**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки та комп’ютерних систем**

Лабораторна робота №1-2

Тема: Знайомство з SCILAB

Роботу виконав

студент 4 курсу

мережевий адміністратор

Цибульський Роман Олександрович

Київ 2023

1. Завантажте та встановіть SCILAB (завантажити його можна з офіційного сайту - https://www.scilab.org/ ).



1. Запустіть SCILAB та ознайомтесь з графічним інтерфейсом. По замовчуванню після запуску він має такі вікна:

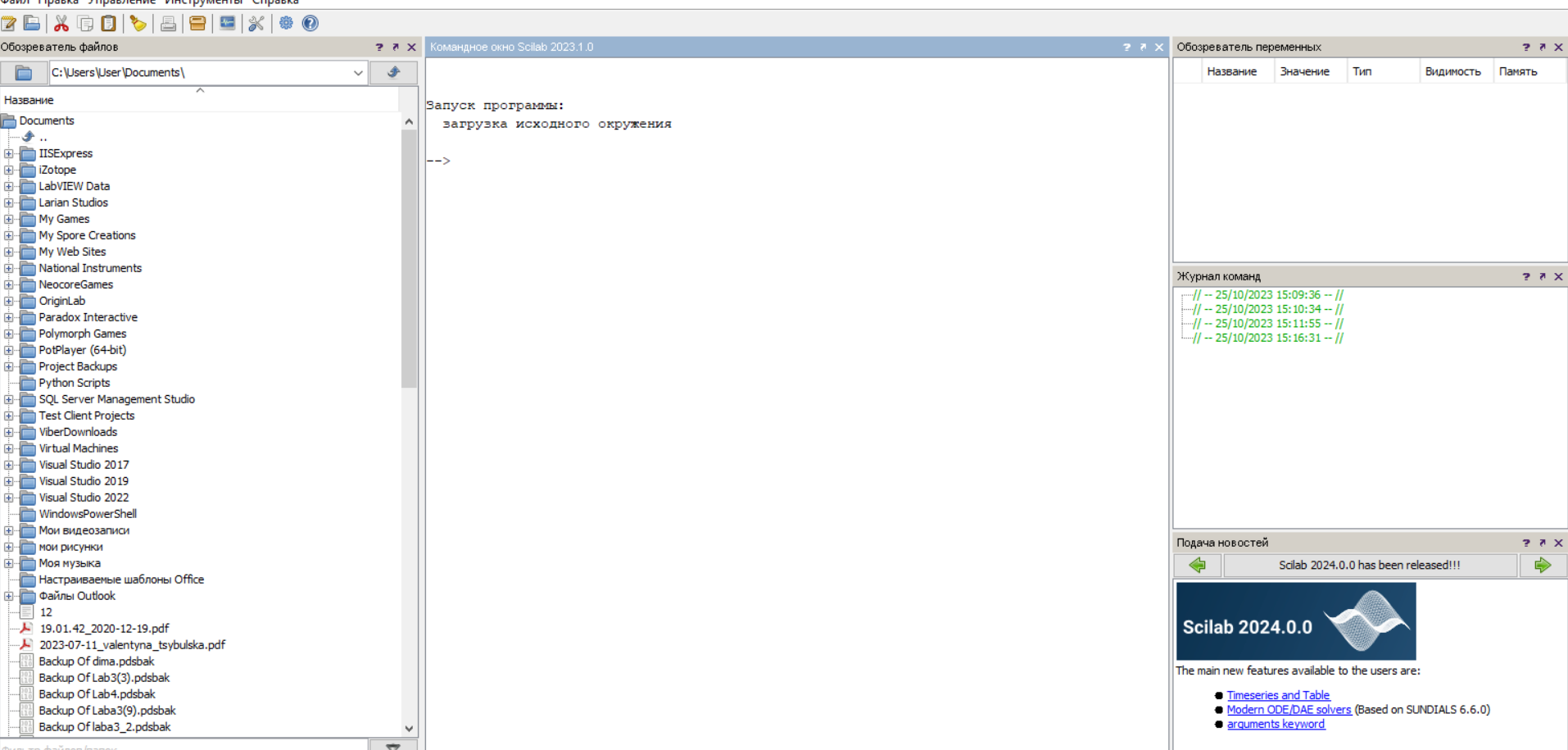
• File browser

• SCILAB console

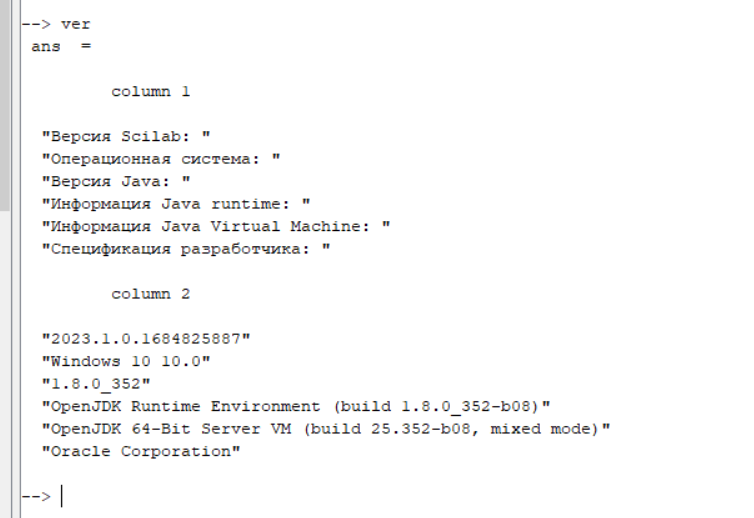
• Variable browser

• Command history

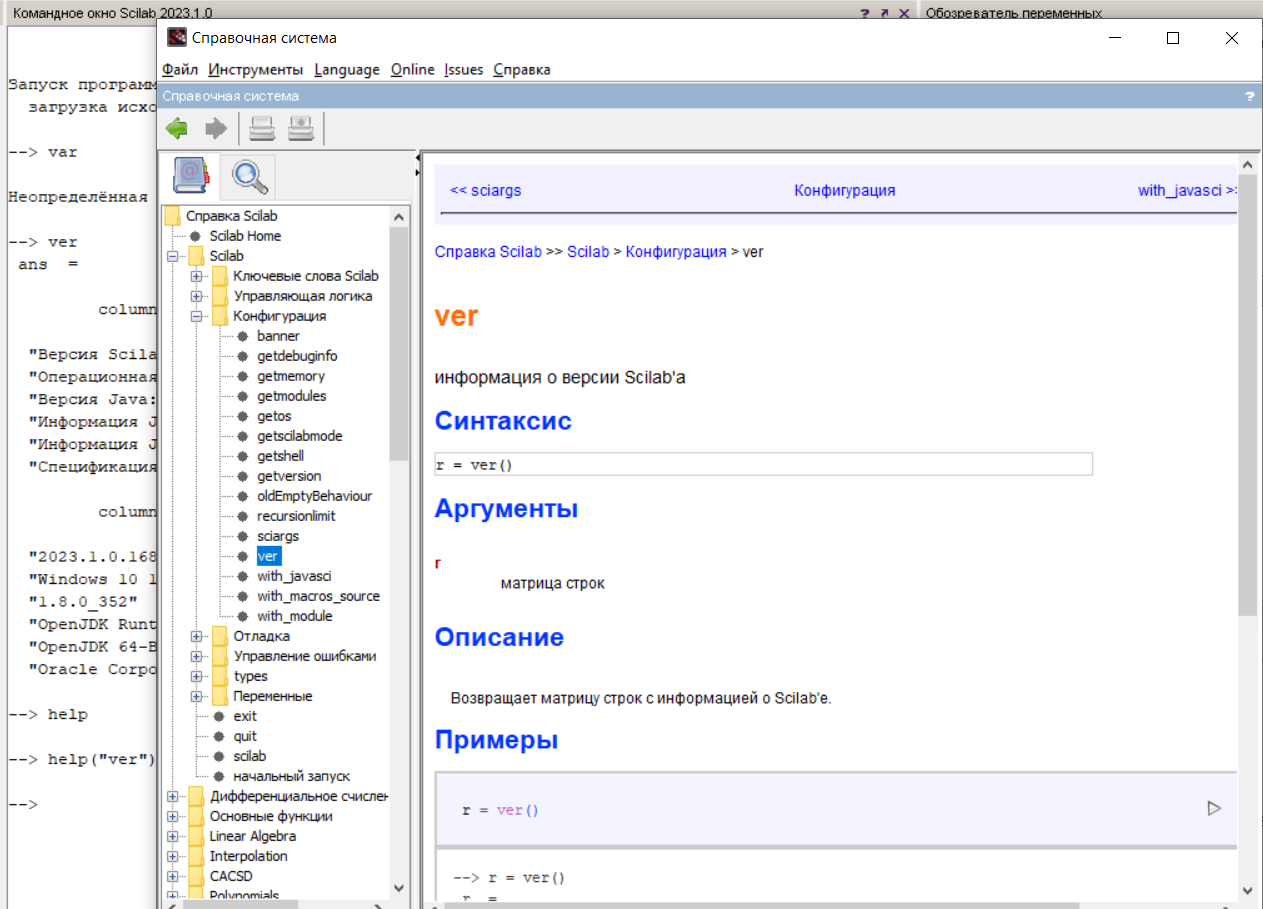
• News feed



1. Для перевірки працездатності введіть в Scilab-консолікоманду ver. Ви повинні отримати вивід, який містить інформацію про версію Scilab, версію Java та вашої операційної системи:



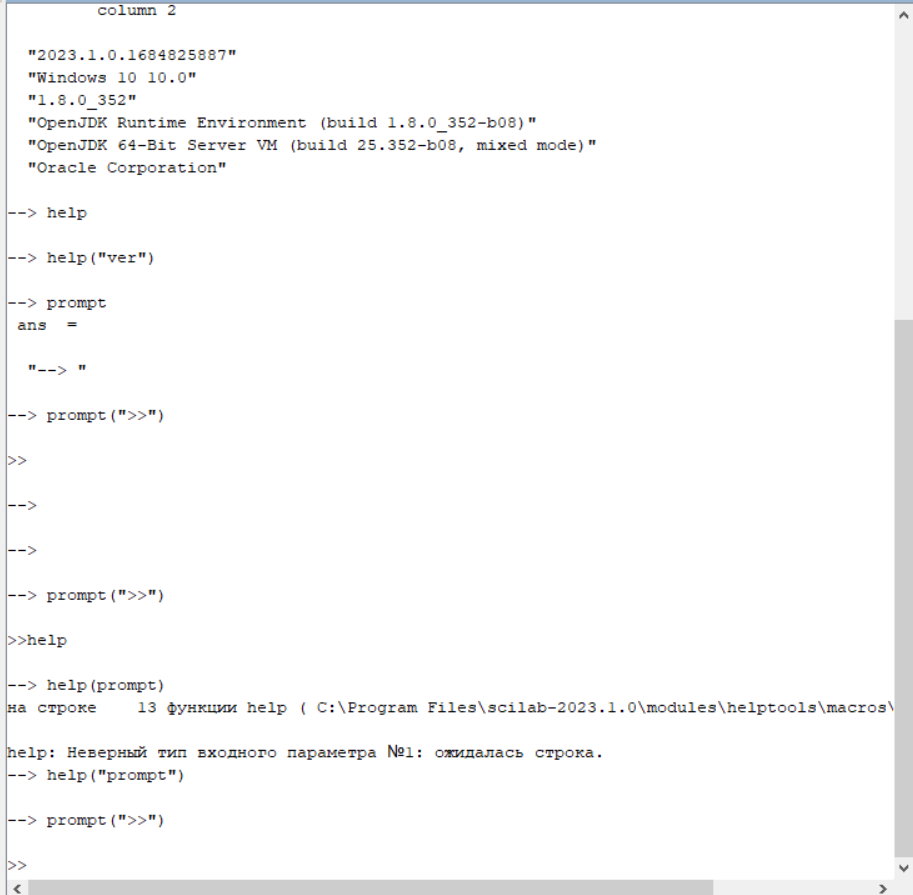
1. Для отримання довідки про будь-яку будь-яку вбудовану функцію в Scilab існує команда help.

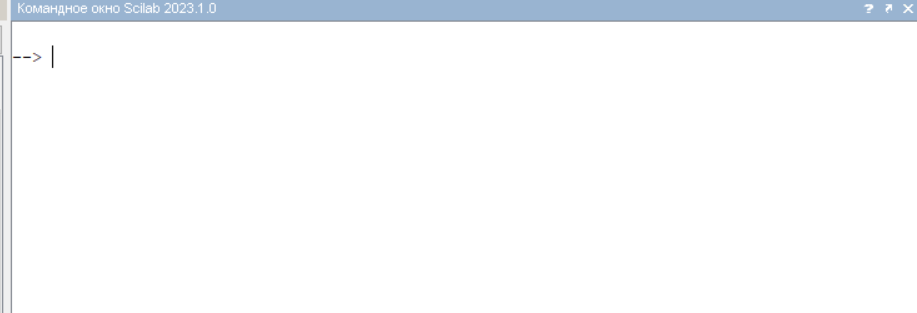


1. По замовчуванню, ”-->” - це стандартний рядок запрошення, його можна змінити за допомогою команди prompt:

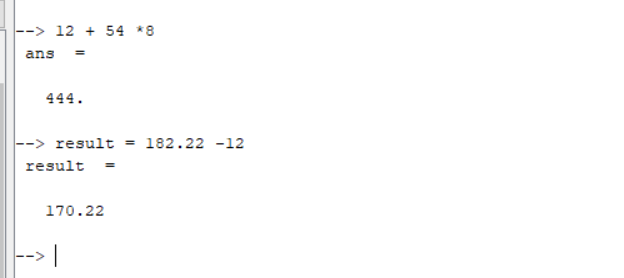


Для очищення Scilab-консолі використовують функцію clc

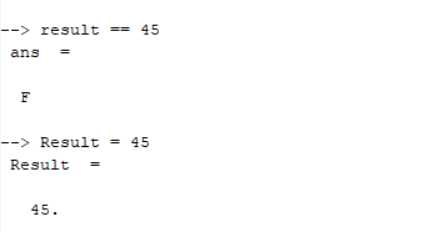




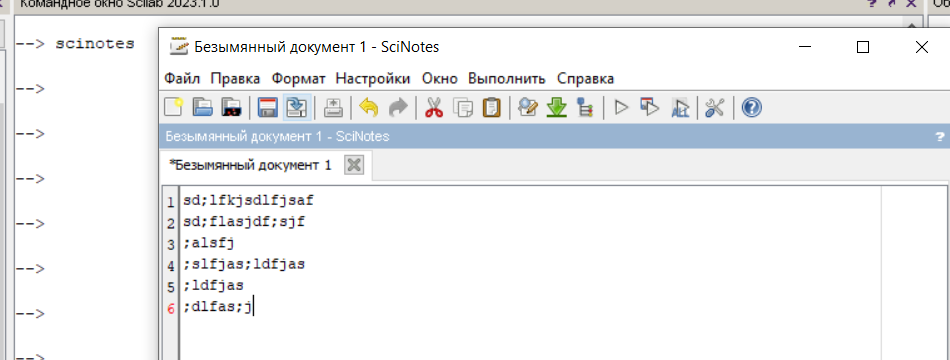
1. Значення всіх змінних ви можете переглянути у вікні Variable Browserа а.



Оператор присвоєння = (можна робити декілька присвоєнь через кому), оператор порівняння ==,



1. У Scilab також є вбудований текстовий редактор, який можна викликати з меню Application -> SciNotes, або ввівши в Scilab-консолі scinotes.

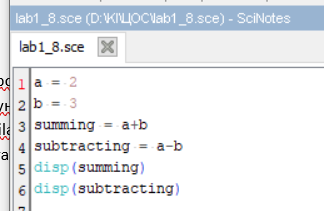


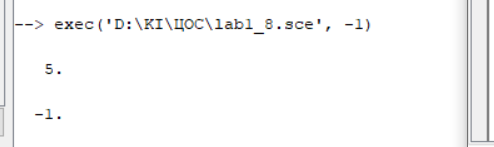
1. Скриптові файли для Scilabа у мають розширення .sci або .sce

• .sci файли містять Scilab-функції та функції, визначені користувачем. Виклик таких файлів просто завантажує їх у середовище Scilab, але не виконує їх.

• .sce файли містять як функції Scilab, так і виконувані команди. Коли ці файли викликаються, вони виконуються.

Спробуйте ввести наступні команди у SciNotes та зберегти їх у файл з розширенням .sce:

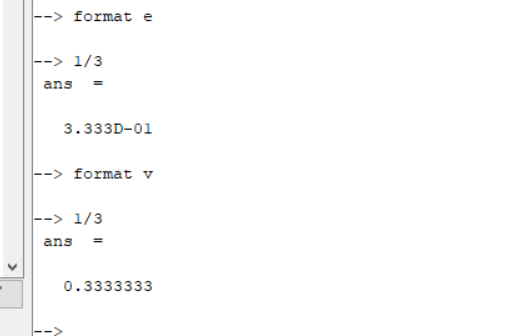




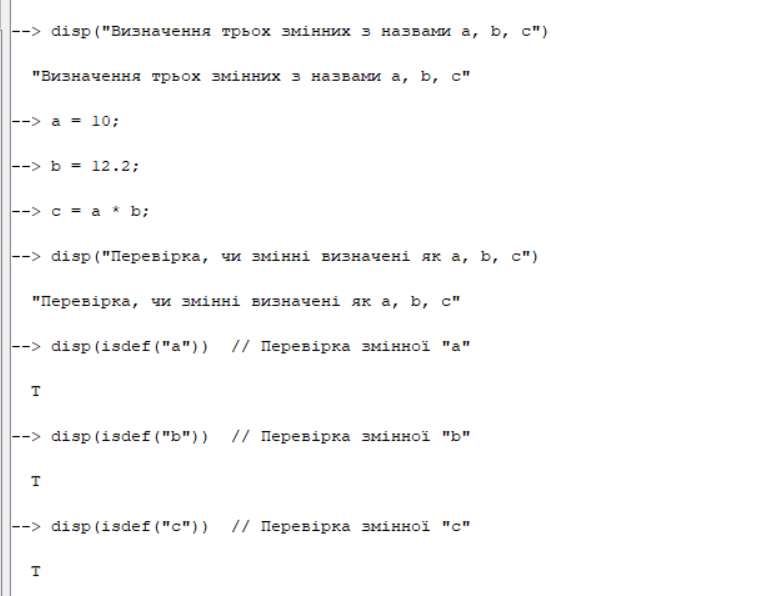
1. В Scilab є два режими виводу:

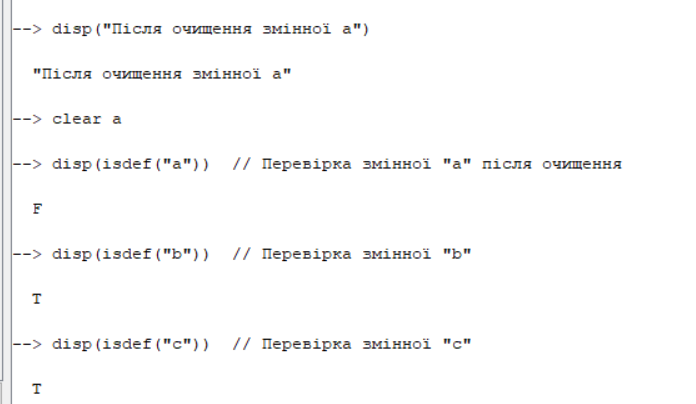
• режим “v” ( variable adaptative formatting), де Scilab самостійно обирає в якій нотації виводити числа

• режим “e” ( exponential /engineering), в якому числа будуть записані у вигляді 3.18711E-11



1. Після визначення змінної, її можна винищити за допомогою функції clear. Також є функція isdef(), яка перевіряє чи визначена така змінна. Ознайомтесь з наступним кодом, виконайте його та поясніть отримані результати:





1. Наукові обчислення вимагають певних попередньо визначених констант. Їм передує знак %. Ознайомтесь з цим списком констант:

• %pi == 3.1415927 (value of pi)

• %eps == 2.220D−16 (value of epsilon 2 −52)

• %inf == Inf (value of infinity ∞)

• %e == 2.7182818 (value of Eulerа s number)

• %i == i (value of imaginary number i)

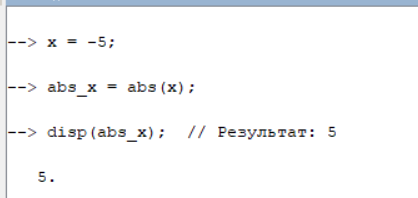
• %t == T (value of boolean True)

