

Лекція 1

Лекція 1

Сайт: [Центр електронного навчання Харківського національного університету імені В. Н. Каразіна](#)
Курс: Символьні обчислення
Книга: Лекція 1
Надруковано: Невмержицька Олена Миколаївна
Дата: неділя, 15 лютий 2015 16:32

Зміст

[1 Стисла характеристика системи Maple](#)

[2 Типи даних Maple](#)

[3 Поняття про функції та оператори](#)

[4 Дії з виразами](#)

1 Стисла характеристика системи Maple

Для розв'язування різноманітних наукових математичних задач в сучасній науці використовують системи комп'ютерної математики (СКМ). Фірма Waterloo Maple, Inc (Канада) створила MAPLE — одну з найпотужніших та «розумних» інтегрованих систем символьної математики.

Maple об'єднує в собі

- 1) потужну мову програмування, редактор програм та документів,
- 2) можливість працювати у діалоговому режимі «введення команди, отримання відповіді системи»,
- 3) потужну довідкову систему з багатьма прикладами,
- 4) числовий та символьний процесор,
- 5) ядро алгоритмів та правил перетворення математичних виразів,
- 6) систему діагностики, бібліотеки вбудованих та додаткових функцій,
- 7) пакети функцій сторонніх виробників та підтримку інших мов програмування.

Maple може виконувати числові розрахунки, символьні перетворення, графічно відображати результати. До складу Maple входить досить розвинене середовище програмування. Крім того, вона містить надійні та ефективні символьні та числові алгоритми для виконання складних алгебраїчних перетворень, Maple вміє розв'язувати лінійні та диференціальні рівняння, системи рівнянь, задачі лінійної та тензорної алгебри, теорії груп, комбінаторики, статистики, найрізноманітніші задачі диференціального та інтегрального числення тощо.

Кілька слів про інтерфейс Maple.

В основі інтерфейсу користувача лежить графічний багато віконний інтерфейс операційної системи Windows. Керування системою Maple можливо за допомогою гарячих клавіш, панелей інструментів, головного меню та палітр. Також в Maple чудово реалізується довідкова система, яка надає повну інформацію англійською мовою про будь-який оператор, функцію або пакет з розібраними прикладами, які можна скопіювати та вставити в редактор або в вікно документів.

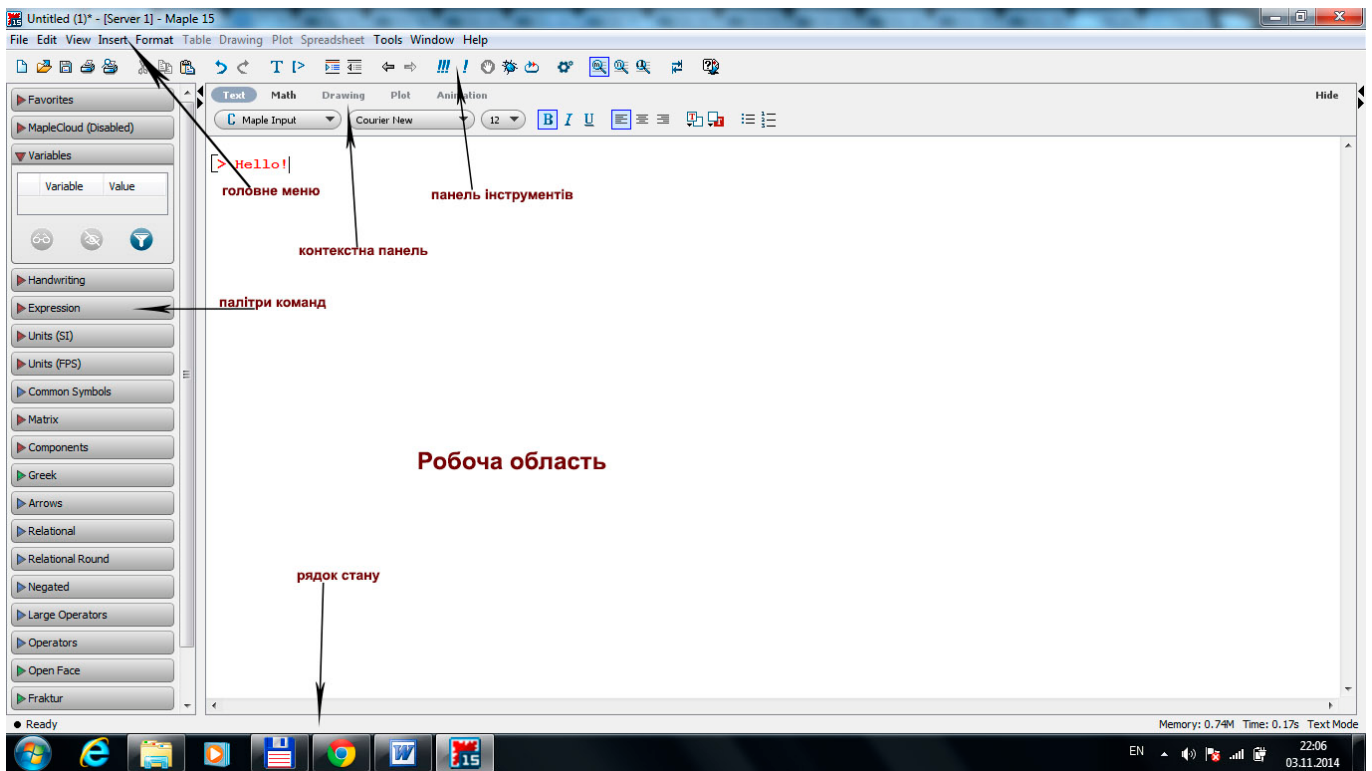
Користувач працює з документами, які одночасно є описами алгоритмів розв'язання задач, програмами та результатами їх виконань. Всі данні та результати команди розташовуються у відповідних комітках. Графічні побудови виконуються як в комітках документа, так і в окремих вікнах, вони мають своє меню для оперативного управління параметрами.

Початок роботи в Maple

Існує декілька інтерфейсів користувача для роботи в системі Maple: стандартний, класичний, командний рядок, графічний калькулятор та меплет-програми. Запуск кожного з них відбувається стандартно через меню Пуск або через відповідний значок на робочому столі Windows.

