МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ-2»

КОНСПЕКТ ЛЕКЦІЙ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 171 «Електроніка»,
освітньою програмою «Електронні компоненти і системи»*

Київ
КПІ ім. Ігоря Сікорського
2019

«Інформаційні технології-2»: конспект лекцій [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка», освітньої програми «Електронні системи» / КПІ ім. Ігоря Сікорського ; уклад.: К. С. Клен. – Електронні текстові данні (1 файл: 6,045 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2019. – 154 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 9 від 30.05.2019 р.)
за поданням Вченої ради факультету електроніки (протокол № 05/2019 від 23.05.2019 р.)

Електронне мережне навчальне видання

«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ-2»
КОНСПЕКТ ЛЕКЦІЙ

Укладачі: *Клен Катерина Сергіївна,* канд. техн. наук.

Відповідальний редактор *Ямненко Ю. С.*, завідувач кафедри промислової електроніки, д-р техн. наук, проф.

Рецензент: *Попов А. О.,* доцент кафедри електронної інженерії, канд. техн. наук, доц.

При підготовці бакалаврів за спеціальністю 171 Електроніка, освітньою програмою «Електронні системи» однією з важливих дисциплін, що є компонентом циклу професійної підготовки, є дисципліна «Інформаційні технології-2».

Метою розробки конспекту лекцій є формування у студентів здатностей до: набуття практичних навичок роботи у пакеті символьних обчислень Mathematica для вирішення математичних задач; побудови двовимірних і тривимірних графіків; написання програм на мові Mathematica; використання елементів інтерфейсу користувача. В результаті вивчення матеріалу конспекту лекцій студент повинен отримати знання - основних констант, операторів та функцій програми Mathematica, правил складання програм, призначень пакетів розширення програми Mathematica; уміння робити інженерні розрахунки, користуватись засобами візуалізації, інтерфейсу користувача і побудови графіків, складати програми обчислень.

Конспект лекцій містить теоретичні відомості до 16 лекцій та список рекомендованої ліератури.

ЗМІСТ

Вступ

Програмний пакет Mathematica є системою комп'ютерної алгебри, призначеною для обробки математичних формул. Серед інших систем комп'ютерної алгебри слід відзначити програми Maxima, Maple, MuPAD, Reduce. Основна задача названих програм є автоматизація алгебраїчних перетворень і обробки символьних виразів. Перша версія пакету Mathematica розроблена у 1988 році, версія 11.0.1 пакету вийшла у вересні 2016 року.

У конспекті лекцій розглянуто основні можливості системи Mathematica з обробки масивів даних, побудови графіків, розв'язку лінійних і трансцендетних рівнянь, розв'язку задач математичного аналізу, апроксимації даних, основ програмування і створення інтерфейсу користувача.

Лекція № 1. Вступ до системи Mathematica

Основні можливості системи Mathematica:

- інтегрування і диференціювання функцій, вирішення систем поліноміальних і тригонометричних рівнянь та нерівностей, рекурентних рівнянь, рішення диференційних рівнянь і рівнянь в частинних похідних, ряди Тейлора, спрощення виразів, розрахунок меж, знаходження скінченних і нескінченних сум і добутків, а також ряд інших задач у символьному вигляді;
- поліноміальна інтерполяція функцій, розрахунок елементарних і спеціальних функцій із заданим ступенем точності, розрахунок перетворень Лапласа, Фур'є, z-перетворення;
- розв'язок задач лінійної алгебри, теорії чисел та інших розділів математики;
- представлення даних у графічному форматі (побудова графіків, параметричних кривих і поверхонь, побудова геометричних фігур, імпорт і експорт графічних даних у растрових і векторних форматах);
- підтримка розподілених обчислень (пакет Parallel Computing Toolkit);
- можливість написання програм на вбудованій процедурно-функціональній мові програмування.

