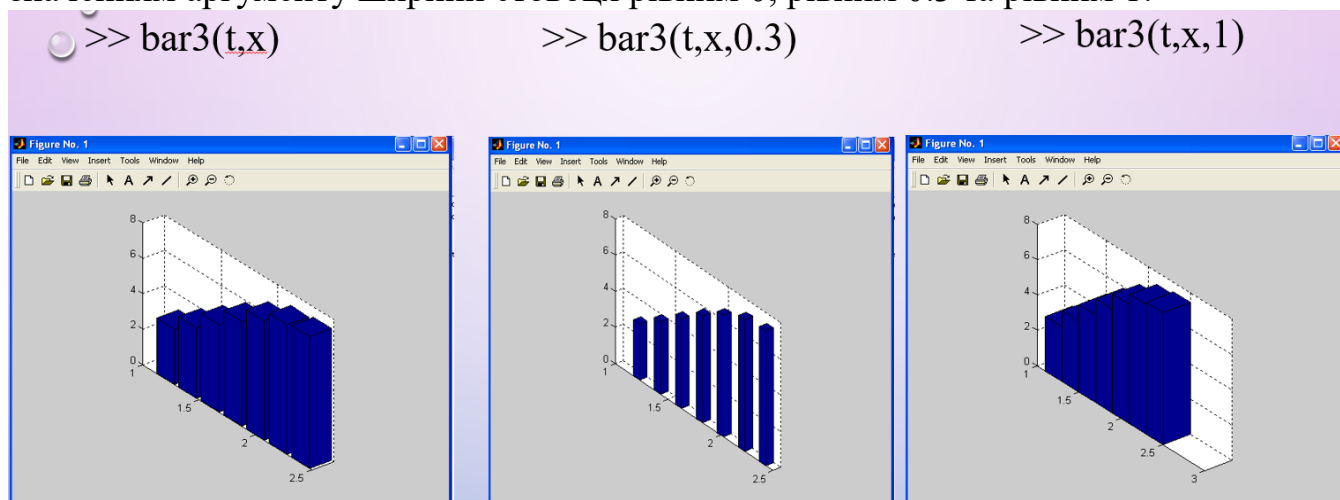


Рис. 69. Об'ємні діаграми із значенням аргументу ширини стовбця рівним 0, рівним 0.3 та рівним 1

Розглянемо приклади виконання завдань.

**Завдання 1:** Скласти таблицю, яка буде показувати кількість студентів, які навчаються за різними формами фінансування. Побудувати кругові діаграми процента студентів, які навчаються за контрактом і по держзамовленню.

Назва вузу	Кількість студентів	Навчається за контрактом	Навчається за держзамовленням	% студентів за контрактом до загальної кількості	% студентів за держзамовленням до загальної кількості
------------	---------------------	--------------------------	-------------------------------	--	---

Зразок виконання завдання:

1) Підготовка інформації, необхідної для виконання завдання в середовищі MATLAB. Заповнення таблиці даними. Кількість рядків повинна бути не менше чотирьох:

Назва вузу	Кількість студентів	Навчається за контрактом	Навчається за держзамовленням	% студентів за контрактом до загальної кількості	% студентів за держзамовленням до загальної кількості
КЕІ	500	400	100	$=(400/500)*100$	$=(100/500)*100$
КНУ	800	500	300	$=(500/800)*100$	$=(300/800)*100$
КПУ	400	200	200	$=(200/400)*100$	$=(200/400)*100$
ДЮІ	700	650	50	$=(650/700)*100$	$=(50/700)*100$

2) Введення матриці, яка складається з двох стовбців (кількість студентів і % студентів, які навчаються за контрактом) і чотирьох інформаційних рядків:

```
>>clear
```

```
>>stud=[500 (400/500)*100; 800 (500/800)*100; 400 (200/400)*100; 700 (650/700)*100]
```

```
stud =
```

```
500.0000 80.0000
```

```
800.0000 62.5000
```

```
400.0000 50.0000
```

```
700.0000 92.8571
```

3) Побудова кругової діаграми:

```
>>pie3(stud)
```

Підписи виконати з допомогою команди меню Insert у вікні Figure 1 (рис. 70):

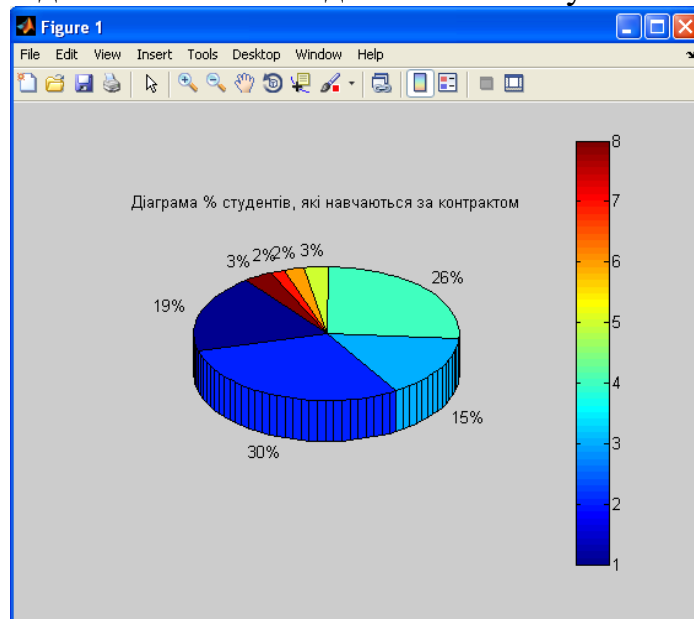


Рис. 70. Графік із затосуванням підписів

**Завдання 2:** Скласти звіт про витрати посадкового матеріалу. Побудувати стовбцеву діаграму загальних витрат матеріалу (план і факт).

1) Заповнення таблиці даними:

Назва матеріалу	Посадкова площа	Витрати на 1м <sup>2</sup>		Загальні витрати	
		План	Факт	План	Факт
1	500	2	3	$=500*2$	$=500*3$



2	800	3	3	=800*3	=800*3
3	300	5	4	=300*5	=300*4
4	720	3	4	=720*3	=720*4
5	150	2	4	=150*2	=150*4

2) Введення матриці, яка складається з двох стовбців (План і Факт) і п'яти інформаційних рядків:

```
>>pf=[500*2 500*3; 800*3 800*3; 300*5 300*4; 720*3 720*4; 150*2 150*4]
pf =
```

```
1000    1500
2400    2400
1500    1200
2160    2880
300      600
```

3) Побудова графіку з підписами:

```
>>bar(pf)
>>title('Загальні витрати матеріалу')
>>xlabel('Вид матеріалу')
>>ylabel('Витрати матеріалу')
>>legend('План', 'Факт')
>> bar(pf)
```

На рисунку 71 представлено графік стовбцевої діаграми з використанням підписів.

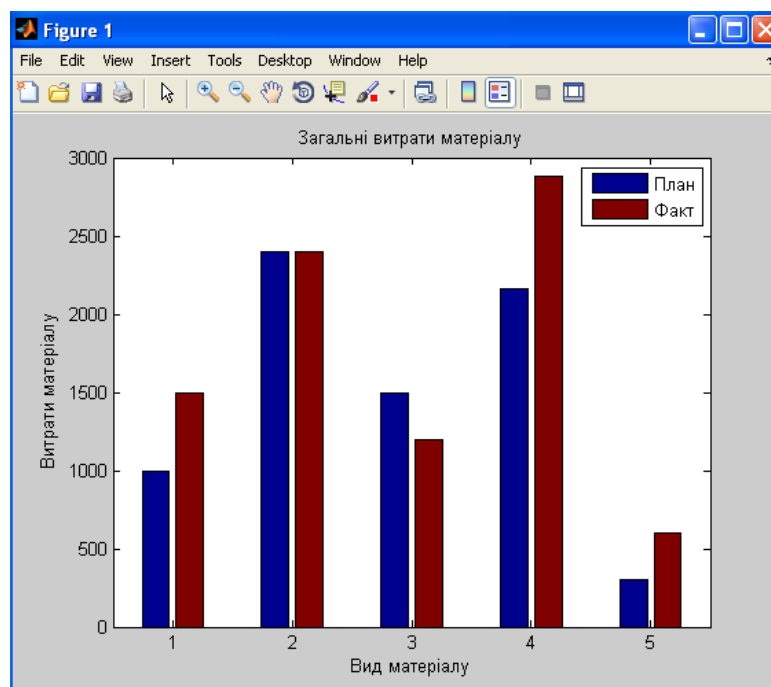


Рис. 71. Стовбцева діаграма загальних витрат матеріалу

## Індивідуальні завдання за темою: «Кругові і стовбцеві діаграми»

### Варіанти індивідуальних завдань

1. Скласти таблицю, в якій зазначити рекламодавців на каналі 1+1 з кількістю рекламних роликів за 1 тиждень. Розрахувати суму затрачених ними

коштів на рекламу. Якщо підсумок складає більше 100000 грн. рекламодавець отримує скидку у розмірі 8%. Побудувати кругову діаграму для кожного рекламодавця від загальних витрат.

Рекламодавець	Категорія реклами	Затрати		Підсумок з урахуванням знижки	% до загальних витрат
		Час хв.	Ціна грн./хв.		

2. Скласти таблицю товарного експорту та імпорту в Україні (по регіонах) за 2020р. Побудувати гістограму різниці між вартістю експорту та імпорту, за регіонами.

Назва регіону	Експорт		Імпорт		Різниця між експортом та імпортом
	Вартість (млрд. грн.)	% до підсумку	Вартість (млрд. грн.)	% до підсумку	

3. Побудуйте таблицю потреб в плодоовочевій продукції по Дніпропетровській області на 2021р. Розрахуйте загальні потреби. Побудувати гістограму потреби продукту на 2021р.

№ з/п	Назва продукту	Мінімальна норма споживання на душу населення на 2021 р. (кг)	Чисельність населення в 2021 р.	Потреба на 2021 р. (кг)
-------	----------------	---	---------------------------------	-------------------------

4. Скласти таблицю товарного експорту та імпорту в Україні (по регіонах) за 2021р. Побудувати кругові діаграми відсотку експорту та відсотку імпорту по регіону до підсумку по регіонах.

Назва регіону	Експорт		Імпорт		Різниця між вартістю експорту та імпорту (млрд. грн.)
	Вартість (млрд. грн.)	% до підсумку	Вартість (млрд. грн.)	% до підсумку	

5. Скласти таблицю товарного експорту та імпорту в Україні (по регіонах) за 2021р. Побудувати кругову діаграму відсотку імпорту по регіону до підсумкової.

Назва регіону	Експорт		Імпорт		Різниця між вартістю експорту та імпорту (млрд. грн.)
	Вартість (млрд. грн.)	% до підсумку	Вартість (млрд. грн.)	% до підсумку	

6. Скласти таблицю прибутків приватних нотаріусів окремо по кожному району міста. Розрахувати суму надходжень до бюджету. Податок складає 20% від прибутку, якщо прибуток менше або дорівнює 1000 грн., 25% - прибуток більше 1000 грн., але менш 10000 грн., 30% якщо прибуток більше 10000 грн. Побудувати гістограму надходжень до бюджету для різних районів міста.