На малюнку 1.1.1 наведено приклад стандартного вікна системи.

Малюнок 1.1.1.



Основними елементами стандартного вікна є **голове меню**, **панель інструментів**, **палітри команд**, **контекстна панель**, **робоча область**, **рядок стану**.

Меню системи Maple

Головне меню системи Maple розташовано безпосередньо під рядком заголовка. Меню надає доступ до основних операцій та параметрів інтерфейсу користувача системи. Основні пункти меню документа:

- File — робота з файлами і друком документів;
- Edit — команди редагування документу і операції з буфером обміну;
- View — управління виглядом інтерфейсу;
- Insert — операції вставки;
- Format — операції завдання форматів;
- Spreadsheet — операції завдання таблиць;
- Options — завдання параметрів;
- Window — управління вікнами;
- Help — робота з довідковою системою.

За допомогою команд меню View ми можемо змінювати інтерфейс системи. В цьому меню можна побачити список палітр Palettes для вводу математичних знаків. Ці палітри можна вивести на екран і розташувати на будь-якому місці. Їх значення легко зрозуміти з назви.

SYMBOL — введення окремих символів (грецьких букв і деяких математичних знаків);

EXPRESSION — введення шаблонів математичних операторів і операцій;

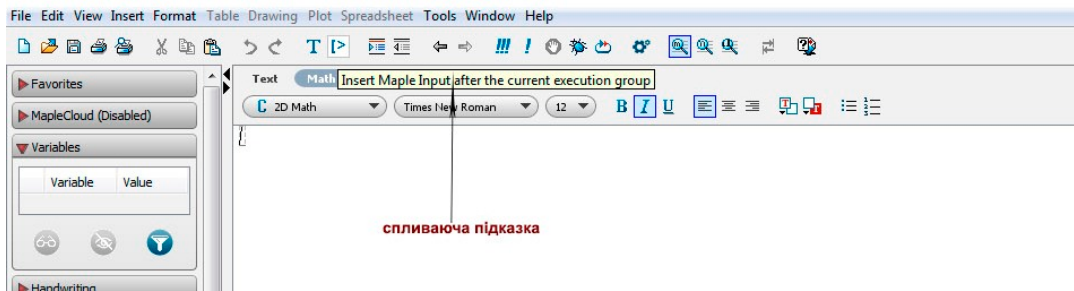
MATRIX — введення шаблонів **матриць** різних розмірів;

VECTOR — введення шаблонів векторів різних розмірів та типів (вектори-рядки та вектори-стовпчики).

Дуже зручний та корисний елемент інтерфейсу – спливаючі підказки. Вони з'являються, якщо навести курсор миші на той чи інший елемент інтерфейсу.

На малюнку 1.1.2. показана одна з спливаючих підказок.

Малюнок 1.1.2

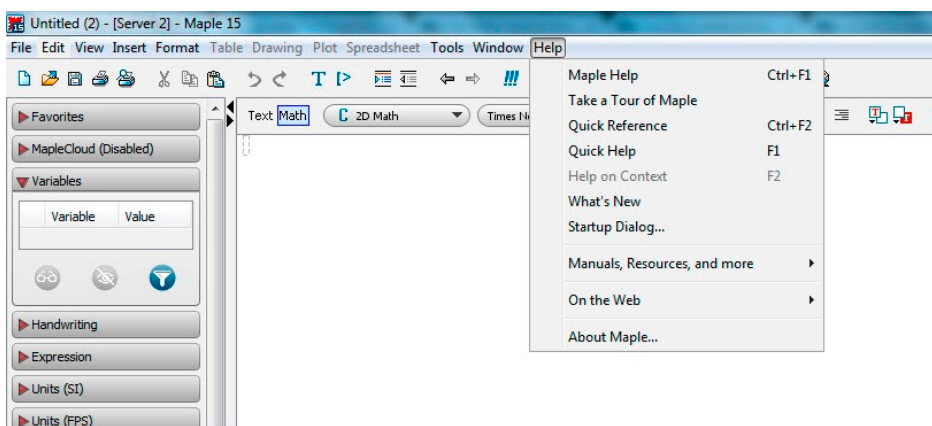


Меню Help

Довідковій системі Maple належить важлива роль. Вона надає повну інформацію про всі функції Maple. В ній наведено синтаксис функцій та операторів, приклади їх використання. Система побудована таким чином, що в ній можна швидко знайти необхідну інформацію.

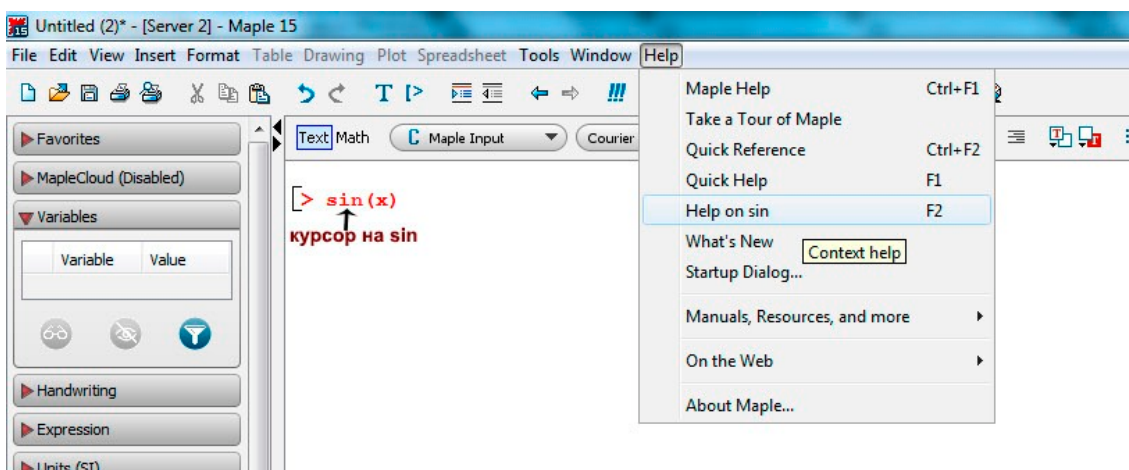
Основні команди для роботи в Maple 15 зосереджені в меню Help, малюнок 1.1.3.