Крім того, система має ряд стандартних пакетів розширення (Add-Ons):

- Algebra – робота з поліномами, алгебраїчними нерівностями, Гамільтоновою алгеброю і т.д.;
- Calculus – символьні обчислення похідних, інтегралів меж функцій, пряме і зворотне перетворення Фур'є і Лапласа, вирішення систем нелінійних рівнянь, реалізація інваріантних методів, вирішення диференційних рівнянь у частинних похідних, знаходження повних інтегралів і диференційних інваріантів нелінійних рівнянь, апроксимація Паде, обчислення еліптичних інтегралів і робота з векторами;
- DiscreteMath – обчислення з області дискретної математики, комбінаторики, обчислювальної геометрії і теорії графів, вирішення рекурентних і різницевих рівнянь, операції з цілими числами і т.д.;
- Geometry – функції для виконання геометричних розрахунків, створення правильних прямокутників і багатогранників, обертання геометричних фігур на площині і в просторі;
- Graphics – побудова графіків спеціального виду, геометричних фігур і поверхонь, графіків параметрично і неявно заданих функцій, опис функція комплексної змінної , відображення ортогональних проекцій тривимірних фігур, імітація тіней, засоби оформлення графіків;
- LinearAlgebra – вирішення задач лінійної алгебри, додаткові векторні і матричні операції, формування ортогональних векторних базисів;
- Miscellaneuos – визначення одиниць вимірювання фізичних величин, даних про хімічні елементи, фізичні константи, географічні дані та ін.;
- NumberTheory – функції теорії чисел;
- NumericalMath – реалізація чисельних методів, апроксимація даних і аналітичних функцій поліномами, сплайнами, тригонометричними рядами,

чисельне інтегрування і диференціювання, вирішення диференційних рівнянь, обчислення коренів нелінійних рівнянь;

- Statistics – статистичні функції для неперервних і дискретних функцій розподілу, реалізація лінійної і нелінійної регресії, обчислення параметрів функцій розподілу;

- Utilities – додаткові утиліти для роботи з бінарними файлами і пам'яттю комп'ютера, підтримки мов, роботи з системами класу AutoCAD і т.д.

Основи роботи з системою

Інтерфейс системи Mathematica складається з головного меню і області введення даних – блокноту, рис. 1.