Назва району всього нотаріату	Кількість приватних нотаріусів	Прибуток нотаріуса	Ставка податку на прибуток	Сума надходжень у вигляді податку
-------------------------------	--------------------------------	--------------------	----------------------------	-----------------------------------

7. Заповніть таблицю рівня забезпеченості ринку плодоовочевою продукцією по області в 2021р. Якщо нормативна потреба більше або дорівнює фактичній реалізації в стовпчику «Вид забезпечення» повинно бути «достатній», в іншому випадку «недостатній». Побудувати гістограму відхилення забезпеченості ринку області в плодово-овочевій продукції.

№ з/п	Назва продукту	Нормативна потреба (тис., т)	Фактична реалізація в 2021 р., тис. т	Відхилення, тис. т	Вид забезпечення
-------	----------------	------------------------------	---------------------------------------	--------------------	------------------

8. Побудувати таблицю фактичних норм споживання плодоовочевої продукції в 2021р. Побудувати гістограму продуктів спожитих недостатньо на душу населення. Якщо фактичне споживання продукту менше ніж норма мінімальна в стовпчику «Вид відхилення» вивести «недостатній» в іншому випадку «достатній».

№ з/п	Назва продукту	Норма споживання на душу		Вид відхилення	Відсоток відхилення
		Мінімальна	Фактичне споживання		

9. Побудувати таблицю облікових даних про продаж товарів у вересні 2020р., якщо вартість покупки проданого товару перевищує 1000 грн. надається скидка 5%. Побудувати гістограму реалізації товарів протягом місяця.

Дата	Продавець	Товар			Вартість з урахуванням знижки
		Назва	Кількість	Ціна	

10. Побудувати таблицю даних про продаж товарів на ринку різними фірмами. Розрахуйте частину ринку, яку займає кожна окрема фірма у % до загальної їх кількості. Якщо вартість більше 1000 грн. скидка 8%. Побудувати кругову діаграму частини ринку кожної фірми до загальної.

Назва фірми	Назва продукції	Обсяги продажу, грн.			Частка ринку до загальної, %
		Кількість	Ціна	Вартість з урахуванням знижки	

11. Побудувати таблицю забезпеченості населення товаром за певний період. Обчислити забезпеченість у грошовому виразі на 1 чол. кожного району. Якщо забезпеченість населення товарами в грошовому виразі на 1 чол. більше 10000 грн. - тип забезпеченості високий, менше або рівно 10000 грн. та більше 1000 грн. - середній, в інших випадках - низький. Побудувати гістограму забезпечення населення товаром для районів міста.

Район	Кількість мешканців	Товари першої необхідності				Забезпеченість населення товаром (грн./1чол.)	Тип забезпеченості
		Назва	Кількість	Ціна	Вартість		

12. Скласти таблицю прибутковості виробництва цементу різних марок Криворізьким ЦГК. Якщо прибуток більше 50000 грн. тип прибутку великий,

якщо менше, або рівно 50000 грн. та більше 10000 грн. - середній, в інших випадках - низький. Побудувати кругову діаграму прибутку від марки цементу.

Марка цементу	Обсяг виробництва, т	Собівартість т.	Ціна продажу	Прибуток	Тип прибутку	% прибутку від марки цементу
---------------	----------------------	-----------------	--------------	----------	--------------	------------------------------

13. Скласти таблицю прибутковості виробництва товарів заводом, окремо для кожного цеху. Якщо прибуток цеху складає більше 50000 грн. в стовпчику Тип прибутку повинно бути – «великий», якщо він менше, або складає 50000 грн. та більше 20000 грн. в стовпчику Тип прибутку повинно бути – «середній», в інших випадках – «низький». Побудувати гістограму прибутку від виробництва кожного цеху заводу.

Назва цеху	Вид продукції	Обсяг виробництва	Собівартість виробництва одиниці продукції	Ринкова ціна продукту	Прибуток від виробництва	Вид прибутку
------------	---------------	-------------------	--	-----------------------	--------------------------	--------------

14. Скласти таблицю виконання замовлень по будівництву споруд будівельними компаніями міста. Якщо кількість замовлень більше або рівно кількості побудованих будівель в стовпчику Виконання замовлень повинно бути - «виконано», якщо ні - «не виконано». Побудувати кругову діаграму виконання від загальної кількості побудованих будівель.

№ з/п	Назва будівельної компанії	Кількість замовлень	Побудовано будівель	% виконання від загальної кількості	Виконання замовлень
-------	----------------------------	---------------------	---------------------	-------------------------------------	---------------------

15. Розрахувати відсоткове співвідношення розглянутих справ та позовів. Якщо розглянуто справ 95% та більше відсотків у стовпчику Тип потрібно вивести - високий, якщо менше 95% та більше 65% - середній, в інших випадках - низький. Побудувати кругову діаграму відсотка розглянутих справ від кількості судових позовів по місту за типом.

№ з/п	Назва суду	Судові позови				% розглянутих справ від кількості судових позовів по місту за типом
		Кількість судових позовів	Розглянуто справ	% розглянутих справ	Тип	

16. Побудувати таблицю прибутків нотаріальної контори за певний період. Розрахувати прибуток, який отримує контора окремо по кожній послугі. Якщо прибуток від послуги перевищує 1000 грн. надається знижка у розмірі 8%. Побудувати гістограму прибутку від кожної послуги з урахуванням знижки.

Назва послуги	Вартість однієї виконаної послуги	Кількість наданих послуг за рік	Прибуток від кожного виду послуг	Прибуток з урахуванням наданої знижки
---------------	-----------------------------------	---------------------------------	----------------------------------	---------------------------------------

17. Скласти таблицю, яка б показувала витрати на виробництво та надходження від продажу товарів компанії Samsung на ринку України. Розрахувати прибуток або збиток фірми та побудувати гістограму. Якщо

різниця між надходженням від продаж і затратами позитивна в стовпчику Прибуток буде його значення, а в стовпчику збиток 0, в іншому випадку навпаки.

№ з/п	Назва товару	Затрати		Надходження від продажу	Прибуток	Збиток
		Виробництво	Реклама			

18. Побудуйте таблицю даних про збут комп'ютерів однією комп'ютерною фірмою. Розрахуйте вартість по кожному покупцю і загальний прибуток фірми. Фірма надає знижку у розмірі 5% покупцям, які придбали обладнання в термін з 1 по 10 число місяця. Побудуйте гістограму продаж впродовж місяцю.

Дата	Покупець	Назва товару	Кількість	Собівартість	Ціна	Надходження від продажу	Прибуток
------	----------	--------------	-----------	--------------	------	-------------------------	----------

19. Скласти таблицю та побудувати гістограму, яка б показувала динаміку часток товарів підприємства на ринку городу. Якщо середнє значення більш 55% в стовпчику Частка повинно бути «велика», якщо менш 55% та більше 25% - «достатня», в інших випадках – «низька».

Товар (код)	Обсяги продажу на ринку, грн.				Середньо зважене значення, %	Частка
	2020 р.		2021 р.			
	I півріччя	II півріччя	I півріччя	II півріччя		

20. Скласти звіт про роботу суду. Побудувати кругову діаграму розглянутих справ кожного судді.

Прізвище судді	Кількість позовів	Розглянуті справи		У судовому розгляді відмовлено	
		Кількість	% до кількості позовів	Кількість	% до кількості позовів

### Графіки параметрично-заданих функцій

Щоб побудувати графік параметричної функції необхідно мати дві функції  $x(t)$  і  $y(t)$ . Параметрична функція задається у вигляді  $y(x)$ .

Для побудови графіків функцій, заданих параметрично, необхідно спочатку згенерувати вектор значень аргументів. Потім обчислити значення функції, і заповнити їх вектори, які треба використовувати в якості аргументів функції *plot*.

Таким чином, для побудови графіка такої функції необхідно виконати наступні дії:

- задати межі і крок зміни аргументу  $t$ ;
- обчислити значення функцій  $x(t)$  і  $y(t)$ ;
- використати функцію `plot(...)` в одному із її варіантів.

Наприклад, побудуємо графіки функцій  $x(t)$  і  $y(t)$  на відрізку від 0 до  $2\pi$ .

```
>> t=0:0.01:2*pi;
>> x=0.5*sin(t);
>> y=0.7*cos(t);
>> plot(x,y)
```

Використовуючи функцію *plot* побудуємо графік параметрично заданої функції (рис. 72):

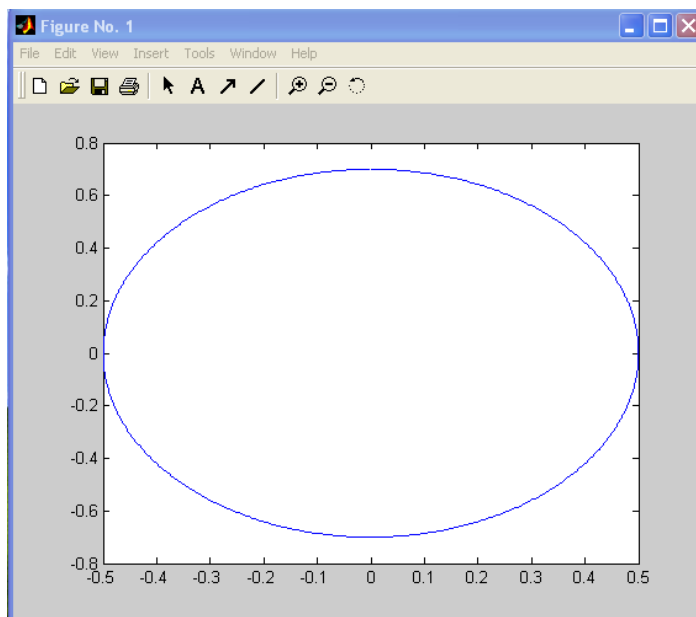


Рис. 72. Графік параметрично заданої функції

**Завдання 1.** Необхідно побудувати графік функції  $y(x)$ , яка задана параметричними формулами:  $x=4e^{-0,05t} \sin t$  та  $y=0,2e^{-0,1t} \sin 2t$ . Задати межі і крок зміни параметра  $t$  від 0 до 50 із кроком 0.1 (рис. 73):

```
Command Window
>> t=0:0.1:50;
>> x=4*exp(-0.05*t).*sin(t);
>> y=0.2*exp(-0.1*t).*sin(2*t);
>> plot(x,y)
>> title('Параметрична функція x=4*exp(-0.05t)*sin(t); y=0.2*exp(-0.1t)*sin(2t)')
>> grid
```