• %f == F (value of boolean False)

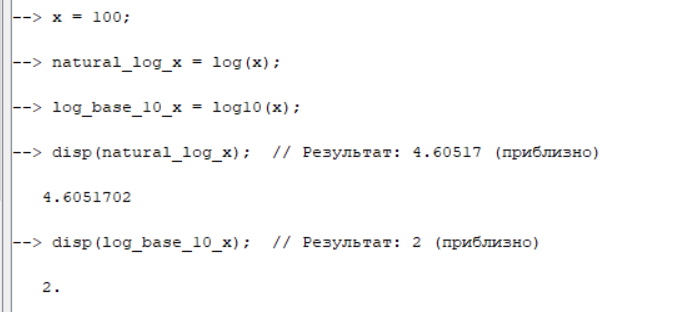
• %nan == Nan (value of Not−a−number variable)

1. В Scilab ви також можете використовувати звичні математичні функції типу:

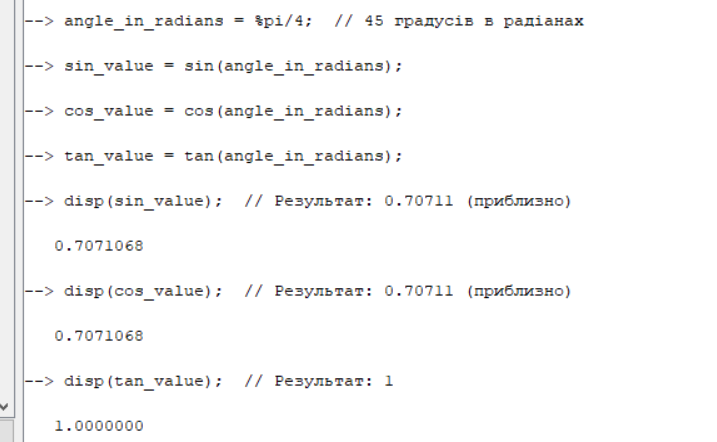
• abs() - модуль



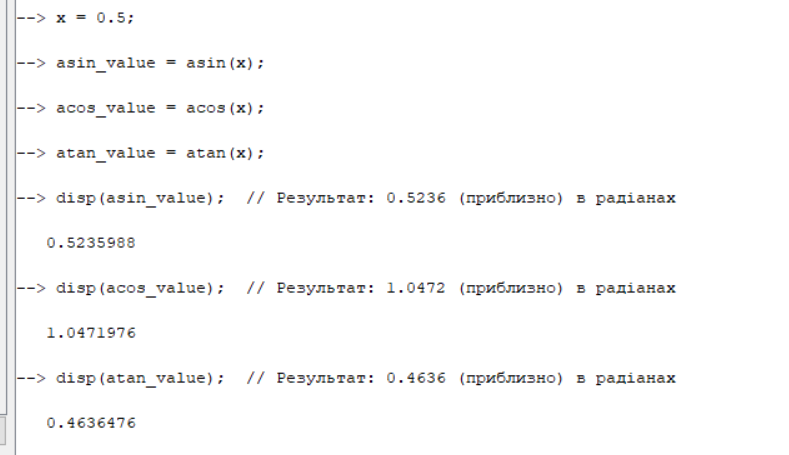
• log(), log10() - функції натурально логарифму та логарифму з основою 10



• sin(), cos(), tan() - тригонометричні функції (аргументи в радіанах)



• asin(), acos(), atan() - зворотні тригонометричні функції



1. Комплексні числа та функції для роботи з ними:

• complex() - створення комплексного числа з вказаної цілої та уявної частини

• real() - взяття дійсної частини з комплексного числа

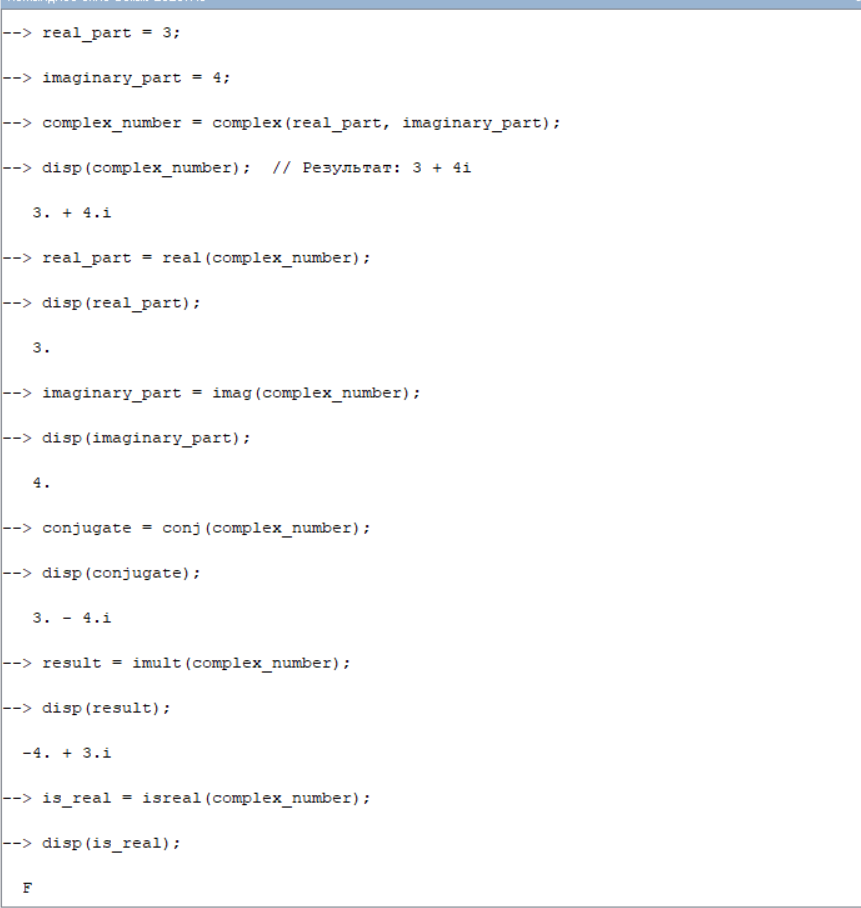
• imag() - взяття уявної частини з комплексного числа

• conj() - отримання кон'югату (сполученого числа)

• imult() - множення комплексного числа на i

• isreal() - перевірка на те, чи має змінна уявну частину

Ці функції також працюють і з матрицями. Всі ці функції досить прості. Продемонструйте навички їх використання.



1. Масиви використовуються для визначення векторів, таблиць та матриць:

• 1-D одновимірний масив виступає як вектор або список

• 2-D масив може використовуватися як таблиця або матриця

• 3-D і більше-D масиви можуть представляти багатовимірні матриці

• size() - визначити розмір матриці

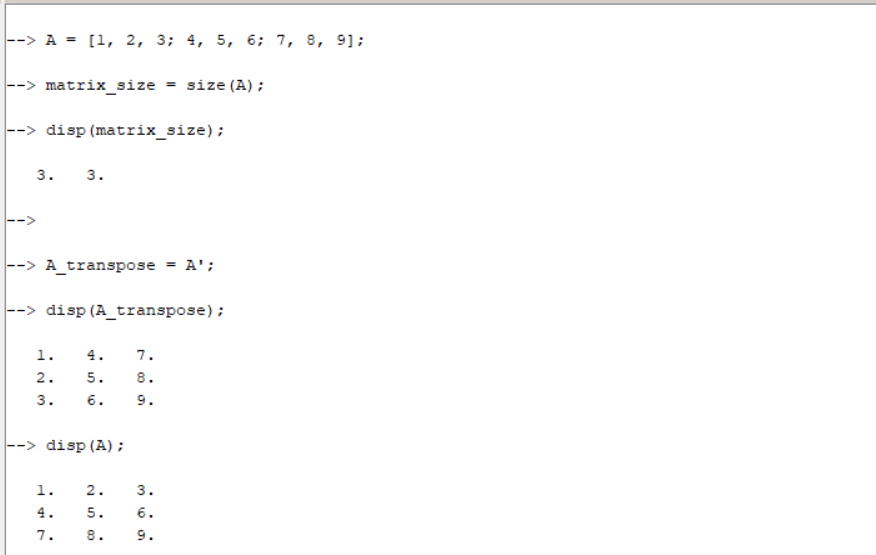
• ’ - транспонування (тобто до a можна просто дописати ’ і отримати транспоновану матрицю a’)

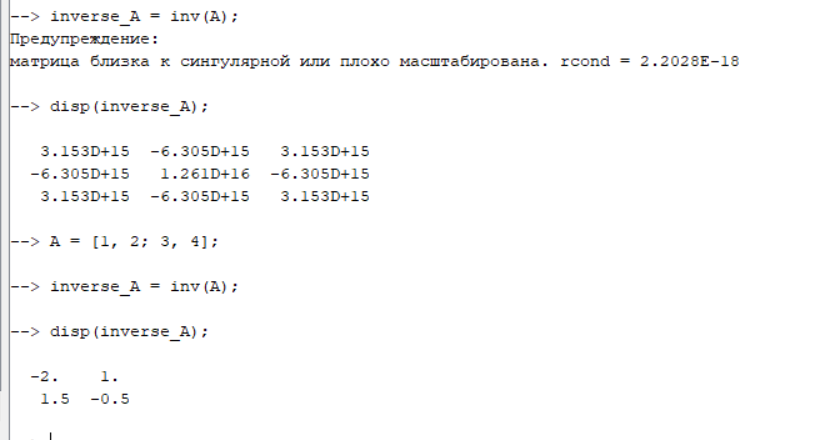
• inv() - інверсія

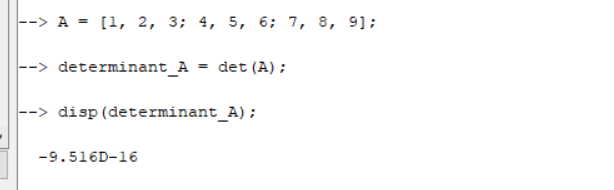
• det() - визначити детермінант

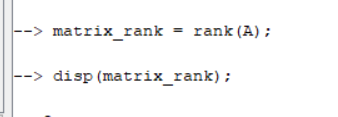
• rank() - ранг матриці

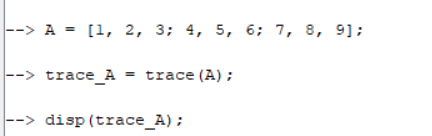
• trace() - сума діагональних елементів матриці