Малюнок 1.1.3.



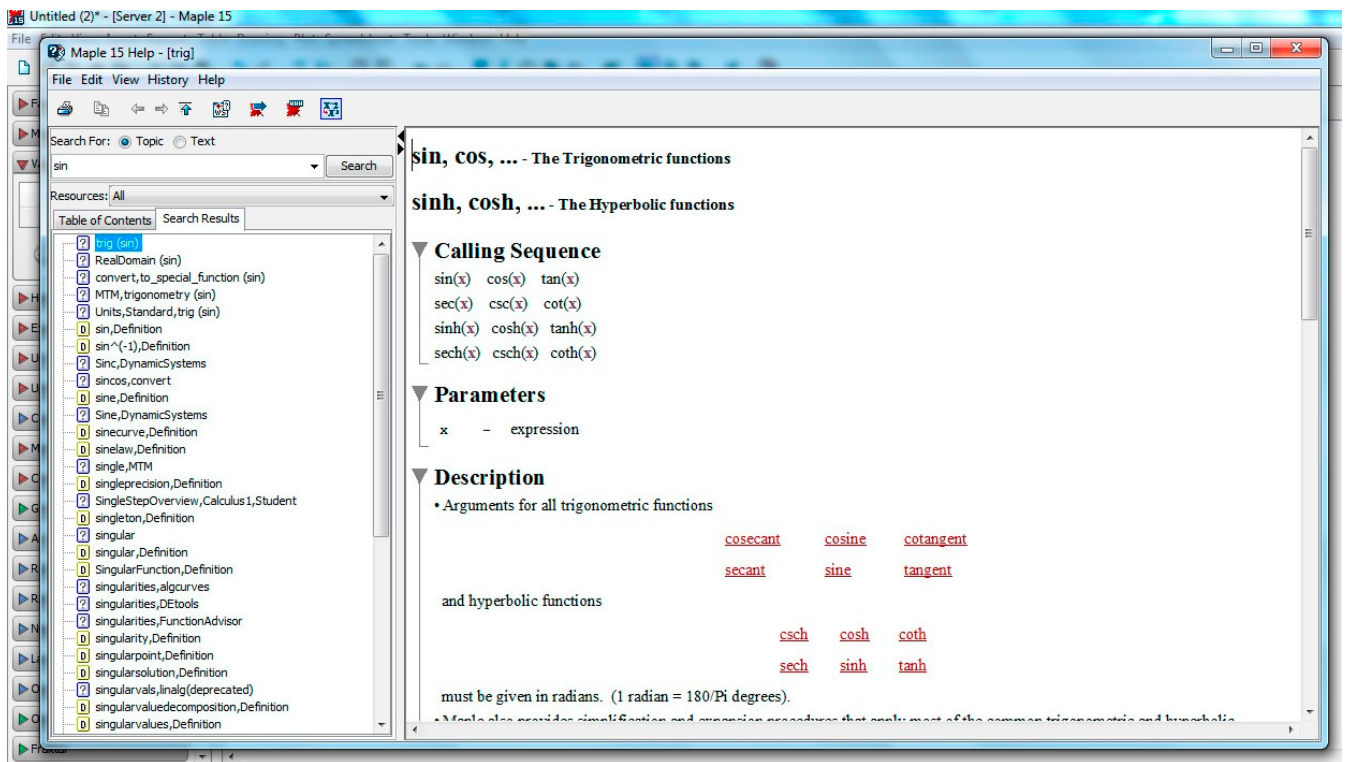
Ми бачимо, що меню Help достатньо громіздке, крім того, вся інформація в ньому надається англійською мовою, що може ускладнювати роботу з довідкою. Але користувачі швидко звикають до нього. До того ж меню Help є команда, яка прискорює роботу з довідковою системою. А саме, це команда **Help on context** (оперативна довідка по контексту). Її зручно використовувати у випадку, коли користувач бажає уточнити властивості та синтаксис об'єкту, що використовується, або під час розборів прикладів, якщо користувач зустрічає незнайомі функції, об'єкти, тощо. Щоб отримати оперативну довідку по контексту, необхідно встановити курсор на відповідний об'єкт, наприклад, на ім'я функції, та відкрити меню Help. Можна побачити, що операція **Help on context** зміниться на **Help on "..."**, де на місці крапок буде стояти слово, на якому зупинився курсор (мал.1.1.4).

Малюнок 1.1.4



На малюнку 1.1.5. показано, як виглядає вікно довідки для обраної функції $\sin x$.

Малюнок 1.1.5.



Більш докладнішу інформацію про меню Help можна знайти на офіційному сайті <http://www.maplesoft.com/support/help/>.

2 Типи даних Maple

Мова Maple

Maple є однією з найпотужніших мов програмування математичних задач, яка містить кілька тисяч операторів, команд і функцій, що входять у ядро, бібліотеку і пакети функцій.

Алфавіт мови Maple містить 26 малих латинських букв (від а до z), 26 великих латинських букв (від А до Z), 10 арабських цифр (от 0 до 9) та 32 спеціальних символів (арифметичні оператори +, -, *, /, знак піднесення до ступеню ^ та інші). Існує п'ять пар альтернативних символів (що мають однакові значення):

« та ** [та ([та)] та) { та ({ та) }

До спеціальних елементів відносяться такі елементи мови:

- 1) % — системна змінна, що зберігає результат попередньої операції;
- 2) : — фіксатор виразу, який попереджає вивід результату обчислення в комірку виводу;
- 3) ; — фіксатор виразу, що надає вивід результату обчислення в комірку виводу;
- 4) # — вказівник програмного коментаря
- 5) « — обмежувач рядку (наприклад, 'string');
- 6) := — оператор присвоєння (привласнення?) (наприклад, z:=7);
- 7) : ; — пустий оператор;
- 8) :: — вказівник типу змінної (наприклад, m::integer або x: -complex);
- 9) \ — знак обененого ділення, який має багато значень в залежності від контексту (справка по цьому знаку – **backslash**).

Також в Maple використовують зарезервовані слова для створення умовних виразів, циклів, процедур та керуючих команд.