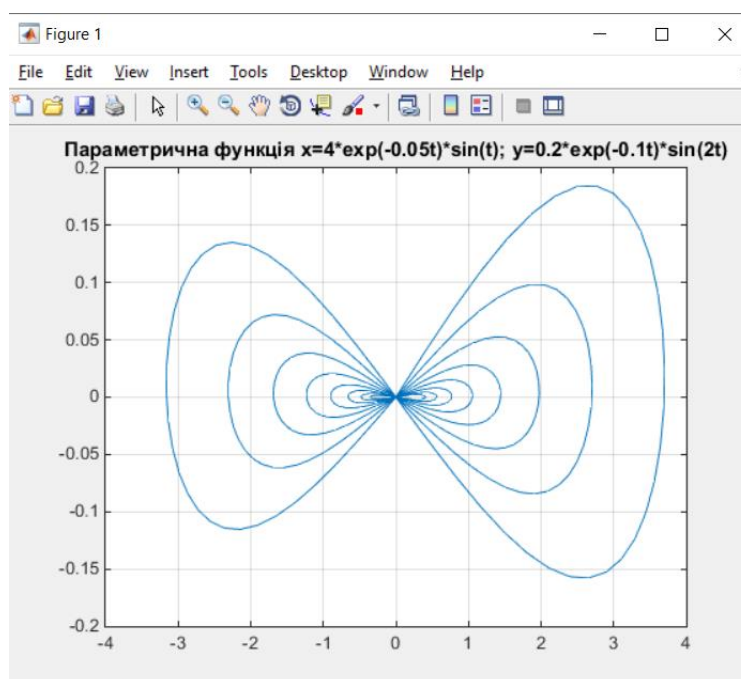


Рис. 73. Графік параметрично заданої функції

## Індивідуальні завдання за темою «Графіки параметрично заданих функцій»

### Варіанти індивідуальних завдань:

1. Використовуючи в якості вхідних функцій  $x(t) = \sin(t)$  і  $y(t) = \cos^3(t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком  $0,1$ .
2. Використовуючи в якості вхідних функцій  $x(t) = \sin(t)$  і  $y(t) = 10 \cdot \cos(t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком  $0,1$ .
3. Використовуючи в якості вхідних функцій  $x(t) = \cos(10 \cdot t)$  і  $y(t) = e^{-t} \cdot \sin(10 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком  $0,03$ .
4. Використовуючи в якості вхідних функцій  $x(t) = e^{-t} \cdot \cos(10 \cdot t)$  і  $y(t) = e^{-t} \cdot \sin(10 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
5. Використовуючи в якості вхідних функцій  $x(t) = \sqrt{t} \cdot \cos(10 \cdot t)$  і  $y(t) = e^{-t} \cdot \sin(10 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
6. Використовуючи в якості вхідних функцій  $x(t) = \sqrt{t} \cdot \cos^2(5 \cdot t)$  і  $y(t) = (1-t) \cdot \sin(t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
7. Використовуючи в якості вхідних функцій  $x(t) = \sqrt{t} \cdot \cos(t)$  і  $y(t) = \sin^2(t) \cdot \sin(10 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
8. Використовуючи в якості вхідних функцій  $x(t) = \arctg(t) \cdot \cos(10 \cdot t)$  і  $y(t) = \sin^2(t) \cdot \sin(5 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
9. Використовуючи в якості вхідних функцій  $x(t) = \cos^2(t) \cdot \cos(10 \cdot t)$  і  $y(t) = \sin^2(t) \cdot \sin(5 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
10. Використовуючи в якості вхідних функцій  $x(t) = \cos^2(t) \cdot \cos(10 \cdot t)$  і  $y(t) = \sin^2(t) \cdot \sin(10 \cdot t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 1]$  з кроком  $0,03$ .
11. Використовуючи в якості вхідних функцій  $x(t) = \cos^2(t) \cdot \cos(\cos(10 \cdot t))$  і  $y(t) = \sin^2(t) \cdot \sin(\sin(10 \cdot t))$ , за допомогою команди *plot* побудувати графік

параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком 0,01.

12. Використовуючи в якості вхідних функцій  $x(t) = \sin^2(t) \cdot \cos(\cos(10 \cdot t))$  і  $y(t) = \sin^2(t) \cdot \sin(\sin(10 \cdot t))$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком 0,01.

13. Використовуючи в якості вхідних функцій  $x(t) = t^2 \cdot \cos(\cos(5 \cdot t))$  і  $y(t) = t^2 \cdot \sin(\sin(5 \cdot t))$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком 0,01.

14. Використовуючи в якості вхідних функцій  $x(t) = t^2 \cdot \cos(\cos(5 \cdot t))$  і  $y(t) = t^2 \cdot \cos(\sin(5 \cdot t))$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком 0,01.

15. Використовуючи в якості вхідних функцій  $x(t) = t^2 \cdot \cos(\cos(5 \cdot t))$  і  $y(t) = \sin^2(t) \cdot \cos(t)$ , за допомогою команди *plot* побудувати графік параметричної функції  $y(x)$  за умови, що  $t$  змінюється на інтервалі  $[0, 2\pi]$  з кроком 0,01.

### Високорівнева графіка

#### Побудова циліндра у вигляді трьохмірної фігури.

#### Виведення графіків в різні графічні вікна за допомогою команди *figure*

Працюючи з графічними об'єктами виникають ситуації, коли необхідно розміщувати кожен з графіків в окреме графічне вікно. Команди функцій побудови графіків автоматично відкривають нове вікно зображення, якщо до цього його не було на екрані. Якщо ж воно існує, то команда використовує його за замовчуванням. Для відкриття нового вікна і вибору його за замовчуванням, необхідно набрати *figure*. Для того, щоб зробити існуюче вікно поточним треба в дужках вказати номер вікна: *figure (n)*

де  $n$  – це номер в заголовку вікна. У цьому випадку результати всіх наступних команд будуть виводитися в це вікно. Якщо треба зберегти кілька графіків, то перед виведенням нового графіка необхідно відкрити нове вікно з наступним номером вікна, який вказується в дужках.

Для побудови циліндра у вигляді тривимірної фігури застосовується функція *cylinder*:

$[X, Y, Z] = \text{cylinder}(R, N)$ , яка створює масиви  $X$ ,  $Y$  і  $Z$ , які описують циліндричну поверхню з радіусом  $R$  і числом вузлових точок  $N$  для подальшої побудови за допомогою функції *surf*( $X, Y, Z$ ).

Для побудови графіка використовується функція *surf*, яка будує каркасну поверхню і заливає кожен клітинку поверхні певним кольором. Для побудови каркасної поверхні також можна використовувати функцію *mesh*.

Приклад побудови об'ємного циліндра (рис. 74):



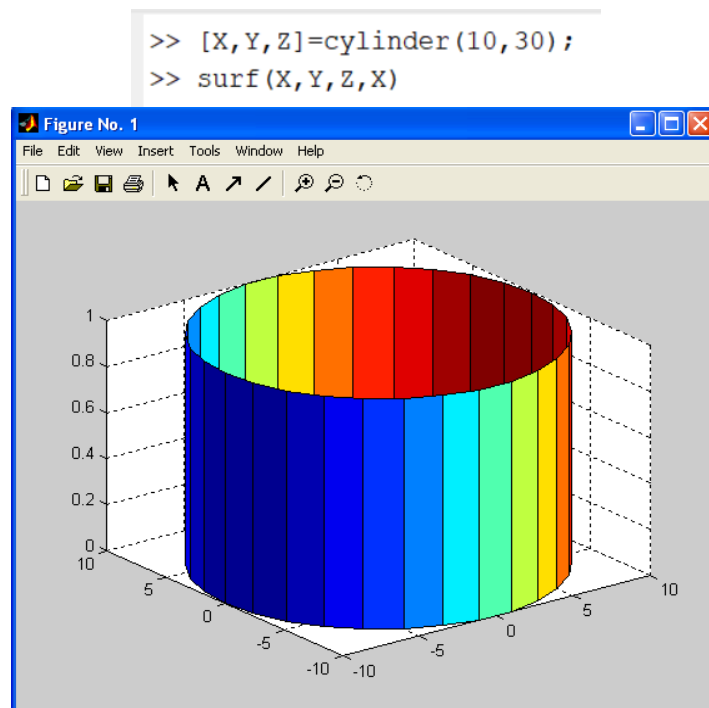


Рис. 74. Графік об'ємного циліндра.

Забарвлення кольором визначено вектором X

Вид циліндра істотно залежить від графічної команди, що використовується для його побудови. Команда `surf` дає можливість задати функціональне забарвлення з кольором, який визначається вектором X, який задається четвертим параметром і робить уявлення циліндра досить наочним.

У команді `surf` змінимо функціональне забарвлення кольором, який визначимо вектором Y, який задаємо четвертим параметром (рис. 75):

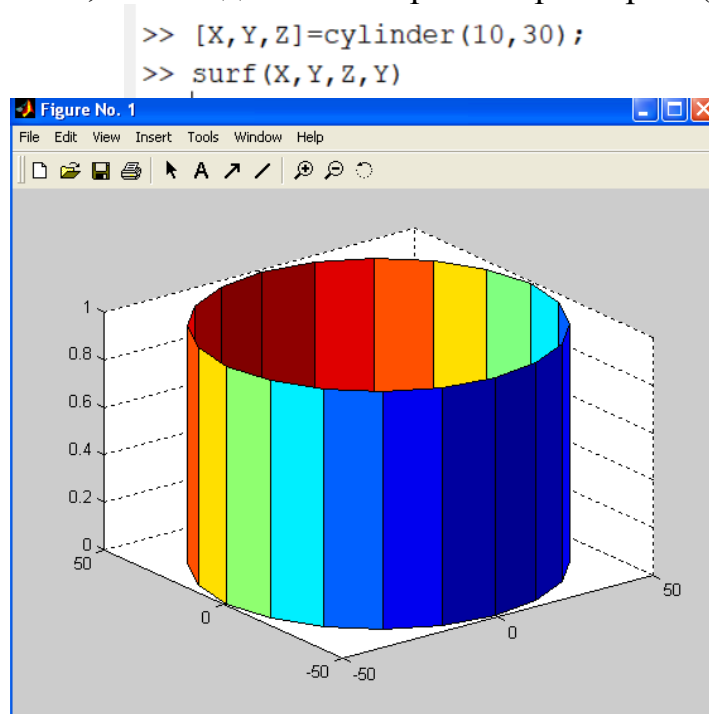


Рис. 75. Графік об'ємного циліндра.

Забарвлення кольором визначено вектором Y

Потім у команді `surf` змінимо функціональне забарвлення кольором, який визначимо вектором Z, який задаємо четвертим параметром (рис.76):

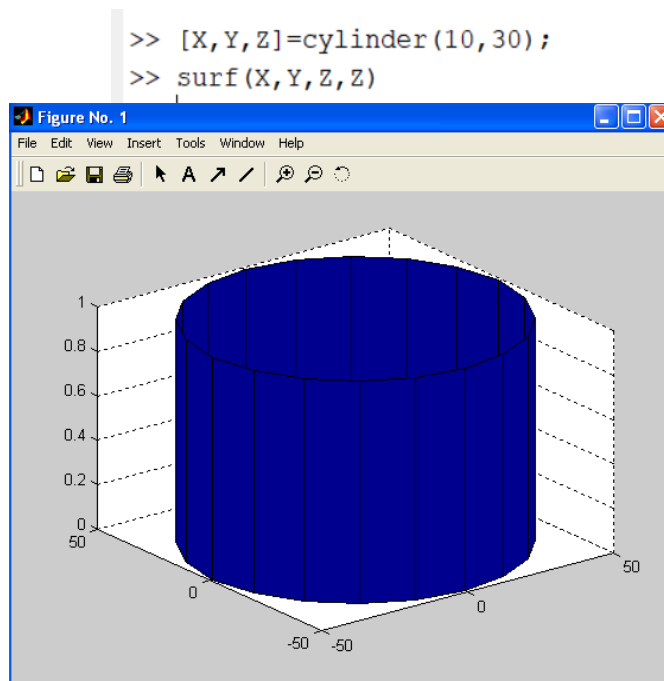


Рис. 76. Графік об'ємного циліндра.