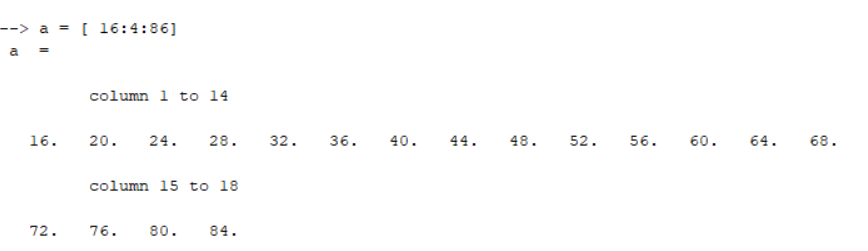




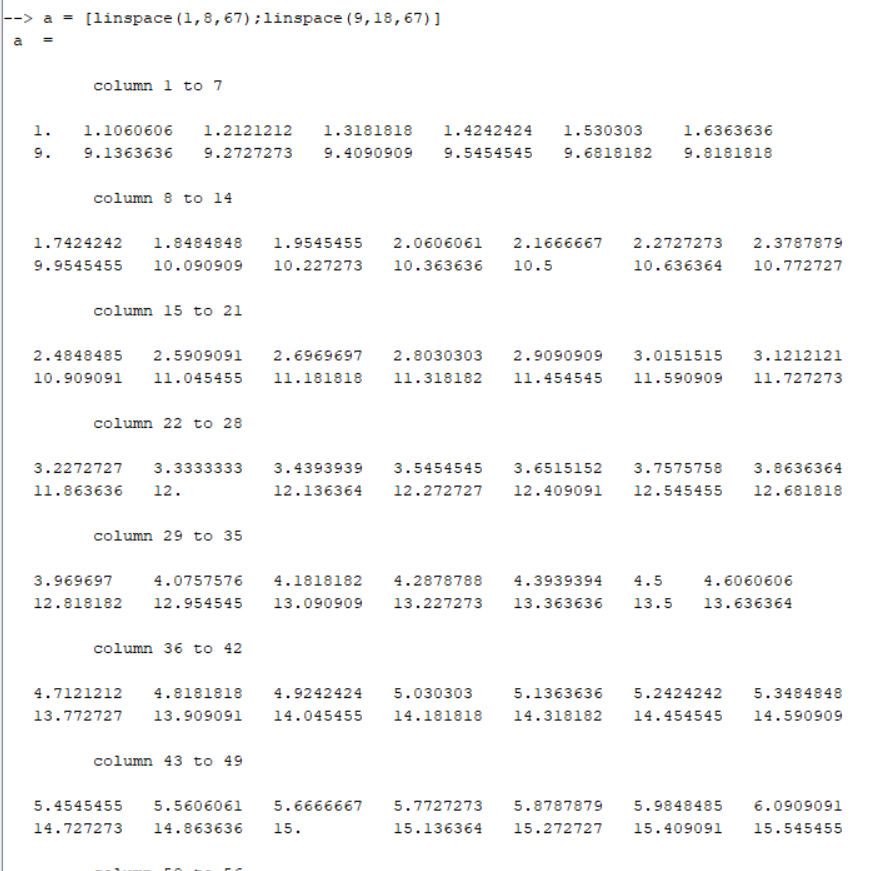




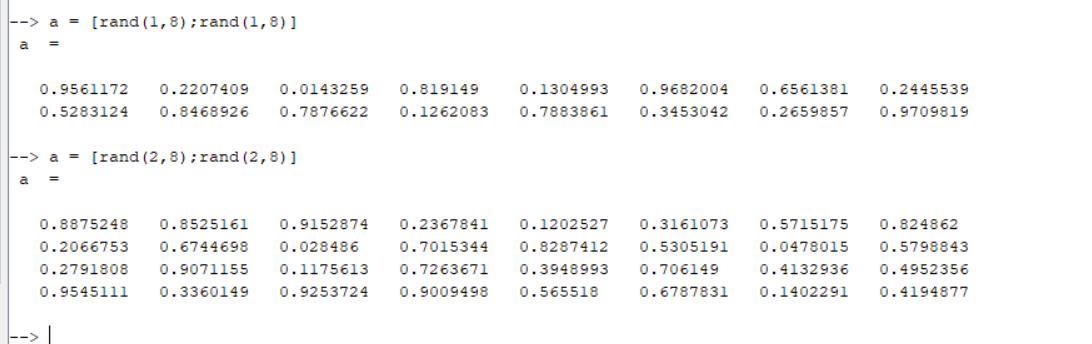
1. Для того, щоб автоматично згенерувати масив (вектор або матрицю), використовують нотацію start:step:stop



функція linspace(start,stop,n),

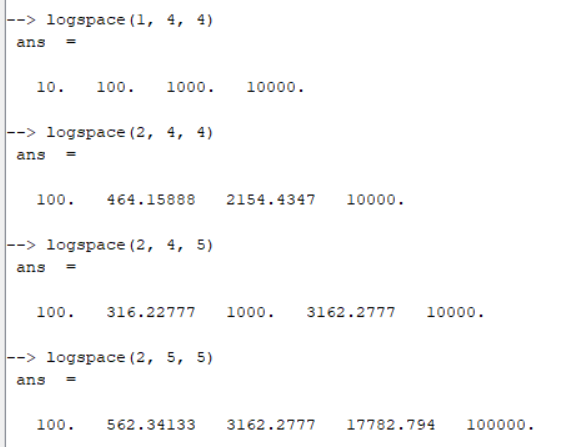


функція rand()

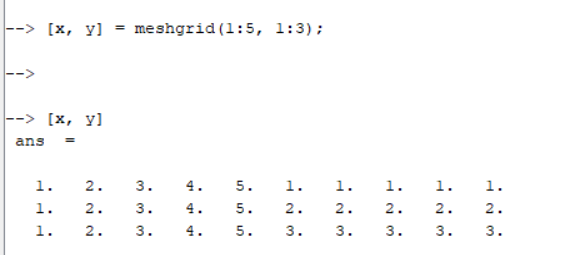


logspace() генерує вектор значень, розміщених рівномірно в логарифмічному масштабі.

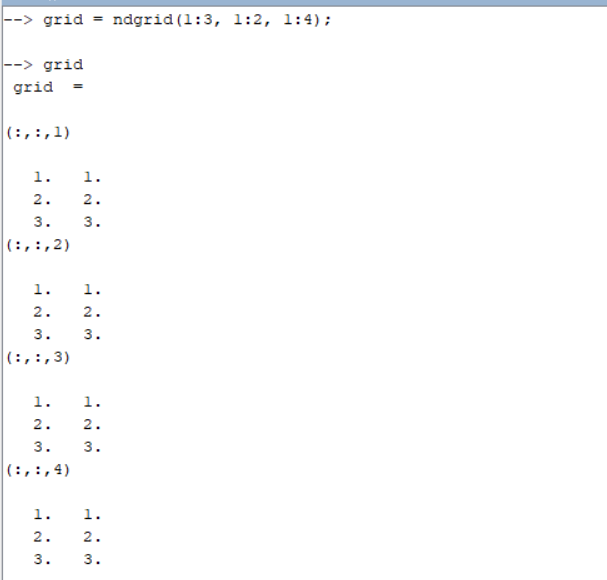
10^(start + (i-1) \* (end - start) / (n-1))

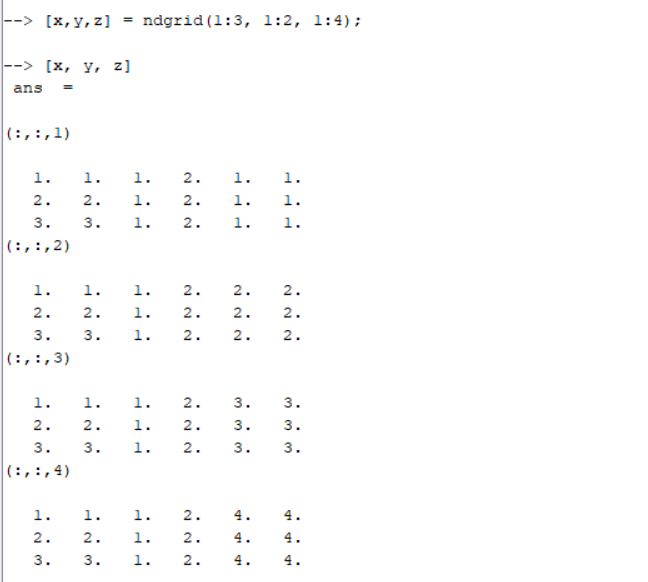


meshgrid() створює дві матриці X і Y, які представляють сітку координат у 2D-просторі



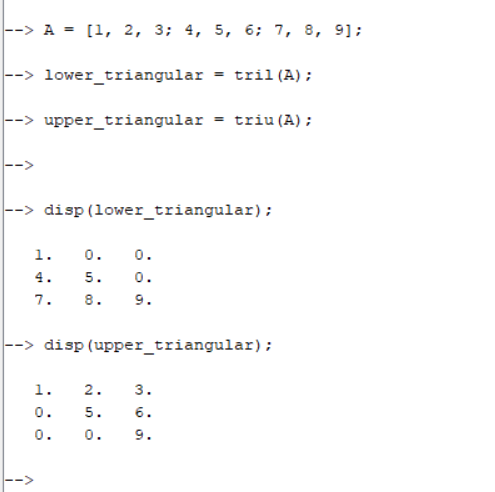
ndgrid() створює об'єкт, який може бути використаний для індексації об'єкту більш високого порядку в багатовимірному просторі.