Таблиця 1.3.1. Мова Maple

And	break	by	catch	description
Do	done	el if	Else	end
Error	export	fi	finally	for
From	global	if	In	intersect
Local	minus	mod	module	next
Not	od	option	options	or
Proc	quit	read	return	save
Stop	then	to	Try	union
Use	while			

Типи даних Maple

Система Maple працює з даними та обробляє їх. Існує досить багато різних типів даних у Maple та засобів роботи з ними.

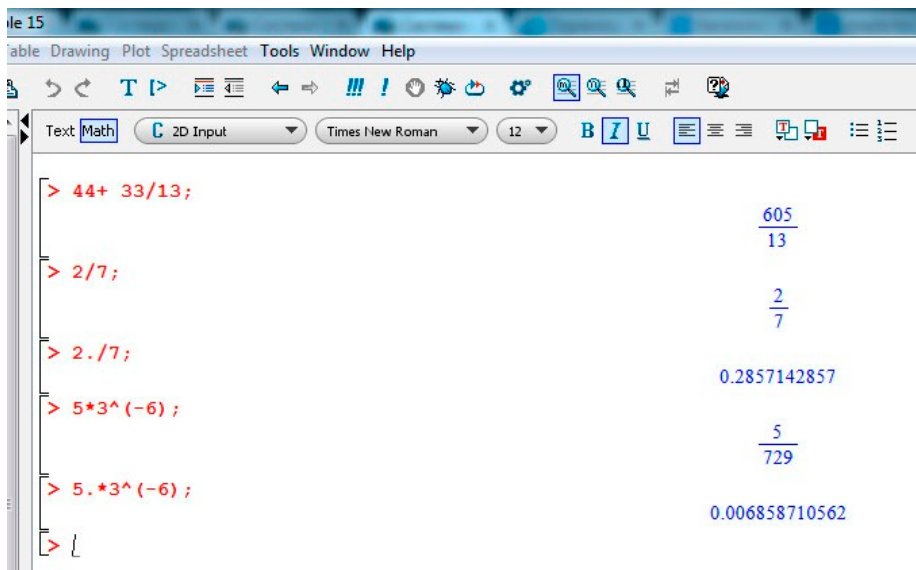
Прості типи даних

До простих типів даних відносять числа та числові константи: цілі числа (0, 1, 123, -456, ...), раціональні числа

(1/2, -123/4, ...), дійсні числа з плаваючою точкою (0.12, -0.34, ...), дійсні числа з мантиєю та порядком (1.29E5, -3.5E-10), комплексні числа (2+3*I).29

Приклад дій з числами наведений в наступному малюнку.

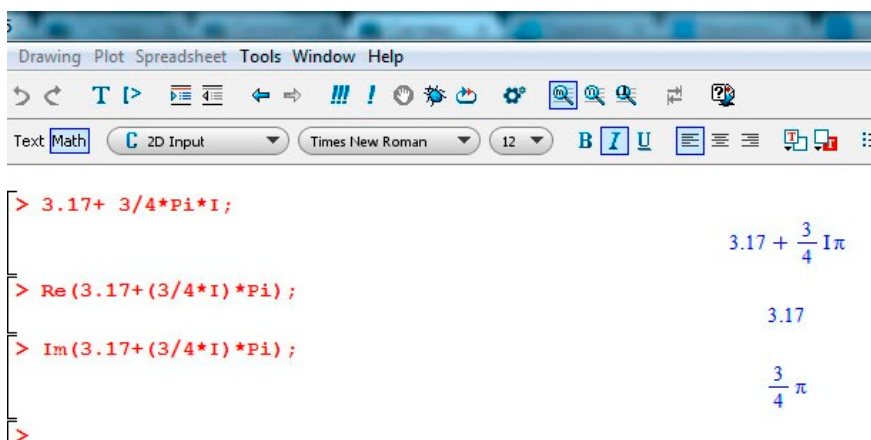
Малюнок 1.3.1.



Для чисел з плаваючою точкою є особливість. Кількістю цифр, що виводяться після десяткової точки, можна керувати, задаючи її значення системною змінною Digits. За замовчуванням Digits= 10.

Для задання комплексного числа в алгебраїчній формі використовується константа уявної одиниці – I . Для виведення дійсної частини комплексного числа використовується функція **Re** (число), а уявної – функція **Im** (число):

Малюнок 1.3.2.



Дані множинного типу

Набори (або множини) – це ряд будь-яких виразів, які задаються за допомогою фігурних дужок $\{ \}$. Відмінною властивістю наборів є те, що Maple автоматично усуває елементи з однаковим значенням, а також упорядковує елементи за збільшенням значення (якщо числа) та за алфавітом (якщо символи та рядки):

> {1,2,3,4,5};

{1,2,3,4,5}

> {x,y,5,z,2,x,3,5};

{2,3,5,x,y,z}

> {"peace", "friends", "happiness"};

{"peace", "friends", "happiness"}

Списки – це упорядковані набори. Вони створюються в квадратних дужках $[]$. Особливістю списків є те, що їх

елементи перетворюються та виводяться лише у тому порядку, в якому були задані:

```
> x:=[1,2,3,4];
```

```
x:=[1,2,3,4];
```

```
> x:=[1,2,3.4,5.65];
```

```
x:=[1,2,3.4,5.65];
```

```
> [cos(a), sin(b),cos(z-y)];
```

```
[cos(a), sin(b),cos(z-y)];
```

Масиви. Масивами називаються узагальнені списки. Вони мають дві

відмінні властивості, а саме:

- а) розмірність масивів може бути більшою за одиницю;
- б) індекси елементів – натуральні числа.

Для створення масивів використовується функція **Array**. Зазвичай вона має наступний вигляд:

- `array[a..b, s1]` — вектор з індексами від `a` до `b` та значеннями в одновимірному списку `s1`;
- `array[a..b, c..d, s2]` — матриця з номерами рядків від `a` до `b`, номерами стовбчиків від `c` до `d` та значенням в двовимірному списку `s2`.

Приклади використання команди:

```
> Array(1..2,1..2,[[a,b],[c,d]]);
```

 задає матрицю 2×2

```
> F:=Array(1..3,1..3,[[a,b,c],[d,e,f],[g,h,i]]);
```

 задає матрицю 3×3

```
> B:=Array(1..3,[u,v,u-v]);
```

 задає вектор з координатами `u`, `v` та `u-v`.