Забарвлення кольором визначено вектором Z

**Завдання 1.** Для рівняння  $f=3+\cos 2t$  побудувати графік, використовуючи функцію `surf`. Значення змінної  $t$  змінюється в інтервалі  $[0, 2\pi]$  з кроком 0.1

Кроки виконання завдання:

По перше необхідно створити масив даних з інтервалом 0.1. По друге потрібно створити масиви X, Y та Z, що описують циліндричну поверхню. У робочому вікні введемо команду створення масиву  $t$ , потім використовуючи дані побудованого масиву  $t$  використовуючи команду `cylinder` створимо масиви X, Y, Z. Використовуючи підготовлену інформацію побудуємо графік за допомогою функції `surf` (рис. 77):

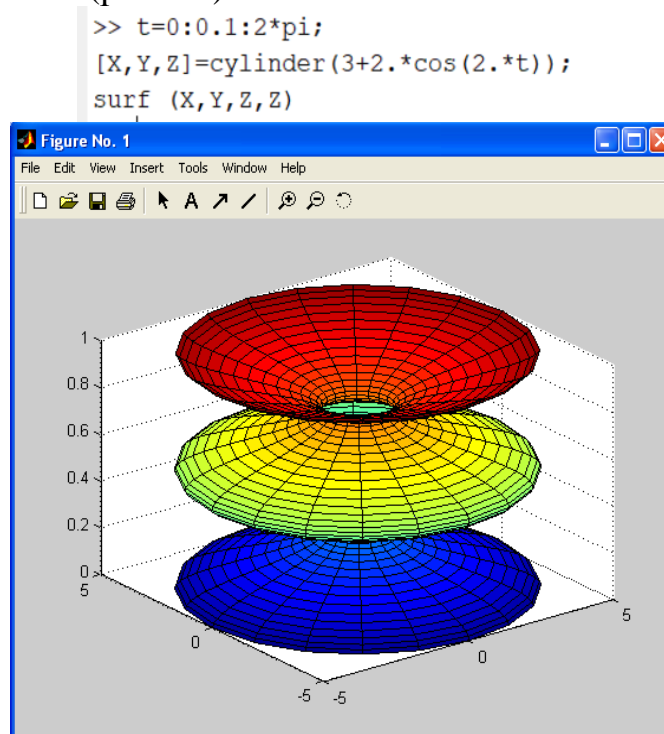


Рис. 77. Графік рівняння  $f=3+\cos 2t$  за допомогою функції `surf`

**Завдання 2:** Для рівняння  $f=2+4\cos t$  побудувати графік, використовуючи функцію surf. Значення  $t$  змінюється в інтервалі  $[0, 2\pi]$  з кроком 0.1

Кроки виконання завдання:

- 1) Створити масив даних з інтервалом 0.1.
- 2) Створити масиви X, Y та Z, що описують циліндричну поверхню.
- 3) Побудувати графік з допомогою функції surf (рис. 78):

```
>> t=0:0.1:2*pi;
>> [X,Y,Z]=cylinder(2+4.*cos(t));
>> surf (X,Y,Z,X)
```

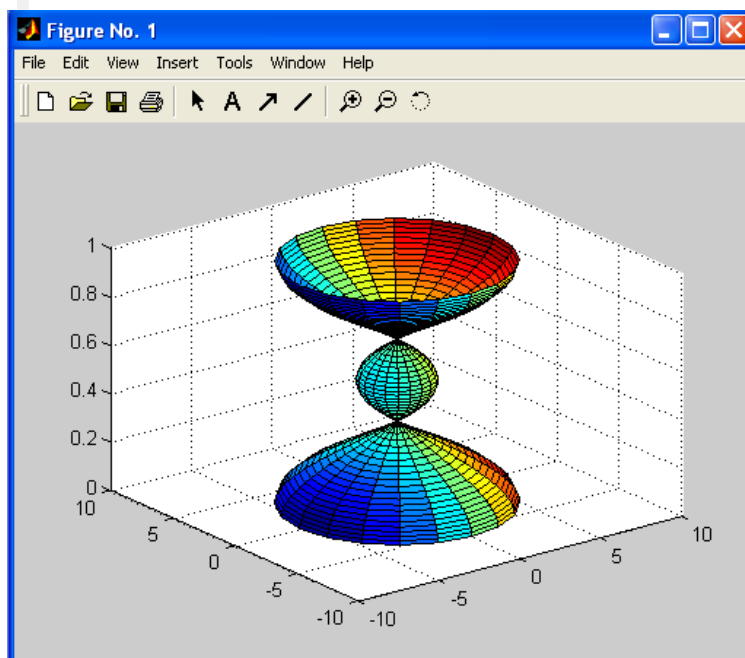


Рис. 78. Графік рівняння  $f=2+4\cos t$  за допомогою функції surf

**Завдання 3:** Побудувати графіки функції  $z=2*\sin(t)+t$  на області визначення  $t=0:\pi/7:5*\pi$ . Виведення графіків функцією surf здійснювати в окремі графічні вікна за допомогою команди figure.

У робочому вікні пакета MatLab здійснити необхідно задати область визначення функції  $t=0:\pi/7:5*\pi$ . Потім виконати формування двомірних масивів X, Y, Z, які описують циліндричну поверхню.

Далі за допомогою команди figure задати номер графічного вікна. Номер вікна задавати в дужках, тобто figure(1).

Виконати виведення першого графіка в графічне вікно № 1 за допомогою функції surf.

```
>> t=0:pi/7:5*pi;
>> [X,Y,Z]=cylinder(2.*sin(t)+t);
>> figure(1)
>> surf(X)
>> |
```

У вікні № 1 виведено графік отриманої матриці X (рис. 79):

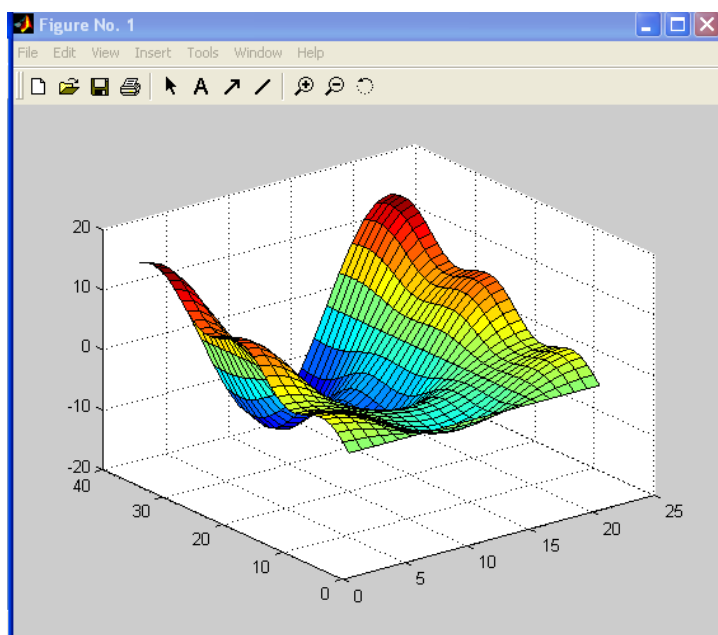


Рис. 79. Графік матриці X

Виконати аналогічні дії, але виведення другого графіка буде у графічному вікні № 2 за допомогою функції surf.

```
>> t=0:pi/7:5*pi;
>> [X,Y,Z]=cylinder(2.*sin(t)+t);
>> figure(2)
>> surf(Y)
>> |
```

У вікні № 2 виведено графік отриманої матриці Y (рис. 80):

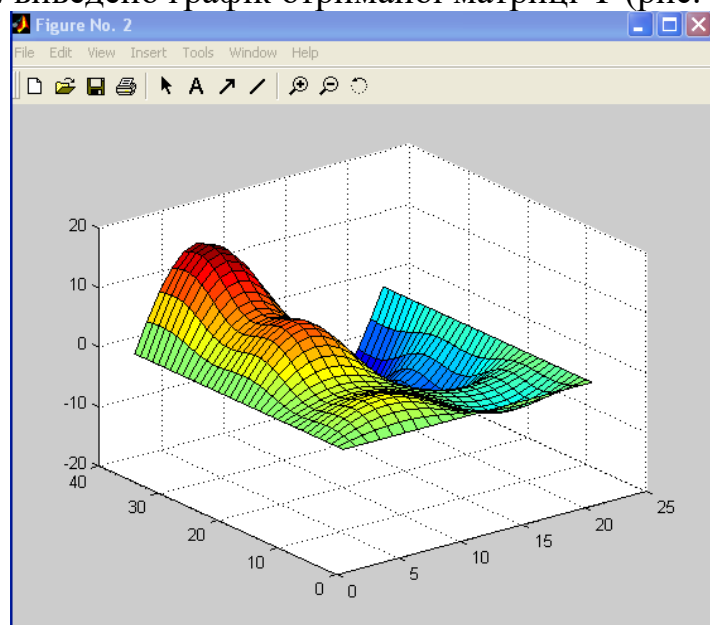


Рис. 80. Графік матриці Y

Виконавши аналогічні дії, третій графік вивести в графічне вікно № 3

```
>> t=0:pi/7:5*pi;
>> [X,Y,Z]=cylinder(2.*sin(t)+t);
>> figure(3)
>> surf(Z)
>> |
```

У вікні №3 виведено графік отриманої матриці Z (рис. 81):