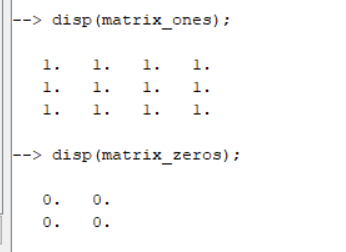


1. Для генерації спеціальних матриць використовують ще такі функції, як

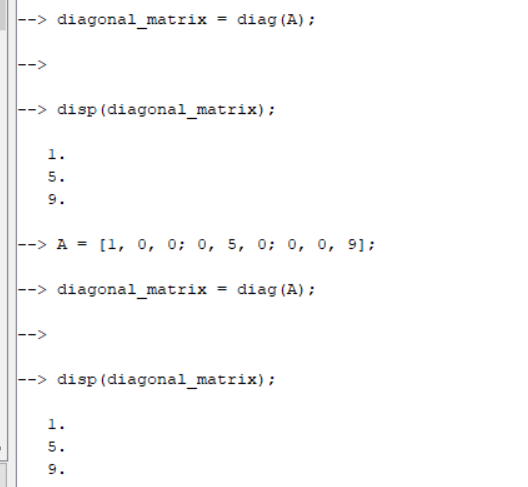
• tril(), triu() - генерація нижньої та верхньої трикутної матриці



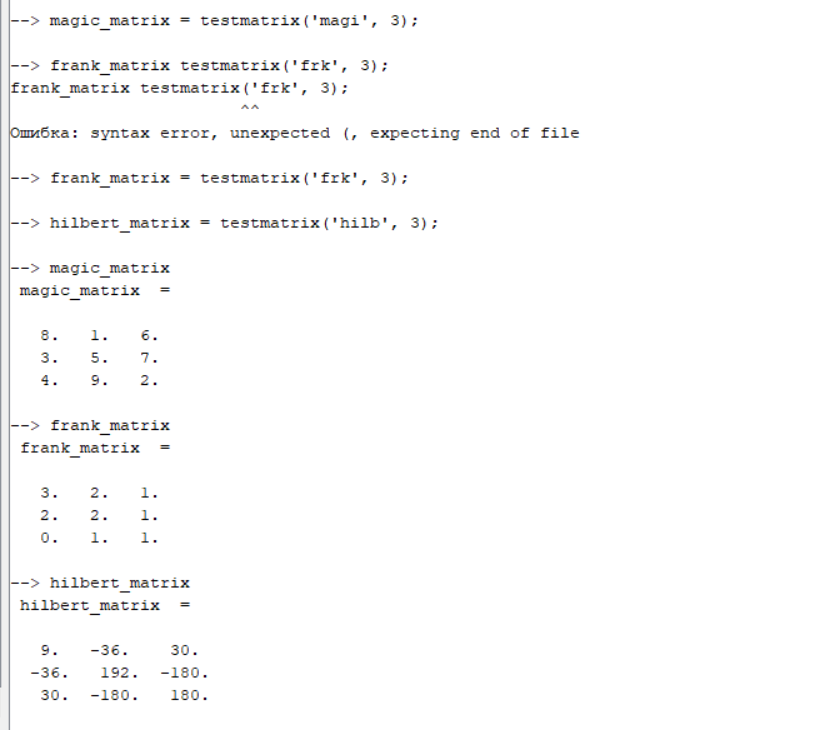
• ones(), zeros() - створення одиничної та нульової матриці



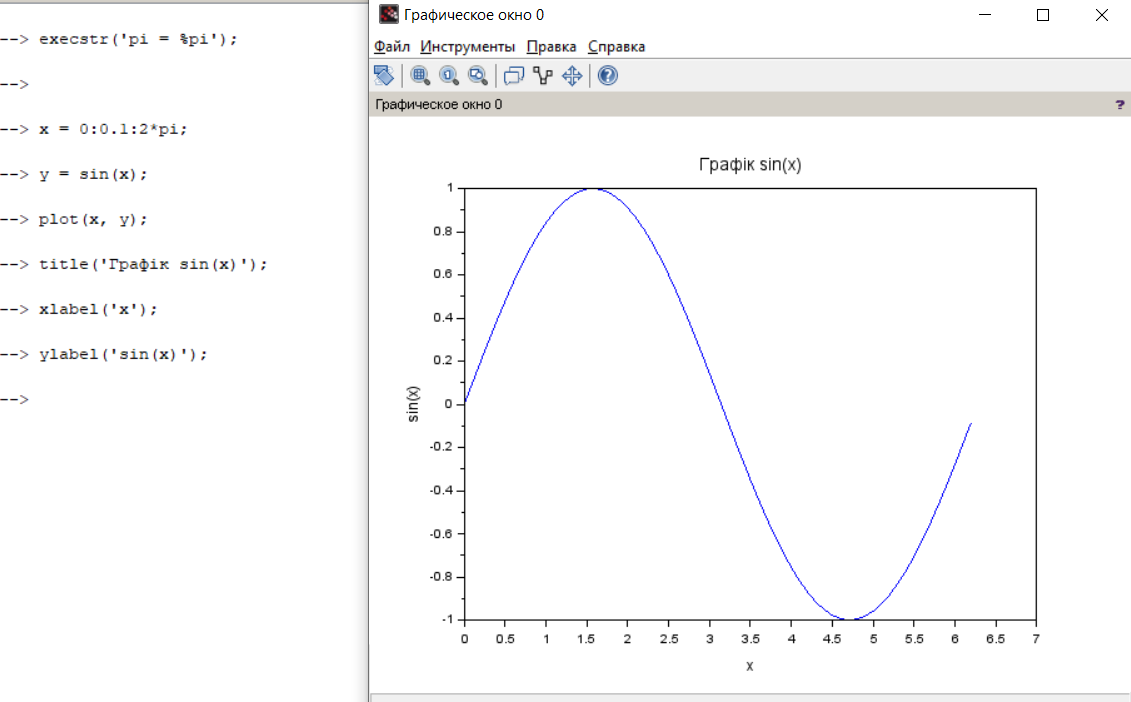
• diag() - отримання діагональної матриці

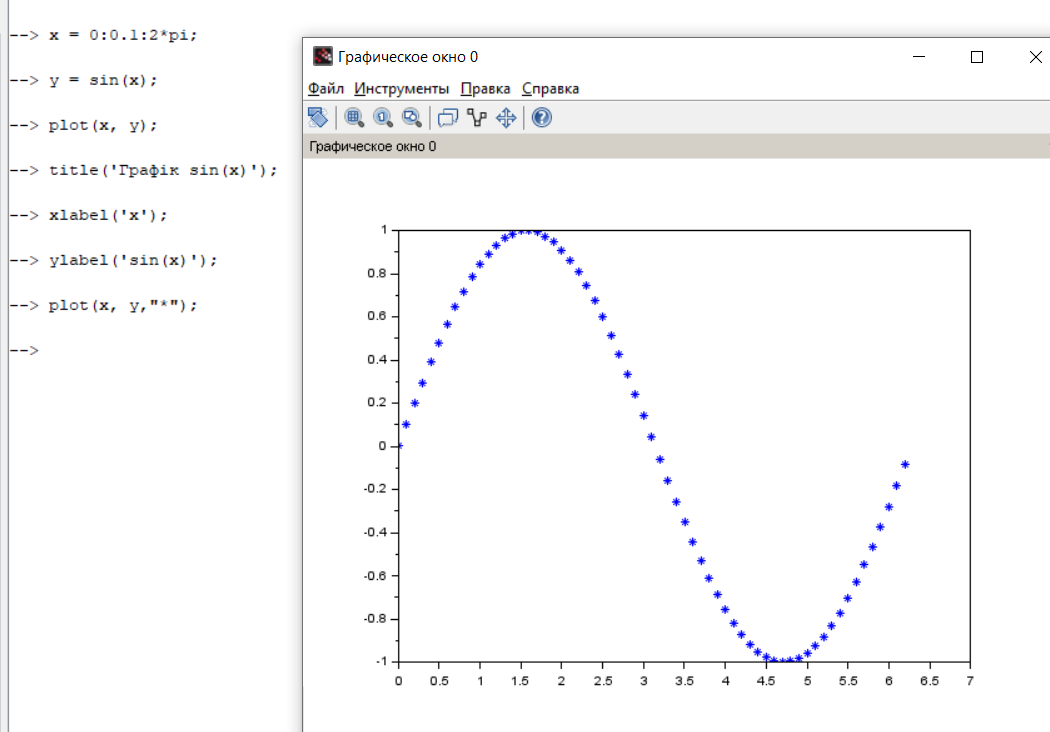


• testmatrix() - генерація спеціальних матриць (магічного квадрату, матриці Франка, матриці Гільберта і тд)

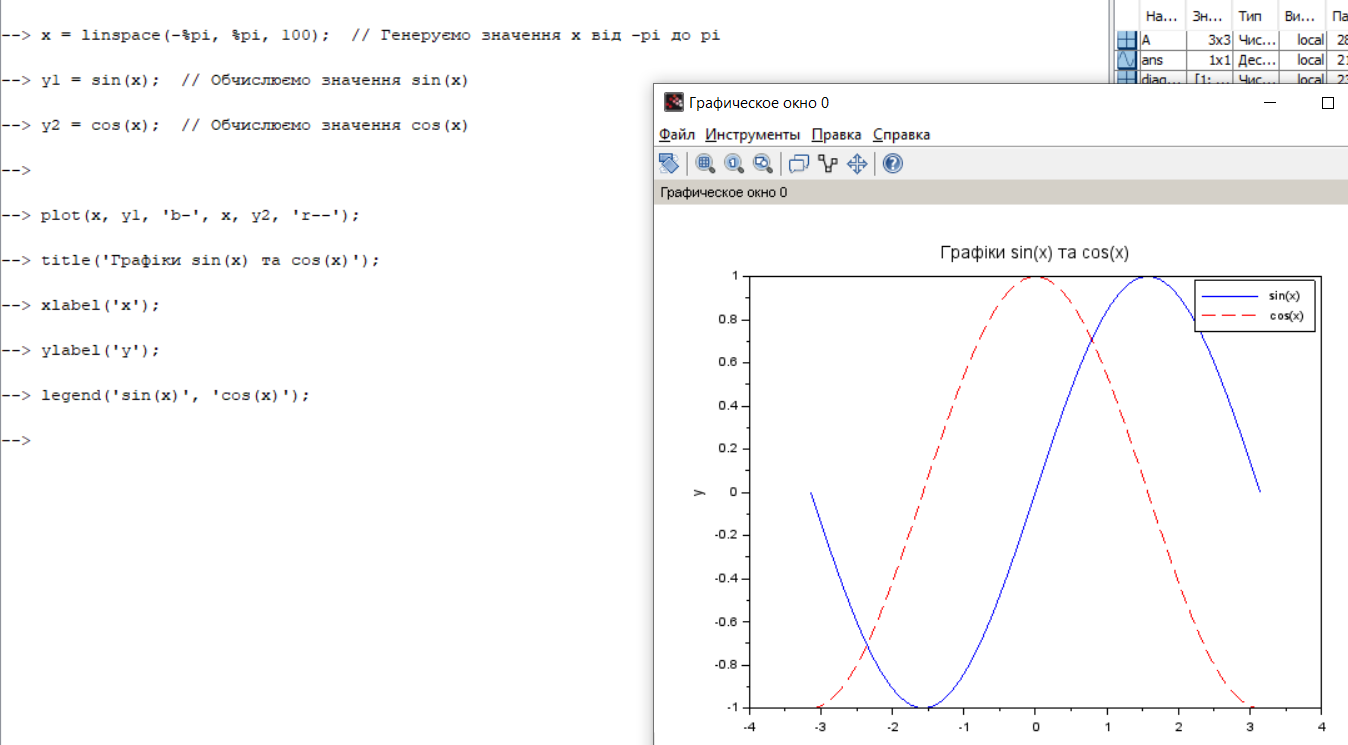


1. Для роботи зі звичайними графіками використовують функцію plot(). Для того щоб уникнути виводу в консоль після якоїсь команди, використовуйте