Малюнок 1.3.3

```
> Array(1..2,1..2,[[a,b],[c,d]]);
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
> F:=Array(1..3,1..3,[[a,b,c],[d,e,f],[g,h,i]]);
```

$$F := \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

```
> B:=Array(1..3,[u,v,u-v]);
```

$$B := \begin{bmatrix} u \\ v \\ u - v \end{bmatrix}$$

Таблиці – ще один важливий тип множинних даних. Вони задають дані з довільною індексацією. Тобто це масиви, для яких користувач може задавати будь-які індекси, не лише натуральні числа. Для створення таблиць існує функція **table**.

В наступному прикладі за індекси використані букви латинського алфавіту `a`, `b`, `c`.

Малюнок 1.3.4.

Вектори та матриці

Вектори та матриці використовуються для задач лінійної алгебри та векторного числення. Для роботи з ними існує спеціальний пакет **LinearAlgebra**. В наступному прикладі показано, як створити вертикальні та горизонтальні вектори.

Малюнок 1.3.5

```

> A:=table([a=1,b=2,c=3]);

A := table([ a = 1, b = 2, c = 3])

> A[1];

A1

> A[c];

3

> A[a];

1

>

```

```

> with(LinearAlgebra):
>
> A=<3,2,4>;

A =  $\begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix}$ 

> V=<a|b|c>;

V =  $\begin{bmatrix} a & b & c \end{bmatrix}$ 

> {

```

Для створення [матриць](#), а також для виконання нескладних операцій із ними, використовується команда **Matrix**. Синтаксис цієї команди відрізняється залежно від задач.

Малюнок 1.3.6

```

> with(LinearAlgebra):
>
> A=Matrix(1..2,1..3,5);

A =  $\begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix}$ 

> B=Matrix([[a,b],[c,d],[e,f]]);

B =  $\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$ 

>

```

Рядки та коментарі

Рядки як тип даних — це ланцюг символів. Рядки беруть в подвійні лапки (" ").

```

> S:="Good luck!";

S:="Good luck!";

> S[6...9];

"luck";

```

Програмні коментарі

Часто виникає необхідність в коментарях до обчислень. Будь який текст після знаку **#** розглядається як коментар, що не виконується, навіть якщо це математичний вираз. При цьому він не обчислюється. Наприклад:

```

> 5*5; #то був приклад, а це коментар: 6*7=, він не виконується

```

Константи

Константи – це найпростіші іменовані об'єкти з наперед визначеним значенням. Константи можна поділити на такі групи:

- а) числові константи;
- б) рядкові константи;
- в) константи, вбудовані в ядро Maple.

Звичайні **числові константи** не мають імен, вони представлені звичайними числами. Іменем такої константи вважається її числове значення.

Під **рядковими константами** мають на увазі довільні ланцюги символів, які поміщають в апострофи: 'geometry', '5656'. Рядкові константи не можна використовувати в арифметичних виразах.

Константи, вбудовані в ядро Maple, вважаються заздалегідь визначеними глобальними змінними.

- *false* — логічне значення «хибно»;
- *true* — логічне значення «істино»;
- *gamma* — константа Ейлера, що дорівнює 0.5772156649...;
- *infinity* — плюс нескінченність;
- *-infinity* - мінус нескінченність;
- *Catalan* — константа Каталана, яка дорівнює 0.915965594...;
- *FAIL* — помилка, неможливо встановити чи існує помилка (дивись довідку);
- *I* — уявна одиниця (квадратний корінь з -1);
- *Pi* — константа 3.141...

Зауважимо, що серед зазначених констант немає виділеної для числа $e = 2,71828...$. Для того, щоб задати цю константу, в Maple використовують експоненціальну функцію ***exp(1)***, або ***exp(1.0)***.

Змінні

Змінні — це об'єкти, які змінюються в процесі виконання документа. Ім'я змінної повинне бути унікальним, складатися з букв та/або цифр, обов'язково починатися з букви. Не можна давати змінним імена, що повторюють ключові слова системи Maple, та включати до імені знаки операторів. Тип змінних визначається значенням, яке йому надають. Це може бути

- 1) *цілочислова (integer)*,
- 2) *раціональна (rational)*,
- 3) *дійсна (real)*,
- 4) *комплексна (complex) або рядкова (string) змінні*.

Також змінні можуть бути символьного типу, їх значенням є математичні вирази, або типа списку.

Щоб явно задати тип змінної, використовують наступну конструкцію: ***type(var,type)***, де ***var*** — ім'я змінної, ***type*** — тип, на який проводиться перевірка. Команда вертає значення ***true*** або ***false*** («істина» або «хибність»).

3 Поняття про функції та оператори

Функції

Важливим поняттям системи Maple 7 є поняття **функції**.

Функція повертає результат деякого перетворення початкових даних – параметрів функції. Maple має багато вбудованих функцій, які входять до його ядра та пакетів.

Щоб задати функцію, необхідно ввести її ім'я та параметри, які містяться в круглих дужках. У функції може бути кілька параметрів. Наприклад, функція ***sqrt(3)*** задає функцію обчислення квадратного корня з числа 3. У відповідь на звернення до функції, вона повертає значення. Наприклад:

```
> cos(2);
```

```
cos(2)
```

```
> cos(2.0);
```

```
-0.4161468365
```

Прийнято називати функціями і ті команди, які повертають, наприклад, не числове значення, а результат іншого типу. Наприклад: функції ***plot*** і ***plot3d***, вони повертають графіки аргументів.