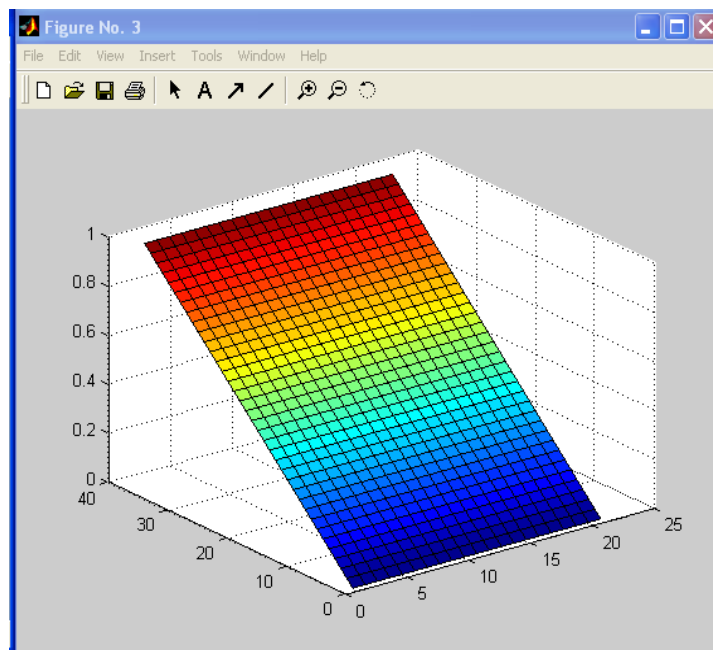


Рис. 81. Графік матриці Z

Аналогічно четвертий графік виведемо в графічне вікно № 4

```
>> t=0:pi/7:5*pi;
>> [X,Y,Z]=cylinder(2.*sin(t)+t);
>> figure(4)
>> surf(X,Y,Z)
>> |
```

У вікно № 4 виведено графік отриманих матриць X, Y, Z (рис. 82):

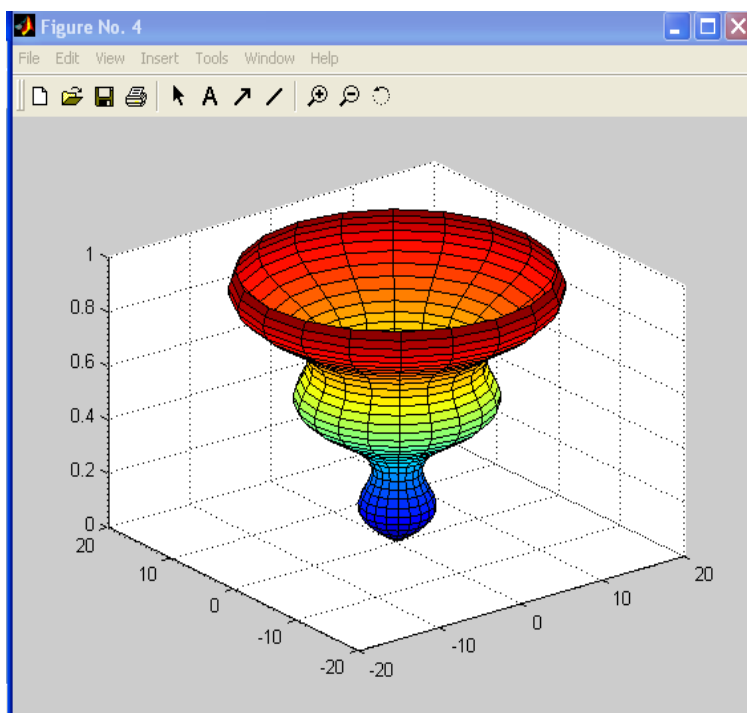


Рис. 82. Графік матриць X, Y, Z

**Індивідуальні завдання за темою: «Побудова циліндра. Виведення графіків в різні графічні вікна»**

**Варіанти індивідуальних завдань**

1.  $z = (t-2)^2 + 3t^2$

2.  $z = t^3 + 3t^2 - 15t - 12t$

3.  $z = t^2 + 4t - t^2 - 6t - 2t$

4.  $z = \sin t \cdot \cos t$

5.  $z = (t+t)^2$

6.  $z = (3t-t)^2$

7.  $z = 3 \cdot \sin(t)^2$

8.  $z = \sqrt{5 \cdot t}$

9.  $z = \sin t + \sin t$

10.  $z = \sin t + t^3$

11.  $z = \sin(t+t)$

12.  $z = t^2 + t^3$

13.  $z = t^2 + t$

14.  $z = \cos t \cdot \cos 2t$

15.  $z = 3t^2 + 5t$

16.  $z = 3 \cdot |t| + t$

17.  $z = 2t^3 + 4t$

18.  $z = \cos t + t^3$

19.  $z = 5t^2 + |t|$

20.  $z = \cos t + \cos t$

21.  $z = \cos t + \sin t$

22.  $z = \tan t + \cos t$

23.  $z = \tan t + \sin t$

24.  $z = \tan t + t^3$

25.  $z = \tan t + \tan t$

**Високорівнева графіка. Побудова сфери як трьохмірної фігури.**

Для формування масивів X, Y та Z - координат точок сфери, як тривимірної фігури використовується функція sphere. Команда mesh (X, Y, Z, C) виводить на екран сітчасту поверхню значень масиву Z, визначених на безлічі значень масивів X і Y. Кольори вузлів поверхні задаються масивом C.

$[X,Y,Z] = \text{sphere}(N)$  – генерує матриці  $X$ ,  $Y$  та  $Z$  розміру  $(N+1)*(N+1)$  для побудови сфери за допомогою команд `surf (X,Y,Z)` або `surf ( X, Y, Z)` чи `mesh (X, Y, Z)`. Розглянемо приклад застосування функції `sphere` при  $N = 20$ , графік виконаний за допомогою `surf` (рис. 83):

```
>> [X, Y, Z] = sphere (30);
>> surf (X, Y, Z)
```

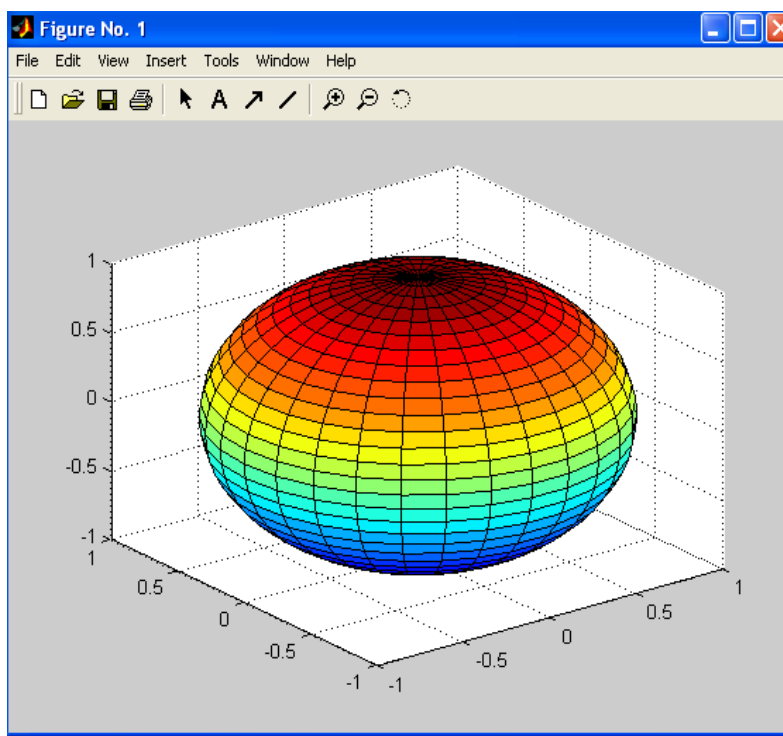


Рис. 83. Графік сфери побудований за допомогою функції `surf`

**Завдання 1.** Для рівняння  $z=5|x+y|+4y^2+3$  побудувати графік, за допомогою функції `mesh` та `surf`. Значення змінних  $x$  та  $y$  необхідно задати.

Розв'язання завдання починається з присвоєння значень вхідним змінним, наприклад,  $x=3$  та  $y=4$ . Далі виконується обчислення значень функції  $z$  та створюються масиви  $X$ ,  $Y$  та  $Z$ , які здійснюють описання сферичної поверхні.

Використовуючи підготовлені дані у вікні `figure (1)` за допомогою функції `mesh` відбувається побудова графіку, зображеного на рисунку 84, а у вікні `figure (2)` функцією `surf` відбувається побудова графіку, який зображений на рисунку 85:

```
>> x = 3;
y = 4;
z = 5*abs (x+y) + 4*y^2 + 3;
[X,Y,Z] = sphere (z);
figure (1)
mesh (X,Y,Z)
figure (2)
surf (X,Y,Z)
```

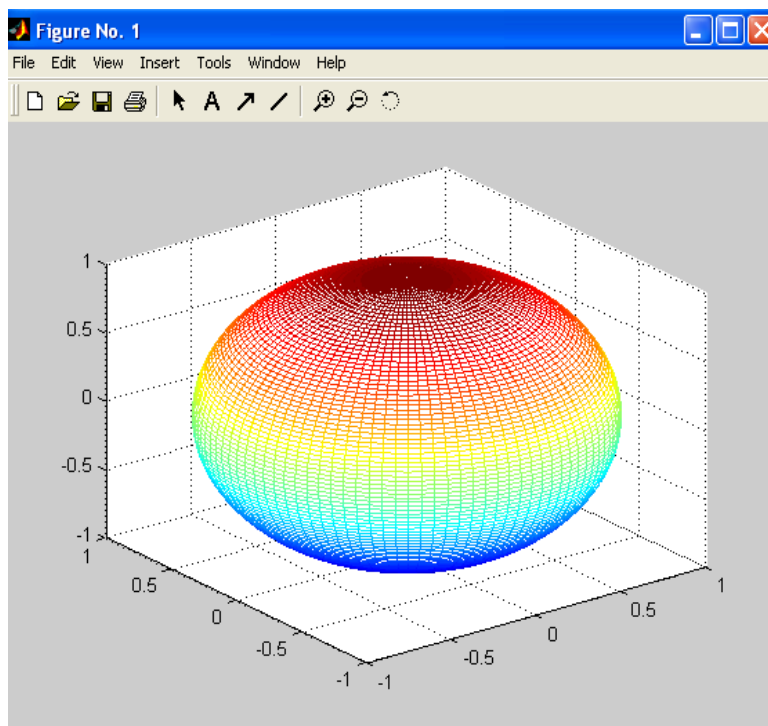


Рис. 84. Графік сфери побудований за допомогою функції mesh

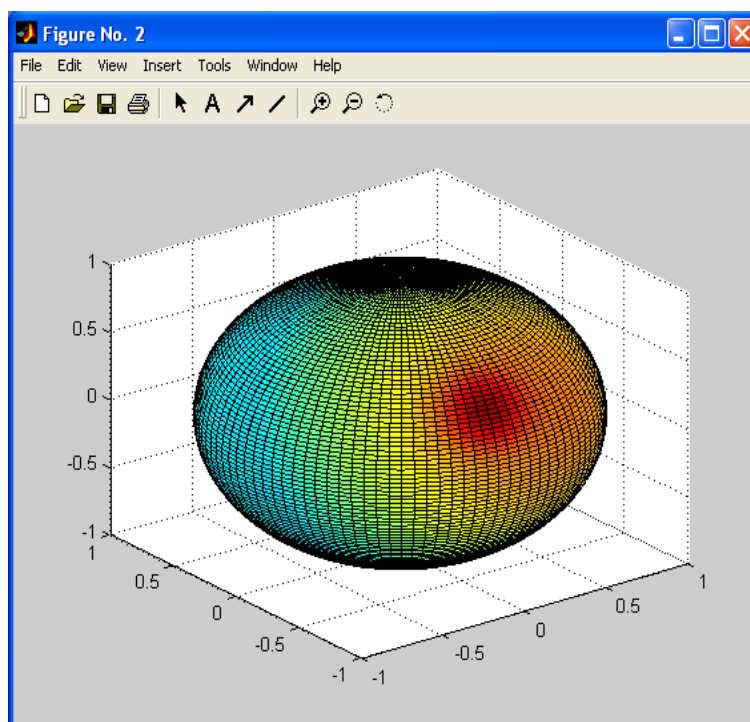


Рис. 85. Графік сфери побудований за допомогою функції surf

**Завдання 2.** Задано рівняння  $z=(x^3+y^3)+(x*y)^2$ . За допомогою функцій mesh та surf побудувати графіки у вікні figure(1) та вікні figure(2).