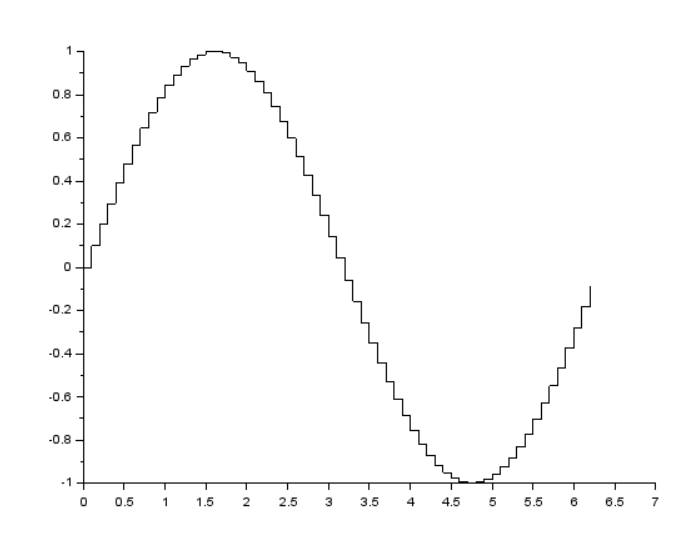




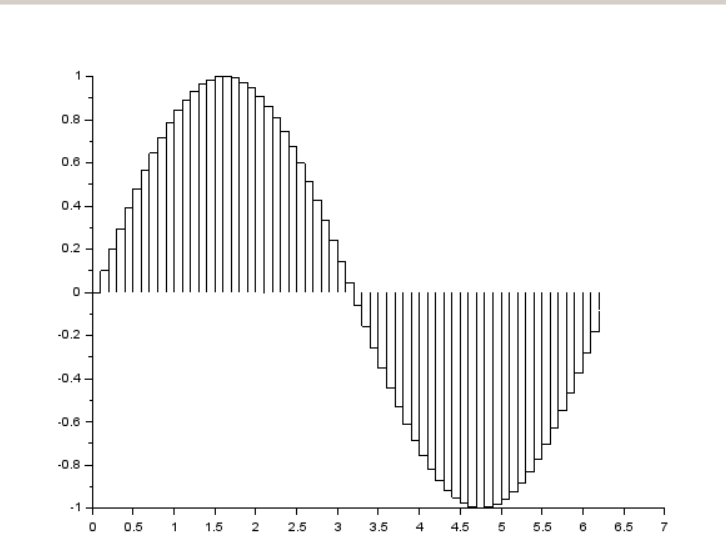
Спробуйте самостійно зобразити на одному графіки функції y1=sin(x), y2=cos(x), де x визначений на проміжку від -pi до pi.

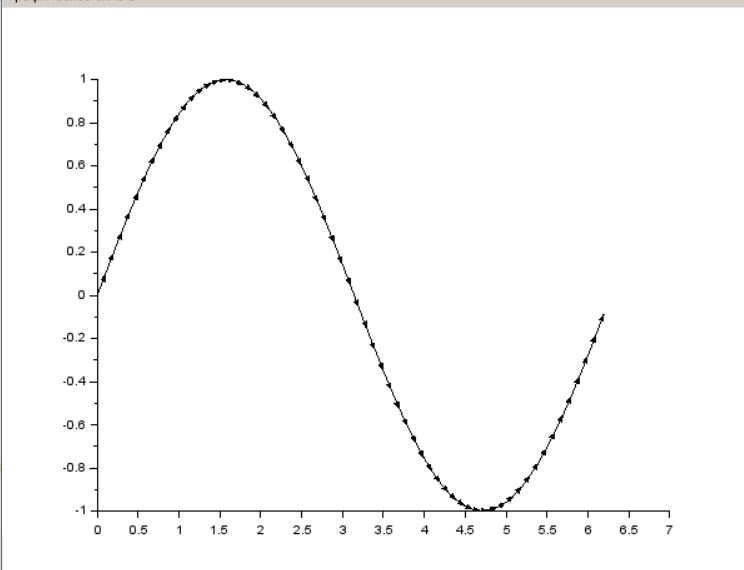


**plot2d2()**: Ця функція призначена для побудови двовимірних графіків. Вона може відображати функції, дані точок,

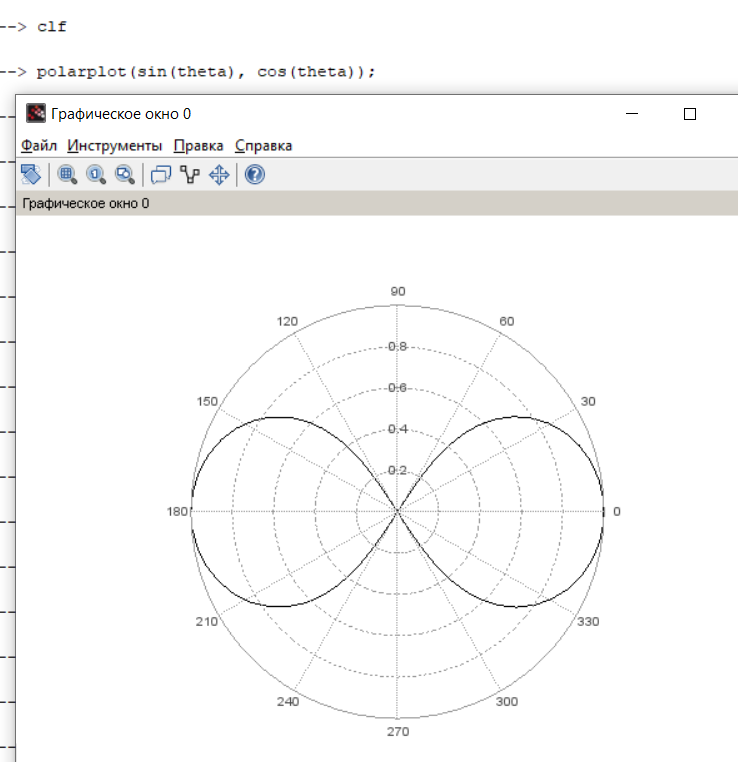


plot2d3(),

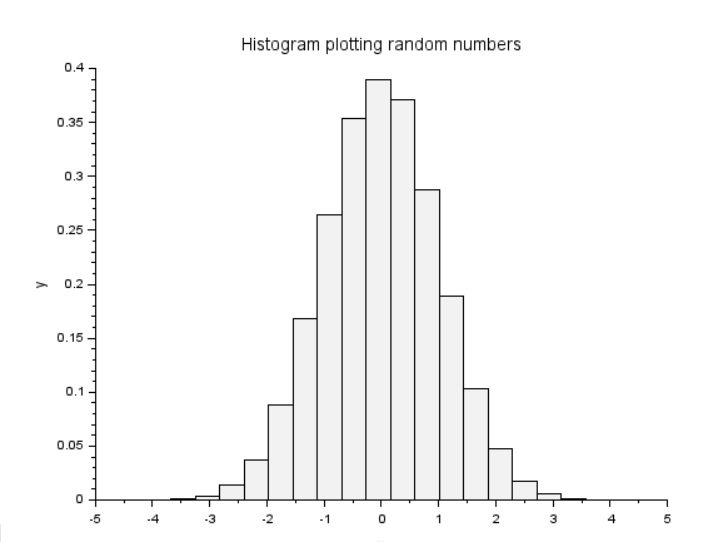




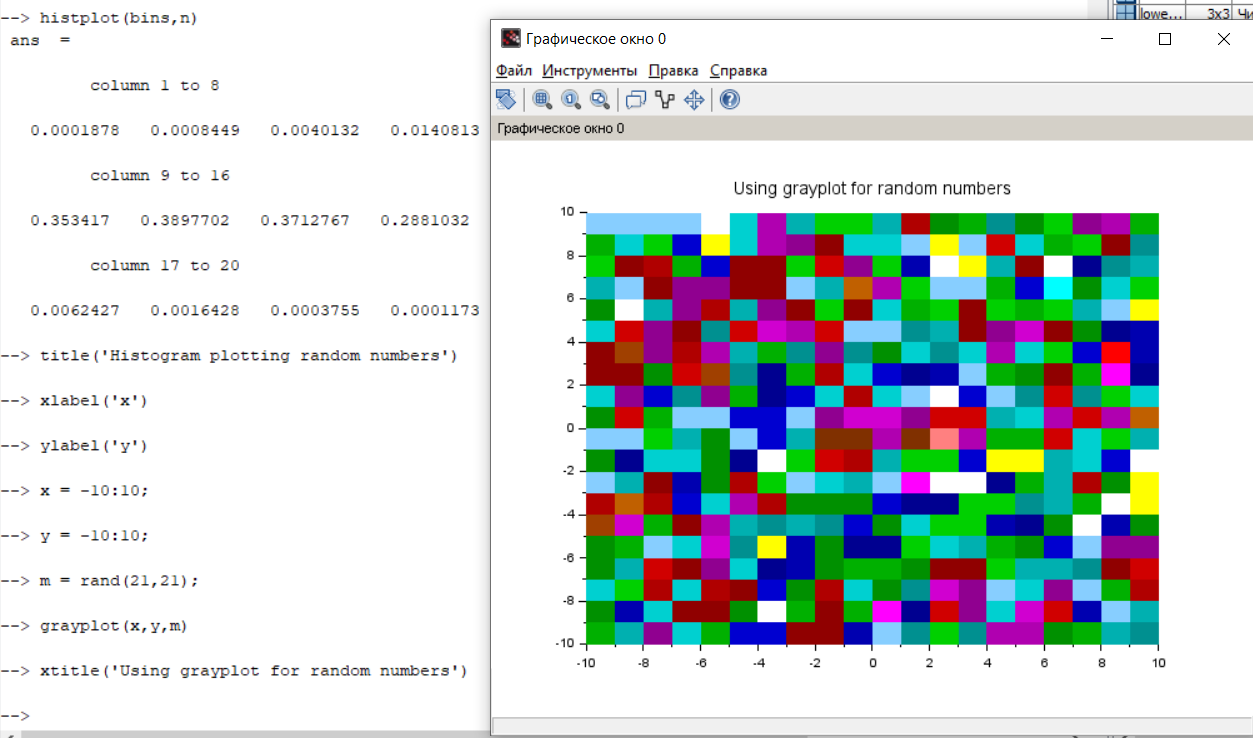
1. Під час математичного аналізу полярні координати стають важливими в тих випадках, коли не видно симетрії в картезіанських системах. Там замість x, використовують координати r, θ, які пов'язані рівняннями. Для побудови графіка можна скористатись функцією polarplot()