Математичні функції

Операції з цілими числами

ifactor(a) – розкладання на множники

igcd(a,b) – найбільший спільний дільник

iquo(a,b) – ціле від ділення a на b

ictm(a,b) – найменше спільне кратне

irem(a,b) – залишок від ділення

isqrt(a) – ціле від кореня квадратного

Степеневі та логарифмічні функції

sqr(x) – квадратний корінь

exp(x) – експонента

log(x) – загальний логарифм

ln(x) – натуральний логарифм

log10(x) – десятковий логарифм

log[a](x) – логарифм за заданою основою

Тригонометричні функції

sin(x) – синус аргументу x

cos(x) – косинус аргументу x

tan(x) – тангенс аргументу x

cot(x) – котангенс аргументу x

arcsin(x) – арксинус аргументу x

arccos(x) – арккосинус аргументу

Гіперболічні функції

$\sinh(x)$ – синус гіперболічний

$\cosh(x)$ – косинус гіперболічний

$\operatorname{arccosh}(x)$ – арккосинус гіперболічний

$\operatorname{arcsinh}(x)$ – арксинус гіперболічний

Функції з елементами порівняння

$\operatorname{ceil}(x)$ – знаходить найменше ціле, яке більше або дорівнює x

$\operatorname{trunk}(x)$ – ціле, округлене в напрямку $x = 0$

$\operatorname{floor}(x)$ – знаходить найбільше ціле, яке менше або дорівнює x

$\operatorname{round}(x)$ – округлене значення числа (до цілого)

$\operatorname{frac}(x)$ – дробова частина числа

$\operatorname{signum}(x)$ – функція знака числа

Функції над комплексними числами

$\operatorname{Re}(x)$ – знаходить дійсну частину комплексного числа

$\operatorname{Im}(x)$ – знаходить уявну частину комплексного числа

$\operatorname{abs}(x)$ – обчислює модуль дійсного або комплексного числа

$\operatorname{polar}(x)$ – задає комплексне число у полярному вигляді

$\operatorname{argument}(x)$ – знаходить аргумент комплексного числа

$\operatorname{conjugate}(x)$ – знаходить комплексно спряжене число

Обчислення похідних. Функція diff

Існує декілька шляхів обчислення похідних. Найпростіший із них – використання шаблонів або із палітри команд **Expression**. Еквівалентом шаблонів в Maple є команда **diff(function, $x_1, x_2, \dots, x_n$)**, де $x_1, x_2, \dots, x_n$ – послідовність незалежних змінних функції *function*, за якими шукають похідну.

Якщо потрібно знайти похідну першого порядку від функції однієї змінної, то використовують запис **diff(f(x))**. Для похідної вищих порядків, використовують запис з оператором послідовності $\$$: **diff(f(x), x\$3)**, що еквівалентно **diff(f(x), x, x, x)** або **diff(diff(diff(f(x), x), x), x)**. Якщо *function* – функція декількох змінних, в цьому випадку шукають частинні похідні. Наприклад, щоб обчислити похідну $\frac{\partial^4 f^4(x,y)}{\partial x^3 \partial y}$ використовують запис **diff(f(x, y), x\$3, y)**.

Малюнок 1.2.1

The screenshot shows the Maple command line with the following commands and their results:

- Command: `> diff(ln(x), x);` Result: $\frac{1}{x}$
- Command: `> Diff(ln(x), x);` Result: $\frac{d}{dx} \ln(x)$
- Command: `> diff(x^5*cos(y), x$3, y$2);` Result: $-60x^2 \cos(y)$
- Command: `> Diff(x^5*cos(y), x$3, y$2);` Result: $\frac{\partial^5}{\partial y^2 \partial x^3} (x^5 \cos(y))$

Зауважимо, що функція **Diff** лише відображає зручний математичний вигляд похідної, але не знаходить її.

Оператори

Оператори як об'єкти Maple самі по собі результат не повертають. Але спільно з функціями і операндами (константами або змінними) допомагають конструювати математичні вирази. Залежно від кількості операндів розрізняють **бінарні**, **унарні**, **нульарні** оператори, і, крім того, існує особливий вид оператора – **функціональний**. Розглянемо види операторів окремо.

Бінарні оператори

Бінарні оператори використовуються з двома операндами. Основні з них перелічені в табл. 1.2.1.

Таблиця 1.2.1. Бінарні оператори

Оператор	Операція
+	додавання
-	віднімання
*	добуток
/	ділення
^	піднесення до степеня
mod	остача від ділення
=	дорівнює
:=	присвоєння
<	менше
<=	менше або дорівнює
>	більше
>=	більше або дорівнює
<>	не дорівнює
@	композиція функцій

Найуживанішим оператором є оператор присвоювання :=. Він використовується надання змінним конкретних значень. Наприклад:

```
> x := y;
```

```
    x := y;
```

```
> y := 5;
```

```
    y := 5;
```

```
> x;
```

```
    5
```

Ще один з часто вживаних операторів – це оператор рівності $=$. Він використовується для запису рівностей та логічних умов (' $x=y$ ', ' $a=1..2$ ').

Інші приклади використання бінарних операторів:

$> 9^3;$
 729
 $> 23 \bmod 8;$
 7
 $> x @ x;$
 $x^{(2)}$
 $> x @ (x @ x);$
 $x^{(3)}$
 $> 6/4^2-5;$
 $\frac{-37}{8}$

Унарні оператори

Ці оператори відповідно використовуються з одним операндом. Якщо оператор стоїть перед операндом, то це префіксний

оператор, якщо після операнда – постфіксний.

Таблиця 1.2.2. Унарні оператори

Оператор	Операція
+	Унарний плюс
-	Унарний мінус
!	Факторіал
not	Логічне ні
.	Десяткова точка
\$	Оператор послідовності
&name	Нейтральний оператор

Приклади використання унарних операторів:

$> [-a, x+(-x), a+(+a)];$
 $[-a, 0, 2 a]$
 $> 4!;$
 24
 $> 1.34;$
 1.34
 $> a\$3;$
 a, a, a

Функціональні оператори є альтернативами функцій. Їх вигляд наданий в наступній таблиці.

Таблиця 1.2.3. Функціональні оператори

Оператор	Запис оператора
«arrow» (стрелочная)	vars -> result
«angle bracket» (в угловых скобках)	<result vars>

Функціональні оператори можна використовувати для підстановок. Наприклад, запис $x \rightarrow x^2$ означає підстановку x^2 на місце змінної x .

Диференціальний оператор D

Для створення функцій із похідними можна використовувати спеціальний оператор – диференціальний оператор D . Він дозволяє створювати більш компактні вирази, ніж функція `diff`.
Формат запису оператора:

- $D(f)$, де f – функція одного аргументу;
- $D[i](f)$, де f – функція кількох аргументів; i – додатне ціле число, виражає їх послідовність, за якими проводиться диференціювання.