Вхідним даним  $x$  та  $y$  необхідно присвоїти довільні значення. Далі здійснити обчислення функції  $z$ .

Використовуючи отримані значення функції створюються масиви  $X$ ,  $Y$  та  $Z$ , які описують сферичну поверхню. Функціями mesh та surf будуються графіки сфери (рис. 86, рис. 87), аргументами цих функцій являються сформовані масиви  $X$ ,  $Y$ ,  $Z$ :



```
>> x=3;
>> y=4;
>> z=(x^3+y^3)+(x*y)^2;
>> [X,Y,Z]=sphere(z);
>> figure(1)
>> mesh(X,Y,Z)
>> % графік сфери, побудований функцією mesh у 1-му вікні - рис. 85
>> figure(2)
>> surf1(X,Y,Z)
>> % графік сфери, побудований функцією surf1 у 2-му вікні - рис. 86
>>
```

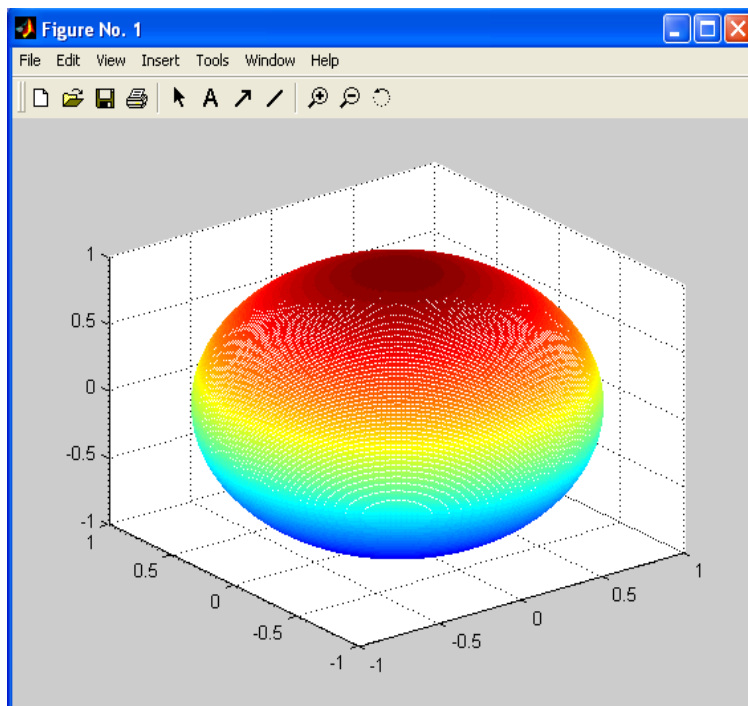


Рис. 86. Графік сфери побудований функцією mesh у вікні figure(1)

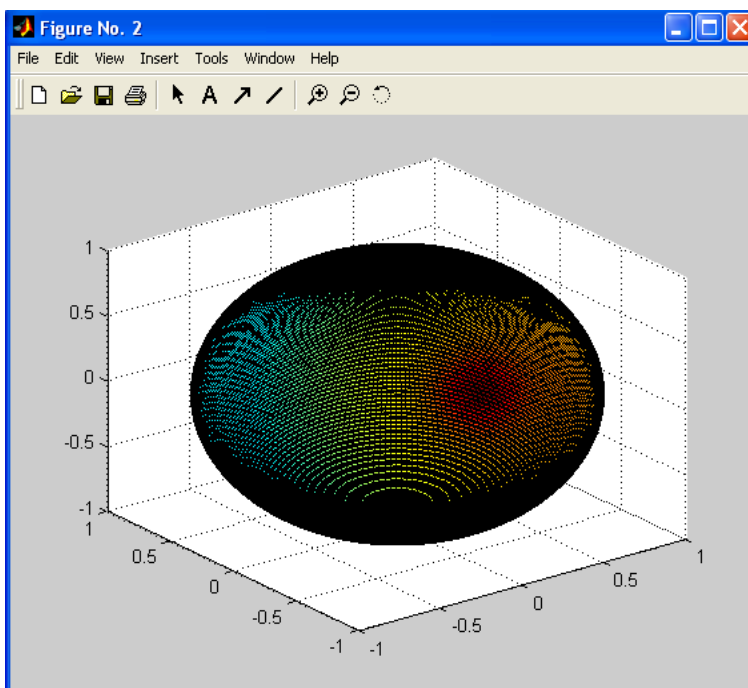


Рис. 87. Графік сфери побудований функцією surf1 у вікні figure(2)

## Виведення різних графіків в одному графічному вікні за допомогою команди subplot

Виникають ситуації, коли необхідно розміщувати в одному графічному вікні кілька координатних осей з різними графіками без накладення їх один на одного. Для цього використовується функція subplot. Вона дозволяє виводити безліч графіків в одному вікні. Синтаксис її: subplot (i, j, n).

Функція subplot розбиває вікно зображень на матрицю, в якій i та j – число підграфіків по вертикалі і горизонталі, а n – номер підграфіка, який треба зробити поточним. Наприклад, щоб представити графічні дані в чотирьох різних підобластях вікна необхідно виконати наступне:

>> subplot(2, 2, 1) – дана команда дає можливість розташувати у вікні figure по горизонталі два підграфіка і по вертикалі два підграфіка, а поточним зробити перший.

Для побудови сфери у вигляді тривимірної фігури застосовується функція sphere:

[X,Y,Z] = sphere(N) – генерує матриці X, Y і Z розміру (N+1)\*(N+1) для подальшої побудови сфери за допомогою команд surf(X,Y,Z) або surfl(X,Y,Z) або mesh(X,Y,Z).

Для побудови графіка використовуємо функцію mesh, яка будує каркасну поверхню.

**Завдання 1.** Побудувати графіки функції  $z=2*\sin(t)+t$  на прямокутній області визначення. Виведення графіків здійснювати в одному графічному вікні. Назви команд, за допомогою яких побудовано графіки, включити до заголовків підграфіків.

Зразок виконання завдання:

1) Задаємо область визначення функції  $t = 0: \pi/7 : \pi$

```
>> t = (0: pi/7 :5*pi);
```

2) Виконаємо формування двомірних масивів X, Y, Z, які описують циліндричну поверхню

```
>> [ X, Y, Z ] = cylinder (2.*sin (t) + t);
```

3) Функція subplot розбиває вікно на певну кількість частин по вертикалі і горизонталі.

Синтаксис subplot(i, j, n),

де i та j – число підграфіків по вертикалі та горизонталі,

а n – номер підграфіка, який треба зробити поточним.

З допомогою функції subplot розбиваємо вікно на дві підобласті по вертикалі і дві підобласті по горизонталі.

Поточними зробимо підобласті по черзі: підобласть 1, потім підобласть 2, далі підобласть3, і підобласть 4

```
>> subplot(2, 2, 1)
```

Результат виконання цієї команди представлено на рис. 88.

З рисунку видно, що для виводу графіків підготовлено вікно і в лівому верхньому куті виділено підобласть 1, на що вказує значення третього аргумента команди subplot. Тут буде виводитися перший графік.

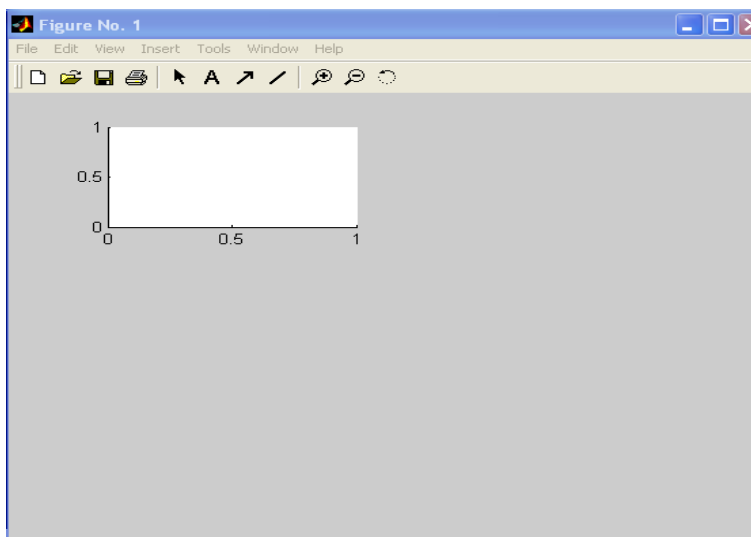


Рис. 88. Графічне вікно з активною підобластю № 1

Побудуємо графік отриманої матриці  $X$ , використовуючи функцію `surf`, яка будує каркасну поверхню і заливає кожен клітинку поверхні певним кольором. Використовуючи функцію `title`, яка дає можливість зробити підпис виводимо заголовок до графіку (рис. 89).

```
>> surf(X)
>> title('surf')
```

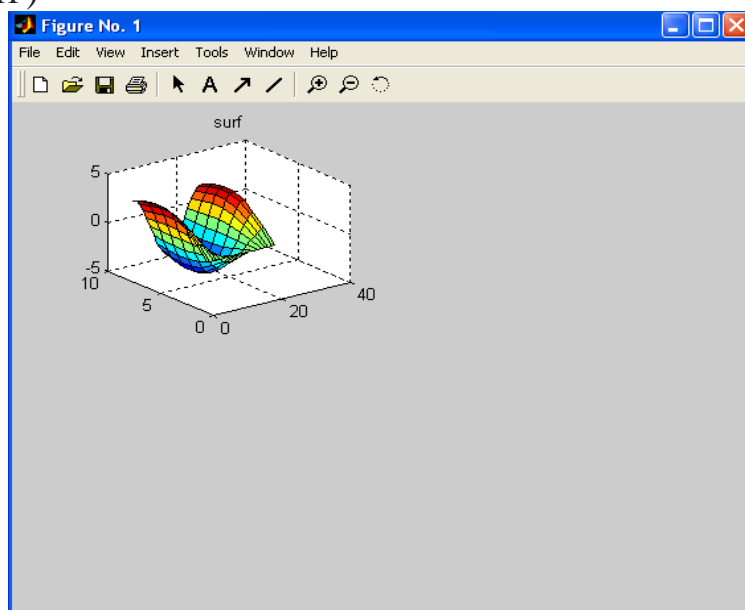


Рис. 89. Графічне вікно з графіком у підобласті № 1

Графік отриманої матриці  $Y$  побудуємо за допомогою функції `surf`. Виведення графіку здійснюється у підобласть № 2, про це свідчить значення третього аргумента команди `subplot`. Функцією `title` виводимо заголовок до графіку (рис. 90).

```
>> subplot(2, 2, 2)
>> surf(Y)
>> title('surf')
```