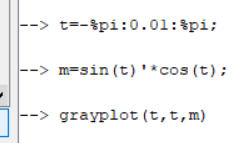


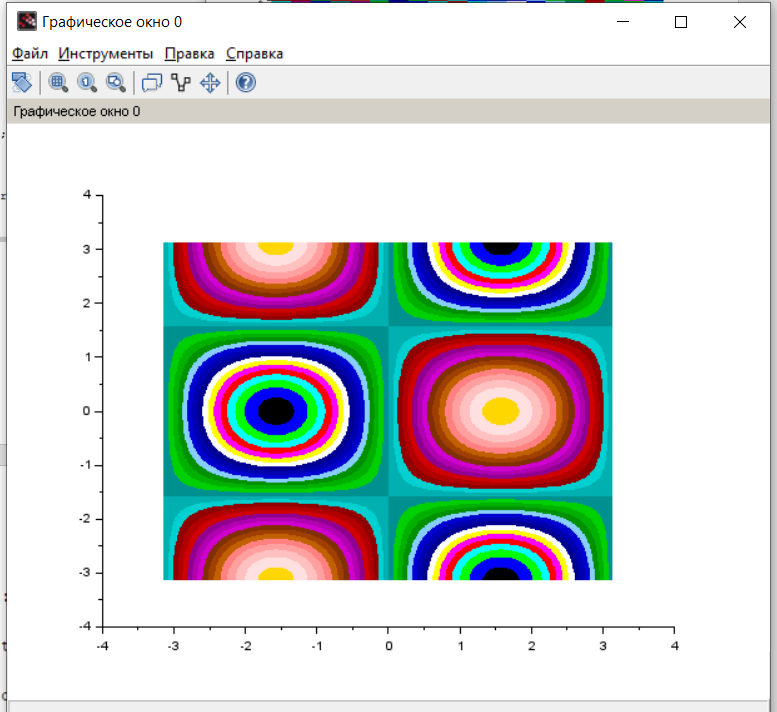
1. Якщо потрібно побудувати діаграму — можна застосувати histplot(). Гістограми традиційно використовуються для візуалізації числа подій, що відбуваються в межах різних діапазонів. В наведеному нижче прикладі використовується звичайний гаусівський розподіл:



1. Іноді нам потрібно побачити значення двомірної функції на 2D-графіку. Це робиться шляхом візуалізації функціонального значення z в координатах x і y. Кожен прямокутник у сітці заповнений сірим або колірним рівнем залежно від середнього значення z в кутах прямокутника. Щоб спробувати цю візуалізацію, використайте функцію grayplot():







1. Контурні карти часто використовуються для візуалізації проекції функції на 2D поверхні. Вони використовуються в різних галузях науки і техніки, особливо в геофізиці, де інформація про параметри вимірювання, як температура, тиск і вологість, проектується на карту. Для цієї мети Scilab має вбудовану функцію

contour2d().

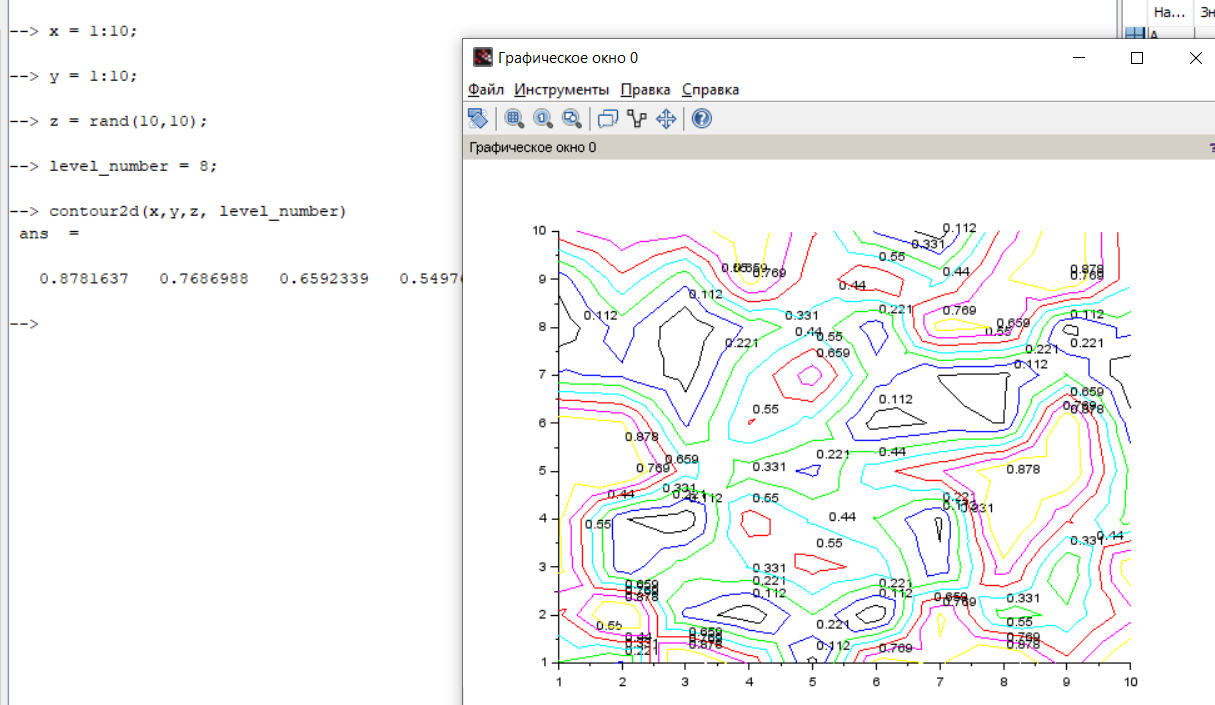
x = 1:10;

y = 1:10;

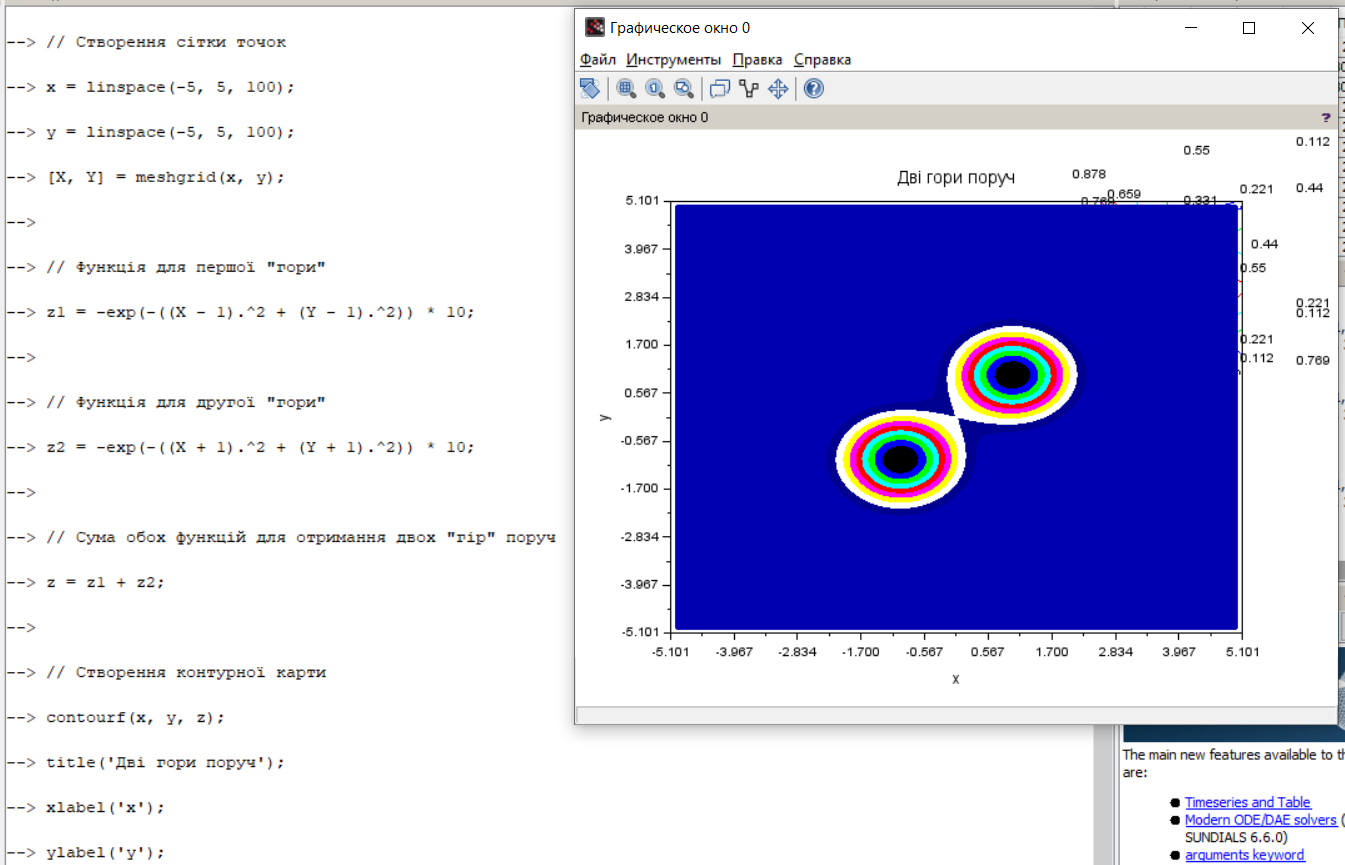
z = rand(10,10);

level\_number = 8;

contour2d(x,y,z, level\_number)