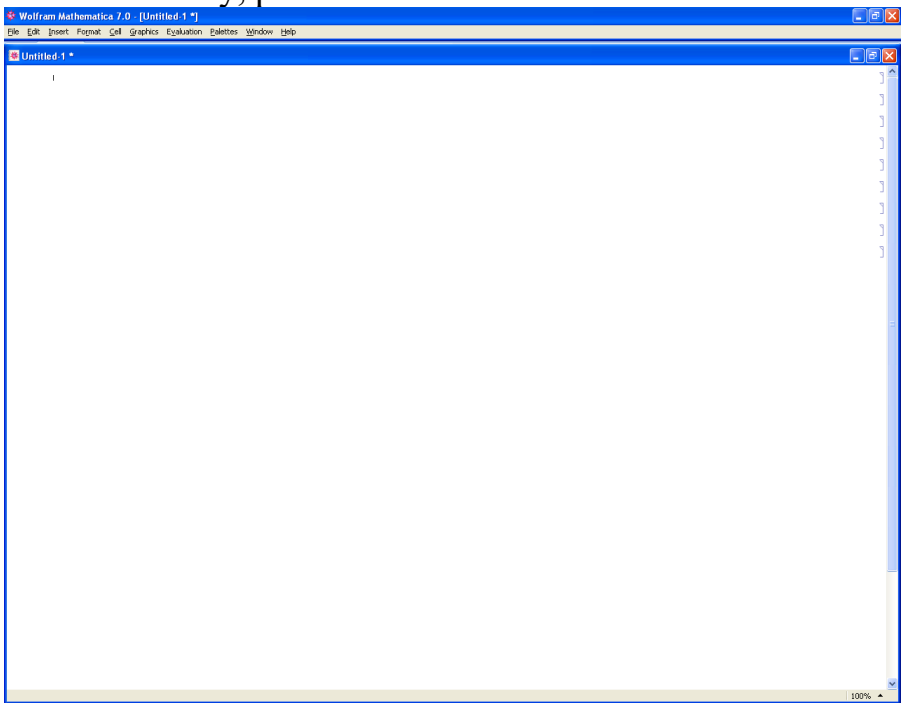


Рис. 1. Інтерфейс системи Mathematica

Робочим документом системи Mathematica є блокнот. Блокнот складається з комірок. Найпростіший спосіб роботи з системою Mathematica – інтерактивний. Система призначає кожному введеному і виведеному виразу номер комірки – In[n] і Out[n] відповідно. Для використання останнього виразу достатньо ввести «%», для посилання на результат, записаний в комірці під номером n – «%n». Для виконання введеної команди в кінці рядка необхідно натиснути клавіші Shift+Enter. Результатом виконання математичних операцій над виразами, вказаними у лістингу 1, будуть вирази, записані у вихідних комірках Out[1]-Out[6], лістинг 2.

```
5^10
4+5
%+a
%2
4+5
9
```

Лістинг 1

```
Out[1]= 9765625
Out[2]= 9
Out[3]= 9+a
Out[4]= 9
Out[5]= 9
Out[6]= 9
```

Лістинг 2

При введенні даних необхідно дотримуватись наступних правил:

- квадратні дужки [] використовуються для позначення аргументів функцій, навіть якщо функція не має аргументів, наприклад Random[];

- фігурні дужки використовуються для створення списків, векторів і матриць;

- круглі дужки використовуються в математичних виразах для задавання пріоритету виконання математичних операцій;

- подвійні квадратні дужки використовуються для індексації елементу у списку. [[i]] – повертає і-й елемент списку, [[i,j]] – повертає j-й елемент в рядку і.

Оперативна довідка

В системі Mathematica наявна коротка довідка по використовуваним в її середовищі об'єктам. Для виводу всього переліку об'єктів необхідно ввести команду:

?*

Можна також отримати довідку по всім об'єктам, назва яких починається на певну літеру:

?U*

Для отримання короткої довідки по конкретному об'єкту необхідно ввести команду ?Name, наприклад:

?Abs

Abs[z] gives the absolute

value of the real or complex number z.

Головне меню. Панель головного меню має всього два рядки:

• з назвами системи і завантаженого файлу;

• позиціями головного меню.

Праворуч і знизу вікна редагування містяться лінійки прокрутки з характерними повзунками, що можуть бути керованими мишею. У самому низу на початку лінійки прокрутки є так звана статусний рядок з інформацією про поточний режим роботи (Status bar).

Головне меню системи (рис. 1) містить такі вкладки:

• File – робота з файлами: створення нового файлу, вибір існуючого файлу з каталогу, закриття файлу, збереження поточного файлу, збереження файлу зі зміною імені, друк документа і вихід в Windows;

• Edit – виконання основних операцій редагування (відміна операції, копіювання виділених частин документа в буфер з їх подальшим видаленням і без нього, перенесення виділених частин, їх стирання);

• Incert – завдання елементів введення (графіків, матриць, гіперпосилань, додавання у робочий ноутбук елементів файлів, вибір кольору робочої комірки та нумерація комірок);

• Format – установка формату для документів;

• Cell – робота з функціональними комірками (об'єднання і роз'єднання комірок, установка статусу комірки, відкриття і закриття);

• Evaluation – управління ядром системи та його конфігурацією;

• Palettes – робота з палітрами математичних операторів та функцій, засоби введення математичних символів та їх опції;

• Window – операції з вікнами і їх розташуванням;

• Help – управління засобами довідкової системи.

Кожен елемент меню, при активації, відриває випадне підменю, що містить пов'язані з ним команди. Назви виконуваних команд виділяються чітким, а ті, які не виконуються в даний час - характерним сірим розпливчастим шрифтом.

Елементи інтерфейсу, зокрема вікно редагування, можна переміщати за допомогою миші та розтягувати в різні боки. Курсор миші зазвичай має вигляд стрілки, проте змінюється при становленні на окремі деталі елементів інтерфейсу. Наприклад, при установці на вертикальну межу вікна він набуває вигляду двосторонніх стрілок ↔, розташованих по горизонталі. Вони вказують на можливість переміщення цієї лінії по горизонталі. Аналогічно можна розтягувати або стискати вікно переміщенням по вертикалі або діагоналі.