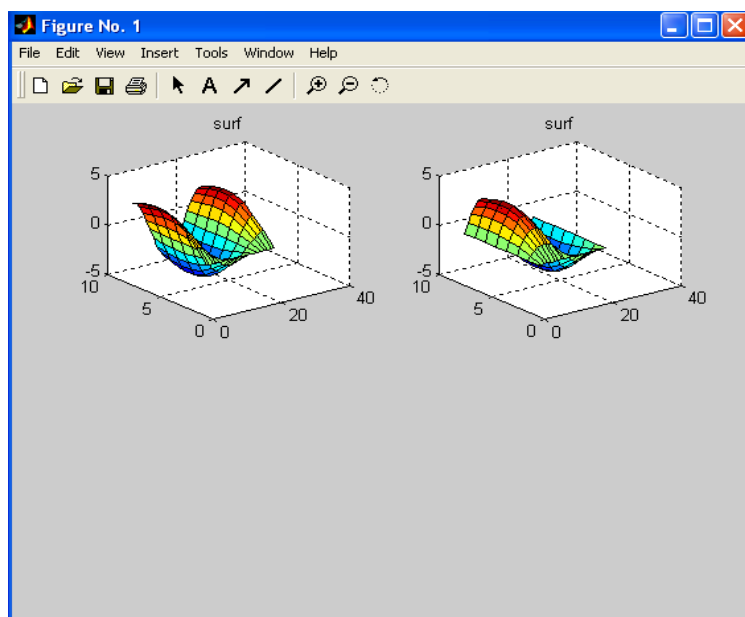


Рис. 90. Графічне вікно з виведеним графіком у підобласть № 2

Функцією `surf` побудуємо графік отриманої матриці  $Z$  у підобласті № 3.

За допомогою цієї функції здійснюється побудова залитої кольором каркасної поверхні та ліній рівня на площині  $xu$ . Побудований графік містить лінії рівня матриці (рис. 91).

```
>> subplot(2, 2, 3)
```

```
>> surf(Z)
```

```
>> title('surf')
```

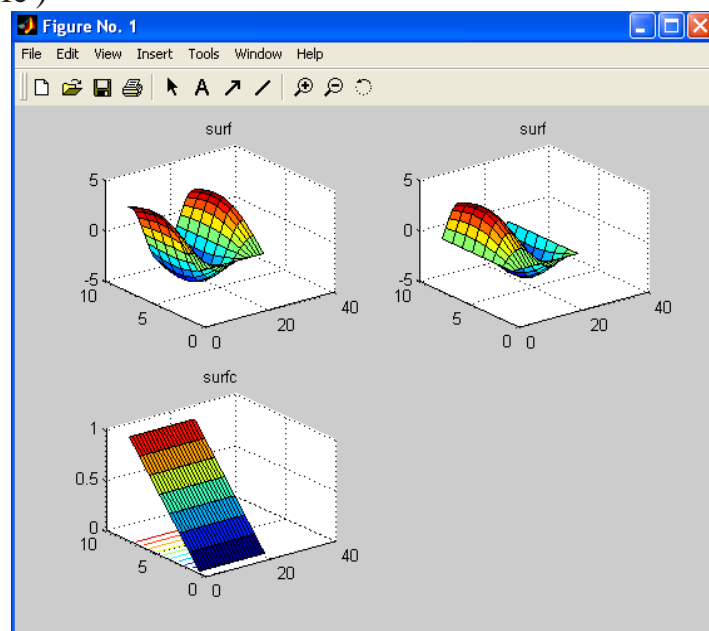


Рис. 91. Графічне вікно з виведеним графіком у підобласть № 3

У четверту підобласть виведемо графік отриманих матриць  $X$ ,  $Y$ ,  $Z$  використовуючи функцію `surfl`, за допомогою якої здійснюється побудова освітленої поверхні (рис. 92).

```
>> subplot(2, 2, 4)
```

```
>> surfl(X, Y, Z)
```

```
>> title('surfl')
```

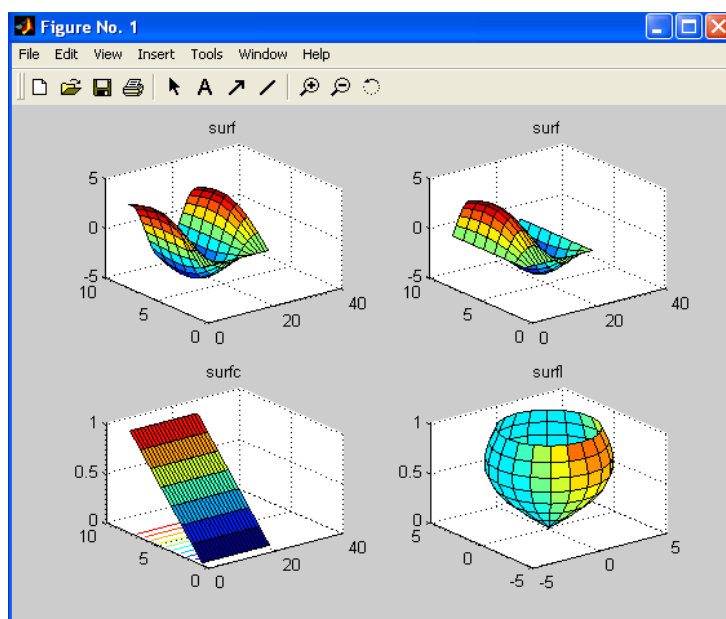


Рис. 92. Графічне вікно з виведеним графіком у підобласть № 4

Підготуємо графіки для друку на чорно-білому принтері. Встановлення палітри кольорів виконується за допомогою функції `colormap`. Для друку на монохромному принтері використовується палітра Gray (рис. 93).

`>> colormap (gray)`

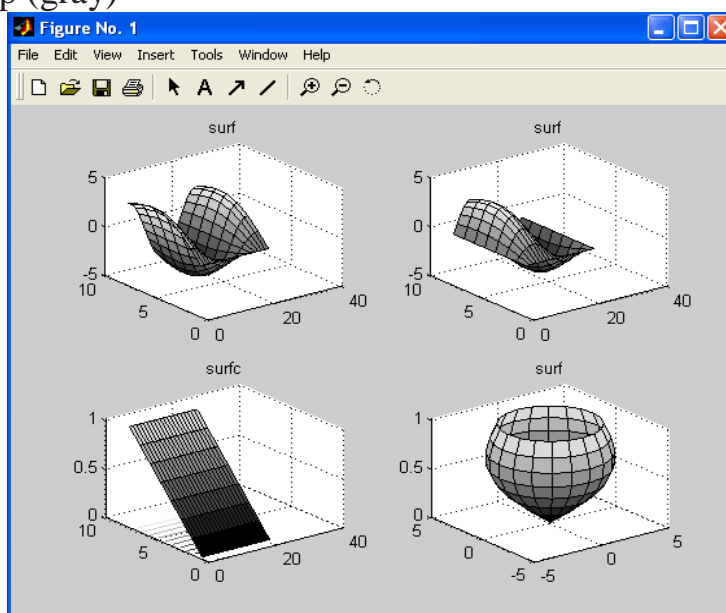


Рис. 93. Графічне вікно з графіками кольору палітри Gray

**Завдання 2.** Обчислити значення функції  $f = \cos 2t + 2\cos t$  і побудувати графіки, використовуючи функцію `mesh` і `surf` за умови, що значення  $t$  змінюється в інтервалі  $[0, 2\pi]$  з кроком 0.1:

```
>> t=0:0.1:2*pi;
>> [X,Y,Z]=cylinder(cos(t).^2+2.*cos(t));
>> subplot(2,2,1)
>> mesh(X,Y,Z)
>> title('mesh')
>> subplot(2,2,2)
>> surf(X,Y,Z)
>> title('surf')
```

В результаті виконання команд у першій та у другій підобласті вікна Figure 1 виведуться графіки функцією mesh і surf (рис. 94):

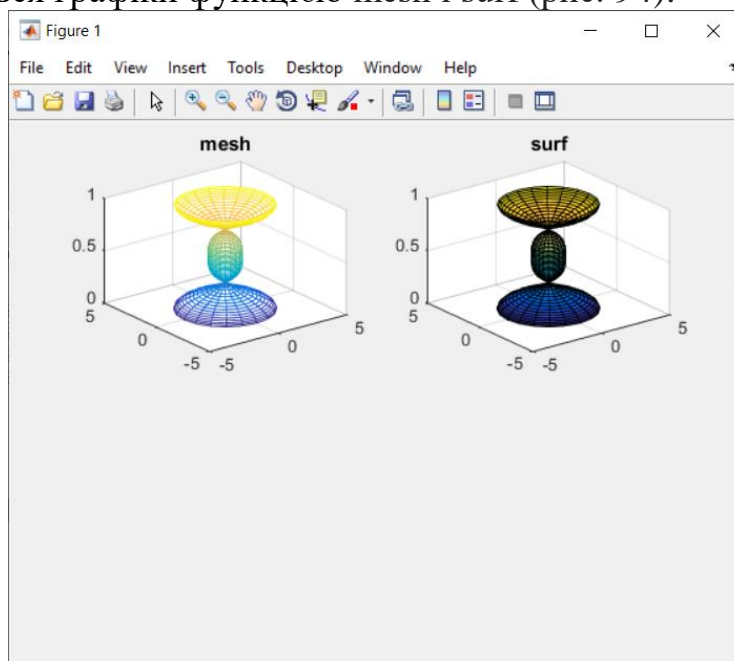


Рис. 94. Графічне вікно з графіками у підобласті № 1 і № 2

Побудувати графіки функції  $z=2*\sin(t)+t$  в області визначення  $t=0:\pi/7:5*\pi$ . Виведення графіків функції surf здійснювати в графічному вікні. Побудову графіків даної функції виконаємо у графічному вікні Figure 1, у підобластях № 3 та № 4 (рис. 95):

```
>> t=0:pi/7:5*pi;
>> [X,Y,Z]=cylinder(2.*sin(t)+t);
>> subplot(2,2,3)
>> mesh(X)
>> title('mesh')
>> subplot(2,2,4)
>> surf(X,Y,Z)
>> title('surf')
```

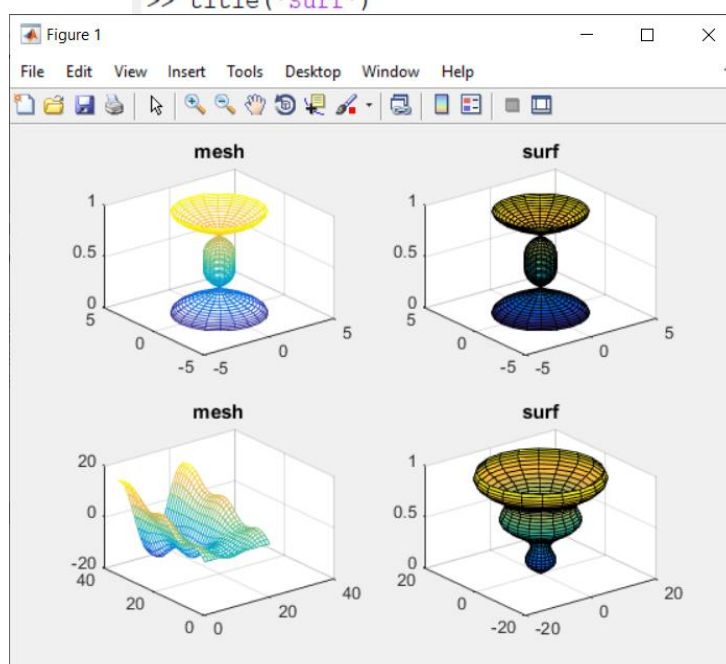


Рис. 95. Графічне вікно з графіками у підобласті №1, №2, №3, №4

**Завдання 3:** Обчислити значення функції  $z$  і побудувати графік використавши функцію *mesh* при умові, що значення  $x = 3$ , а значення  $y = 4$ .