Для того щоб створити похідну, необхідно даним оператором подіяти на певну змінну (змінні). Наприклад, обчислити похідну функції

Приклад 2.11. Знайти похідну функції $\ln(x)\cos(x)$.

Малюнок 1.2.2



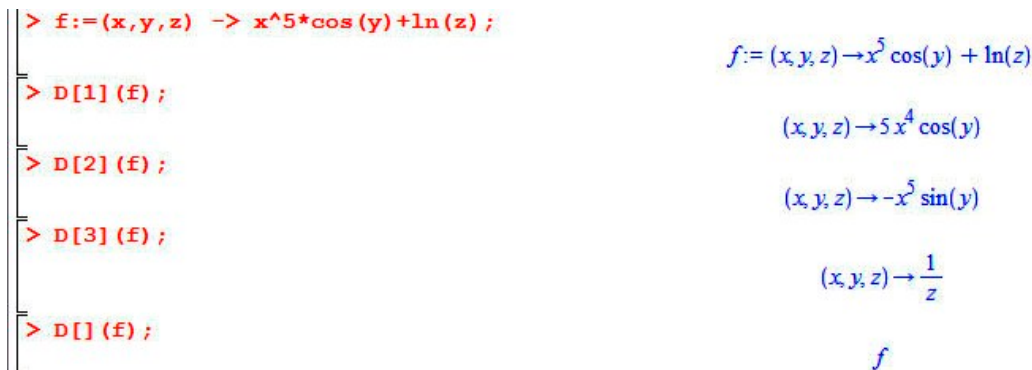
```

> D(ln*cos)(x);
      cos(x)
      -----
       x
- ln(x) sin(x)

> D(ln*cos)(1);
      cos(1)
  
```

Розглянемо ще один приклад.. Знайти частинну похідну функції трьох змінних за кожною змінною:

малюнок 1.2.3



```

> f := (x,y,z) -> x^5*cos(y) + ln(z);
      f := (x,y,z) -> x^5 cos(y) + ln(z)

> D[1](f);
      (x,y,z) -> 5 x^4 cos(y)

> D[2](f);
      (x,y,z) -> -x^5 sin(y)

> D[3](f);
      (x,y,z) -> 1/z

> D[](f);
      f
  
```

Якщо квадратні дужки залишити пустими, похідна не обчислюється.

Щоб обчислити похідну вищих порядків, використовується наступний вигляд оператора D : $(D@@n)(f)$, де n – порядок похідної.

Наприклад:

малюнок 1.2.4



```

> D(D(g));
      D^(2)(g)

> (D@@2)(g);
      D^(2)(g)
  
```

Порівняйте дві форми запису похідної, представлених на попередньому малюнку.

Зауважимо, що оператор D знаходить похідні від операторів, а функція `diff` – від виразів. Отже, аргументом і результатом D є функціональні оператори, а аргументом і результатом `diff` є вирази.

Інші оператори

Оператор % забезпечує підстановку в рядок вводу бао в вираз останнього результату операції, $x\%$ - предостаннього і $\%\%\%$ - третього з кінця.

- **eval (array)** — повертає обчислені елементи масива array;
- **evalf(expr, n)** — обчислює expr і повертає обчислене значення у формі числа з плаваючою точкою, що має n цифр після десяткової точки;
- **evalf(int(f, x=a..b))** — оцінює та повертає значення визначеного інтеграла $\int(f, x=a..b)$;
- **evalm(mexpr)** — обчислює значення матричного виразу mexpr і повертає його.

Обробка та індикація помилок

При роботі з системою Maple необхідно дотримуватись правил коректного вводу виразів та інших об'єктів мови Maple, тобто не порушувати правил синтаксису. Важливо, що помилки в виразах, які не порушують правила синтаксису Maple, система не розпізнає. Наприклад, якщо в виразі $\text{diff}(\sin(x), a)$ замість змінної x ви задасте параметр a , отримаєте в результаті диференціювання 0. Ця помилка не буде розпізнаною. Контроль за подібними помилками повністю лягає на користувача.

Решту помилок система розпізнає за допомогою вбудованого синтаксичного аналізатору. Наприклад, якщо ви задали неправильне ім'я функції, то це буде опізнано синтаксичним аналізатором і обчислення не будуть виконуватись:

```
>sqr(4.0);
```

```
sqr(4.0)
```

В цьому прикладі ім'я функції `sqr` введено невірно. Maple сприймає її як деяку функцію, що введена користувачем. Тому програма просто повторює запис і не повідомляє про помилку. В іншому прикладі функція введена правильно, але дісне число задано невірно (використовується кома замість крапки в десятковому запису). Maple «впізнає» помилку:

```
>sqr(4,0);
```

```
Error, (in sqrt) expecting 1 argument, got 2
```

4 Дії з виразами

Система Maple має широкі можливості для аналітичних, та числових обчислень.

Символьні обчислення – це математичні перетворення виразів, що містять: а) символьні або абстрактні величини (змінні, функції, оператори); б) точні числа (цілі числа, раціональні, π , e та інші). Мета цих перетворень – отримання виразів у більш простій формі або встановлення співвідношень з іншими, більш зрозумілими, формулами.

Числові обчислення – це застосування до виразів арифметичних операцій зі скінченною точністю. Вирази в такому разі містять конкретні числа. Точні числа замінюються на наближені числа з плаваючою точкою із заданою точністю.

Перетворення виразів

Для виконання найбільш поширених дій над виразами взагалішому (символьному) вигляді в системі Maple передбачені такі функції:

1) *simplify()* – використовується для спрощення виразів. Це багатоцільова функція, вона забезпечує спрощення математичних виразів, виконуючи наступні типові дії:

- комбінує цифрові підвирази: $2*x*8 \rightarrow 16*x$; $16*y/8 \rightarrow 2*y$;
- зведення подібних доданків: $x+2*x+8+3*x \rightarrow 6*x+8$; $3*y^2*y^2 \rightarrow 6*y^4$;
- зниження степеня поліномів там, де це можливо;
- скорочує вираз на найбільшій спільний множник чи на інший множник;

та інше. Але функція *simplify* не завжди здатна виконати можливі спрощення. В цьому випадку її необхідно "допомогти", тобто вказати, де шукати спрощення і де знайти відповідне перетворення.