На початку титульних рядків головного меню і вікна редагування є кнопка з логотипом системи, що відкриває підменю з наступними командами:

- Відновити – відновити розміри елемента інтерфейсу;
- Перемістити – перемістити елемент інтерфейсу;
- Розмір – задати розміри елемента інтерфейсу;
- Згорнути – згорнути елемент в бирку в панелі завдань Windows;
- Розгорнути – розгорнути елемент інтерфейсу;
- Закрити – закрити елемент інтерфейсу.

Це підменю створюється засобами операційної системи Windows. Також, в кінці цих рядків є характерні кнопки, що повторюють три останні команди. Вони служать для управління вікнами відповідних елементів інтерфейсу.

Лекція № 2. Wolfram Mathematica і дані оточуючого середовища

Мова Wolfram Language дозволяє програмістам працювати на значно більш високому рівні, ніж будь-коли раніше, використовуючи вбудований обчислювальний інтелект, який спирається на величезну глибину алгоритмів і реальних знань, ретельно інтегрованих протягом трьох десятиліть. Будучи ефективним для створення програм розміром від крихітної до дуже великий, які природним чином допускають введення в широке використання як локально, так і хмарі, мова Wolfram Language, грунтуючись на чітких принципах-й елегантною уніфікованої символічної структуре- створює те, що стає найпродуктивнішим мовою програмування в світі, а також першим справжнім обчислювальним мовою спілкування між людьми і системами штучного інтелекту.

Арифметичні операції

Операція	Арифметичний оператор	Скорочена форма	Функція
Додавання	+	+=	Plus[x1, x2,…xn]
Віднімання	-	-=	-
Множення	*	*=	Times[x1, x2,…xn]
Ділення	/	/=	Divide[x1, x2]
Піднесення до степеня	^	-	-

Типи даних

Тип чисел	Позначення	Приклад
Цілочисельні	Integer	35, -2
Раціональні	Rational	35/46,
Дійсні	Real	2.18, 3.6*10^-5
Комплексні	Complex	2+3*I

Іменовані константи

E – число e;

Pi – число π;

I – уявна одиниця $\sqrt{-1}$;

Infinity – уявна нескінченість +∞, при від'ємній нескінченості ставиться знак -;

Degree – число радіан в градусі π/180;

EulerGamma – постійна Ейлера 0,577216;

GoldenRatio – константа золотого перерізу $\frac{1+\sqrt{5}}{2}$;

Catalan – константа Каталана 0.915966.

Для отримання чисельного значення константи необхідно застосувати функцію N[].

N[E]

Out[1]= 2.71828

Якщо необхідно вивести значення числа із заданою кількістю знаків, використовується функція N[k,n], де k – число, n – кількість знаків.

N[5/88,10]

Out[1]= 0.05681818182

Вбудовані математичні функції

Функції визначення дільників, найменшого кратного цілих чисел

Divisors[n] – повертає цілочисельні дільники числа n;

ExtendedGCD[n,m] – повертає найбілший спільний дільник чисел n і m;

GCD[n1,n2…] – повертає найбільший спільний дільник чисел n1, n2…;

LCM[n1,n2…] – повертає найменше спільне кратне чисел n1, n2…

MOD[x,y] – повертає залишок від ділення числа x на y.

Функції округлення дійсних чисел

Round[x] – округлення до найближчого цілого;

Floor[x] – повертає найбільше число, що не перевищує x;

Ceiling[x] – повертає найменше число, що більше або дорівнює x;

Quotient[x,y] – повертає округлене ціле число x/y, що не перевищує значення x/y.

Обчислення факторіалів

Factorial[n] – повертає значення n!;

Factorial2[n] – повертає значення n!!=n*(n-2)*(n-4)...

Отримання простих чисел

Prime[n] – повертає n-е просте число;

PrimePi[n] – повертає кількість простих чисел, що не перевищують n.

Елементарні функції

Степеневі і логарифмічні функції

$\sqrt{z} \rightarrow$ Sqrt[z];

$z^a \rightarrow$ Power[z,a];

$e^z \rightarrow$ Exp[z];

$\ln(z) \rightarrow$ log[z];

$\log_a(z) \rightarrow$ log[a,z];

Тригонометричні функції

$\sin(z) \rightarrow$ Sin[z];

$\cos(z) \rightarrow$ Cos[z];

$tg(z) \rightarrow$ Tan[z];

$ctg(z) \rightarrow$ Cot[z];

$\csc(z) \rightarrow$ Csc[z];

$sec(z) \rightarrow$ Sec[z].