Виведення графіків виконувати в одному графічному вікні на окремих підграфіках за допомогою функції *subplot*.

Зразок виконання завдання:

В робочому вікні Command Window задати значення вхідним змінним  $x$  та  $y$ , обчислити значення функції  $z$ . Далі виконати формування матриць  $X$ ,  $Y$ ,  $Z$ , для побудови сфери. За допомогою команди *figure* відкрити нове графічне вікно і в дужках задати номер вікна, наприклад, *figure(5)*. За допомогою функції *subplot* розбити вікно на дві підобласті по вертикалі і горизонталі, поточними робитимемо по черзі: підобласть 1, потім підобласть 2, далі 3, і потім 4. Лістинг виведення графіків в підобласті за допомогою функції *mesh*:

```
>> x=3;
>> y=4;
>> z=(2*(x+y)^3-(x+y)^2)/7;
>> [X,Y,Z]=sphere(z);
>> figure(5)
>> subplot(2,2,1)
>> mesh(X)
>> subplot(2,2,2)
>> mesh(Y)
>> subplot(2,2,3)
>> mesh(Z)
>> subplot(2,2,4)
>> mesh(X,Y,Z)
```



У вікно № 5 виведено графіки: матриці  $X$  у підграфіку № 1, матриці  $Y$  у підграфіку № 2, матриці  $Z$  у підграфіку № 3, матриць  $X$ ,  $Y$ ,  $Z$  у підграфіку № 4 (рис. 96):

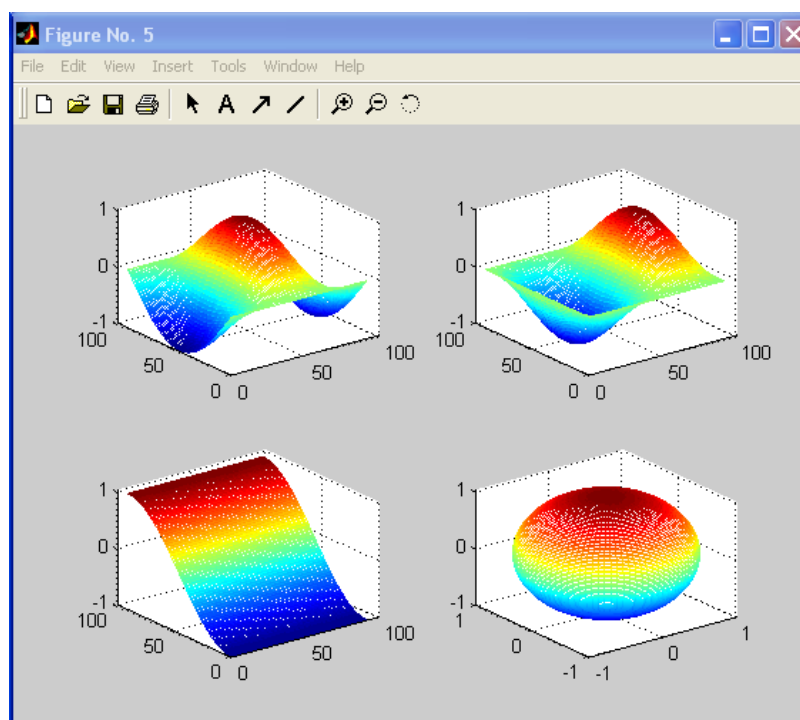


Рис. 96. Виведення чотирьох графіків в одному графічному вікні

**Індивідуальні завдання за темою: «Виведення декількох графіків в одному графічному вікні»**

**Варіанти індивідуальних завдань**

Обчислити значення функції  $z$  і побудувати графік використавши функцію *mesh* при умові, що значення  $x = 3$ , а значення  $y = 4$ .

1.  $z = (x - 2)^2 + 3y^2$
2.  $z = x^3 + 3xy^2 - 15x - 12y$
3.  $z = x^2 + 4xy - y^2 - 6x - 2y$
4.  $z = (2x + 3y) - 10$
5.  $z = (x + y)^3$
6.  $z = (4x + y^2) \cdot x^2$
7.  $z = (x^2 + 6y) + 5 \cdot (y^2 + 2x)$
8.  $z = 5 + xy + 2x^3$
9.  $z = 2xy + 3xy + x^2 + y^2$
10.  $z = (6xy + x^2y^2) + 55$
11.  $z = (2x + 2y)^2$
12.  $z = x^2 + y^2$
13.  $z = x^2 + y$
14.  $z = |xy| + 2(x + y) + (xy)^2$
15.  $z = 3x^2 + 5y$
16.  $z = 3 \cdot |x| + y$
17.  $z = 2x^3 + 4xy$
18.  $z = (x^3 + y^3) + xy$
19.  $z = 5x + |y|$
20.  $z = 7x + 8y + (xy)^2 + 72$
21.  $z = 3x^3 + 2y^3 + 5xy$
22.  $z = 5 \cdot |x + y| + 4y^2 + 3$
23.  $z = 2xy^2 + 2x^2y + xy$
24.  $z = 12x^2 + 10y^2$
25.  $z = (x^3 + y^3) + (xy)^2$



## ВИСНОВКИ

MATLAB – це високорівнева мова технічних розрахунків, інтерактивне середовище розробки алгоритмів та сучасний інструмент аналізу даних. MATLAB у порівнянні з традиційними мовами програмування (C/C++, Java, Pascal, FORTRAN) дозволяє на порядок скоротити час вирішення типових завдань та значно спрощує розробку нових алгоритмів.

MATLAB є основою всього сімейства продуктів MathWorks і є головним інструментом для вирішення широкого спектру наукових та прикладних завдань, в таких галузях як: моделювання об'єктів та розробка систем управління, проектування комунікаційних систем, обробка сигналів та зображень, вимірювання сигналів та тестування, фінансове моделювання, обчислювальна біологія та інші.

Вже перші орієнтовані на Microsoft Windows версії системи (MATLAB 4.x) мали потужні засоби. У галузі математичних обчислень: матричні, векторні, логічні оператори; елементарні та спеціальні функції; поліноміальна арифметика; багатовимірні масиви; масиви записів; масиви осередків.

У сфері реалізації чисельних методів: диференціальне рівняння; обчислення одновимірних та двовимірних квадратур; пошук коренів нелінійних рівнянь алгебри; оптимізація функцій кількох змінних; одновимірна та багатовимірна інтерполяція.

У галузі програмування: понад 500 вбудованих математичних функцій; введення/виведення двійкових та текстових файлів; застосування програм, написаних на Сі та ФОРТРАН; автоматичне перекодування процедур MATLAB у тексти програм мовами Сі та С++; типові керуючі структури.

В галузі візуалізації та графіки: можливість створення двовимірних та тривимірних графіків; здійснення візуального аналізу даних.

Ці засоби поєднувалися з відкритою архітектурою систем, що дозволяє змінювати існуючі функції і додавати свої власні. Програма Simulink, що входить до складу MATLAB, дає можливість імітувати реальні системи та пристрої, задаючи їх моделями, складеними з функціональних блоків. Simulink має велику бібліотеку блоків, що розширюється користувачами, і прості засоби завдання та зміни їх параметрів.

Нові версії системи MATLAB мають не тільки можливості попередніх версій, але й характеризується цілою низкою нових та важливих можливостей.

## СПИСОК ЛІТЕРАТУРИ

1. Дьяконов В. П., Абраменкова И. В., Круглов В. В. MATLAB 5 с пакетами расширений. – М.: Нолидж, 2001.
2. Дьяконов В.П. MATLAB: Учебный курс. – СПб.: ПИТЕР, 2001.
3. Дьяконов В.П. MATLAB 6: Учебный курс. – СПб.: ПИТЕР, 2001.
4. Дьяконов В. П., Круглов В. В. Математические пакеты расширения MATLAB: Специальный справочник. – СПб.: ПИТЕР, 2001.
5. Дьяконов В. П., Круглов В. В. MATLAB. Анализ, идентификация и моделирование систем: Специальный справочник. – СПб.: ПИТЕР, 2002.
6. Дьяконов В. П. MATLAB 6/6.1/6.5 + Simulink 4/5 в. Обработка сигналов и изображения: Полное руководство пользователя. – М.: Солон\_Р, 2004.
7. Дьяконов В. П. MATLAB 6.5 SP1/7 + Simulink 5/6 в. Работа с изображениями и видеопотоками. – М.: Солон\_Р, 2005.
8. Потемкин В. Г. Система MATLAB: Справочное пособие. – М.: Диалог\_МИФИ, 1997.
9. Потемкин В. Г. Инструментальные средства MATLAB 5.x. – М.: ДИАЛОГ\_МИФИ, 2000.
10. Потемкин В. Г. Вычисления в среде MATLAB. – М.: ДИАЛОГ\_МИФИ, 2004.
11. Лазарев Ю. Ф. MATLAB 5.x. (серия «Библиотека студента»). – Киев: Издательская группа ВНУ, 2000.
12. Ануфриев. И. MATLAB 5.3/6.x: Самоучитель. – СПб.: БХВ\_Петербург, 2002.
13. Ануфриев И. Е., Смирнов А. Б., Смирнова Е. Н. MATLAB 7. – СПб.: БХВ\_Петербург, 2005.
14. Гультяев. А. Визуальное моделирование в среде MATLAB: Учебный курс. – СПб.: Питер, 2000.
15. Лазарев Ю. Моделирование процессов и систем в MATLAB: Учебный курс. – СПб.: Питер; Киев: Изд. группа ВНУ, 2005.
16. Медведев В. С., Потемкин В. Г. Control System Toolbox. MATLAB 5 для студентов. – М.: ДИАЛОГ\_МИФИ, 2004.
17. Мартынов Н. Н., Иванов А. П. MATLAB 5.x. Вычисления, визуализация, программирование. – М.: КУДИЦ\_ОБРАЗ, 2000.