Приклади використання функції див. на мал. 1.4.1:

Малюнок 1.4.1

```
> simplify(1/2*16-3*5);
-7
> simplify(cos(x)-sin(3*x)+cos(3*x)-sin(x));
2 cos(x) (-2 sin(x) cos(x) - 1 + 2 cos(x)^2)
> simplify(sin(x)^2+cos(x)^2, trig);
1
```

2) *factor()* – розкласти на множники. Приклад використання функції:

Малюнок 1.4.2.

```
> factor(x^2-10*x+25);
(x-5)^2
> factor(a^3-6*a^2*b+12*a*b^2-8*b^3);
(a-2*b)^3
> factor(x^3*y-x^2*y^2+4*x^4-x^2);
x^2 (xy-y^2+4x^2-1)
> [
```

3) *expand()* – розкрити дужки.

Малюнок 1.4.3

```
> expand(cos(x-y));
cos(x) cos(y) + sin(x) sin(y)
> expand((x-y)*(x+y));
x^2 - y^2
```

4) *combine()* – забезпечує об'єднання коефіцієнтів степеневих функцій та перетворення тригонометричних та

деяких інших функцій:

> combine(cos(x)^2- sin(x)^2);

$\cos(2x)$

> combine(log(exp(x))-2*log(exp(x)));

$-\ln(\exp(x))$

5) convert() – проста але потужна функція. Вона надає можливість перетворювати вирази в інші форми. Існує більш ніж сім десятків параметрів, які ми можемо задавати цією функцією. Їх перелік можна знайти в довідці по функції *convert*.

Наприклад, параметр *binary* перетворює задане число в двійкову систему:

> convert(7,binary);

111

Параметр *vector* перетворює список у вектор:

> convert([a,b,c,d],vector);

[a,b,c,d]

Інші приклади використання функції на мал.1.4.4:

Малюнок 1.4.4

```
> simplify(1/2*16-3*5);
-7

> simplify(cos(x)-sin(3*x)+cos(3*x)-sin(x));
2 cos(x) (-2 sin(x) cos(x) - 1 + 2 cos(x)^2)

> simplify(sin(x)^2+cos(x)^2, trig);
1
```

6) normal() – звести до нормального вигляду з розкладанням на множники:

Малюнок 1.4.5

```
> normal((cos(x)-cos(x)*sin(x)^2)/cos(x));
1 - sin(x)^2

> normal((x^2-y^2)/(x+y));
x - y
```

7) sort() – сортування елементів (упорядкування):

Малюнок 1.4.6

```
> sort([-3,11,1,-6,-0,2,0,5,3,7]);
[-6, -3, 0, 0, 1, 2, 3, 5, 7, 11]

> sort(x^(-2)+x*y^2+x^2*y+y^(-1)+x*y+x^(-1)*y^(-1)+x+y);
x^2 y + x y^2 + x y + x + y + 1/y + 1/(x y) + 1/x^2
```

Більшість цих команд також доступна з контекстного меню.

Функції для роботи з частинами виразів

Іноді виникає необхідність працювати не з цілим виразом, а лише з його частиною. В Maple для існують спеціальні команди, які дозволяють працювати з окремими компонентами виразу.

Команди **lhs()** – для добування лівої частини виразу;
rhs() – для добування правої частини виразу:

малюнок 1.4.7

```

> z:=cos(x)^2-sin(x)^2=cos(2*x);
> lhs(z);
> rhs(z);

```

$$z := \cos(x)^2 - \sin(x)^2 = \cos(2x)$$

$$\cos(x)^2 - \sin(x)^2$$

$$\cos(2x)$$

numer() – добути числовик дробу;

denom() – добути знаменник дробу:

nops() – визначити кількість операндів. Наприклад, визначити кількість доданків або визначити кількість елементів списку.

op() – добути операнд. Для цього як перший аргумент даної функції зазначається порядковий номер потрібного елемента.

indets() – визначити невизначені елементи виразу, тобто ті, які не мають присвоєного значення.

Обчислення значень виразів

Для обчислення значень виразів найчастіше вживається функція *subs(s1, s2, ..., sn, expr)*, де *s1, s2, ..., sn* – вирази типу *x = y*, які підставляються у вираз *expr*. Також для обчислення значення виразу використовується функція *eval(expr)* (знаходження точного значення):

Малюнок 1.4.8

```

> eval(cos(x)-sin(x), x=Pi/3);
> f:=x^2-3*x^5+11*x-7*x^3;
> subs(x=2, f);

```

$$\frac{1}{2} - \frac{1}{2}\sqrt{3}$$

$$f := x^2 - 3x^5 + 11x - 7x^3$$

$$-126$$

Щоб обчислити значення виразу *expr* у вигляді числа з плаваючою точкою (тобто наближене числове значення) користуються функцією *evalf(expr)*.

Функція *evalc(expr)* дозволяє обчислити вираз *expr* у вигляді комплексного числа, а *evalb(expr)* – обчислити значення логічного виразу *expr* (із можливим результатом true, false, FAIL).

Точні та наближені обчислення

Система Maple проводить точні обчислення з такими величинами, як цілі числа, раціональні, математичні константи, матриці. Їм можна присвоїти певні імена, наприклад *x* або *y*, які поряд із математичними функціями типу *cos(x)* розглядатимуться системою як символічні об'єкти. Результатом символічних обчислень також буде точний результат, який за необхідності можна потім окремо перетворити на наближений.

Наприклад, результатом додавання простих дробів є також простий дріб:

малюнок 1.4.9

```

> 4/7 + 11/17;
> |

```

$$\frac{145}{119}$$

Або при обчисленні значень, наприклад тригонометричних функцій, *tan(π/3)* є точним числом, а *1,734050808* його округлене значення. Отже, Maple надає перевагу точному результату:

малюнок 1.4.10

```
> tan(Pi/3);  
> evalf(%);  
> {
```

$$\sqrt{3}$$
$$1.732050808$$

Якщо в обчисленні є числа із плаваючою точкою, таке обчислення є наближенням. Система реагує на десяткову точку числовими наближеннями в обчисленнях. Результат також буде у вигляді числа з десятковою точкою.