Самостійно побудуйте дві гори поруч, використовуючи схожу функцію contourf()



1. Настав час побудувати 3D-графік. Для цього скористаємось функцією

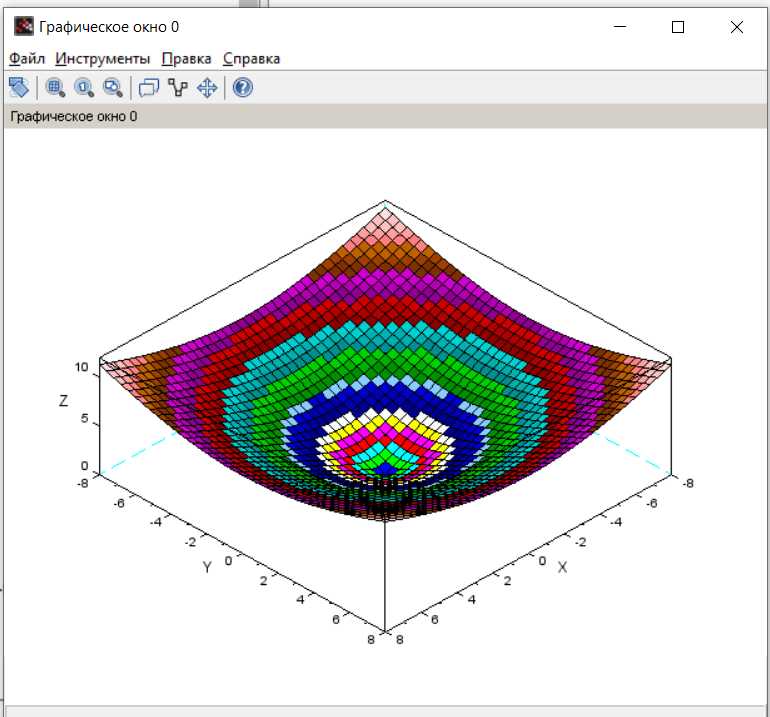
plot3d1(): a = linspace(-8,8,41)';

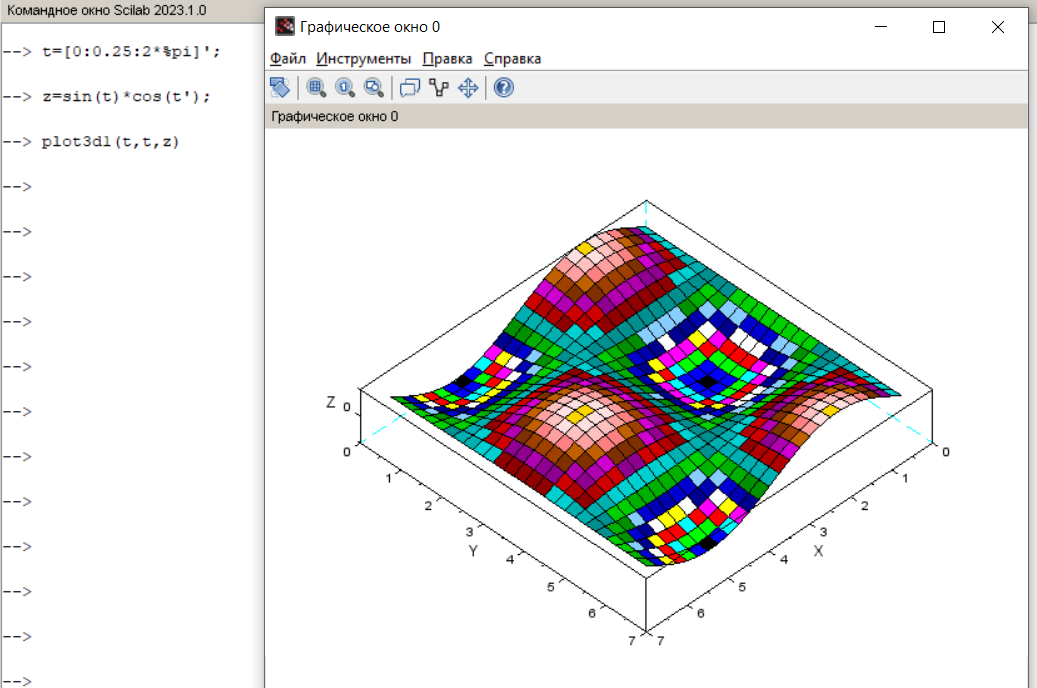
b = linspace(-8,8,41)';

[xx, yy] = meshgrid(a,b);

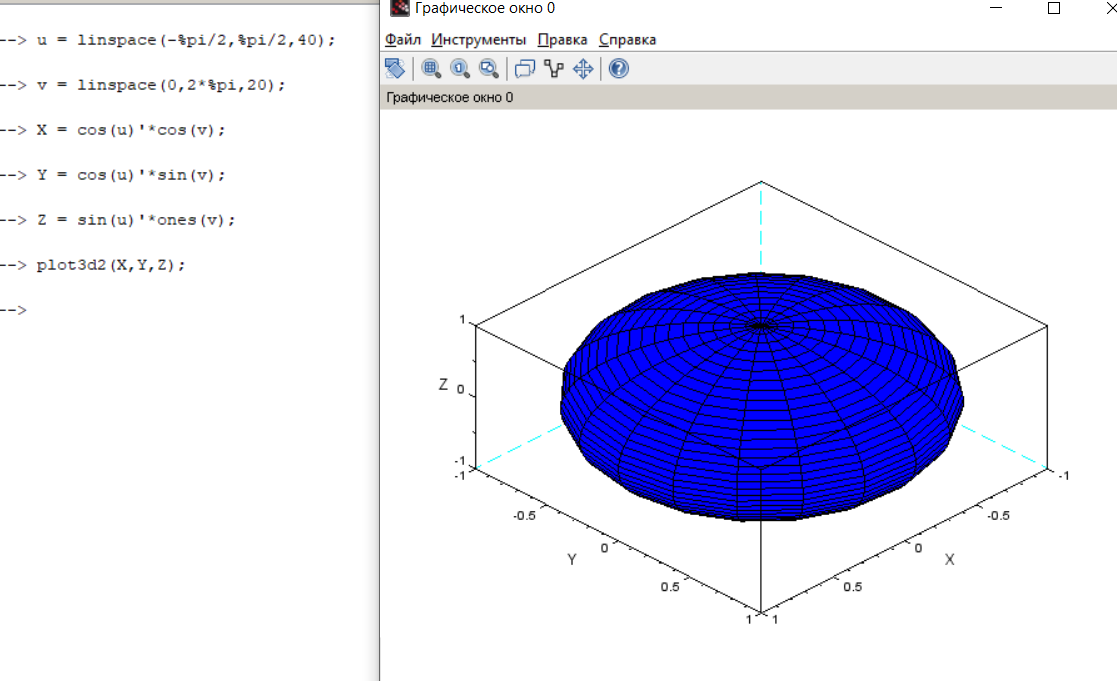
c = sqrt(xx.^2 + yy.^2)+%eps;

plot3d1(a,b,c)

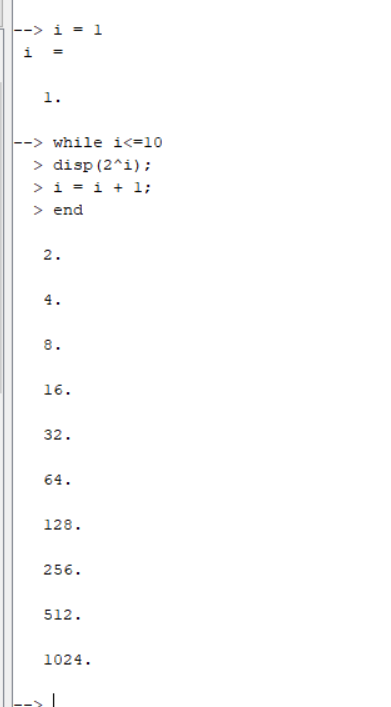




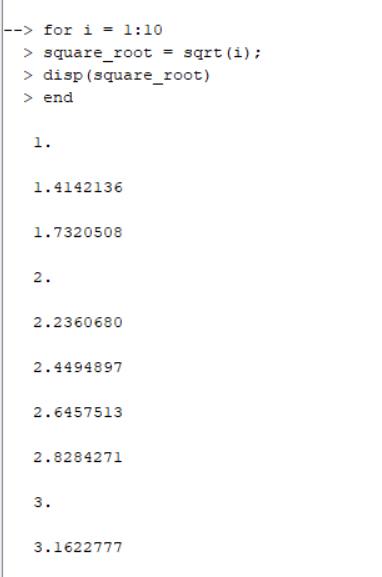
1. Якщо є plot3d1(), то логічно що є і plot3d2(), і plot3d3().



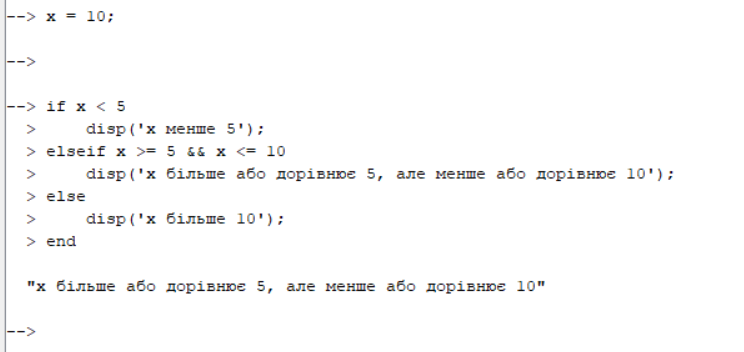
1. Цикл while визначає логічний стан і, поки він не задовільняється, він виконує блок коду. Синтаксис циклу while:



1. Синтаксис циклу for:



1. Синтаксис умовного оператора if-elseif-else:

Висновок: розглянуто синтаксис багатьох команд SciLab, обчислено та виведено в консоль результати роботи команд: математичних функцій, створення 2D та 3D графіків.

Також розглянуто роботу циклів for та while, і умовного оператора if