Обернені тригонометричні функції

$arc\sin(z) \rightarrow$ ArcSin[z];

$arc\cos(z) \rightarrow$ ArcCos[z];

$arcctg(z) \rightarrow$ ArcCot[z];

$arc\csc(z) \rightarrow$ ArcCsc[z];

$arcsec(z) \rightarrow$ ArcSec[z].

Гіперболічні функції

$\sinh(z) \rightarrow$ Sinh[z];

$\cosh(z) \rightarrow$ Cosh[z];

$tgh(z) \rightarrow$ Tanh[z];

$ctgh(z) \rightarrow$ Coth[z];

$\csc h(z) \rightarrow$ Csch[z];

$\sec h(z) \rightarrow$ Sech[z].

Обернені гіперболічні функції

$arc\sinh(z) \rightarrow$ ArcSinh[z];

$arc\cosh(z) \rightarrow$ ArcCosh[z];

$arcctgh(z) \rightarrow$ ArcCoth[z];

$arc\csc h(z) \rightarrow$ ArcCsch[z];

$arc\sec h(z) \rightarrow$ ArcSech[z].

Арифметичні операції з цілими і раціональними числами

Система Mathematica виконує обчислення з цілими і раціональними числами без похибок, що ілюструється в лістингу 2.

```
100!!
 3424322470251197624824643289520818597511 8
67505371919882791565446348800000000000
 1/2+2/3+1/6
  4
  -
  3
 2/17+3/7+5/22+1/121
  22513
  -----
  28798
```

Лістинг 3

Арифметичні операції з дійсними числами

Дійсні числа в системі Mathematica представляються в звичайній або нормальній формі. При представленні числа в звичайній формі ціла частина числа відділяється від дробовою крапкою: 1.35, 0.24. При цьому 0 цілих можна не писати і замість 0.25 використовувати запис - .25. Числа записані в такій формі запису називаються числа з фіксованою крапкою.

Крапка в кінці числа є ознакою, що число є дійсним. Наприклад число 131 є цілим, число 131. – дійсним, число 2/5 – є раціональним, а число 2./5 – дійсним.

При представленні числа в нормальній формі, число записується у вигляді мантиси з цілою і дробовою частиною і порядку у вигляді степеня

числа: 5.*10^-3, 3.335*10^-6. Замість знаку множення можна використовувати пробіл. Арифметичні операції над дійсними числами дають наближений результат. Система Mathematica оперує числами в діапазоні $2.22507^{-308}...1.79769\cdot10^{308}$. Для підвищення точності дійсні числа можна переводити в раціональні за допомогою функцій:

Rationalize[z] – дійсне число z перетворюється в раціональне;

Rationalize[z,dz] – дійсне число z перетворюється в раціональне з точністю dz;

Арифметичні операції з комплексними числами

Комплексне число представляється в такому виді:

$z = \mathrm{Re}(z) + I * \mathrm{Im}(z)$.

Функції виконання операцій над комплексними числами:

Abs[z] – обчислює модуль комплексного числа z;

Arg[z] – обчислює аргумент комплексного числа z;

Conjugate[z] – обчислює комплексноспряжене число до z;

Вирази їх перетворення і обчислення

Підстановки

Підстановки є математичним апаратом, призначеним для обчислення функцій при чисельно заданих значеннях аргументу. Вони дозволяють табулювати значення функцій. В системі Mathematica для цього використовується символ «/.»

```
f(x) /. x->a;
f(x,y,…) /. {x->a,y->b,…};
{f1(x),f2(x),…} /. x->a;
{f1(x,y,..),f2(x,y,…),…} /. {x->a,y->b,…};
f(x) /. x->{x0,x1,…}.
```

Вирази підстановок мають наступний зміст:

f(x) /. x->a – здійснює підстановку в вираз f(x) значення x=a.

f(x,y,…) /. {x->a,y->b,…} – здійснює підстановку в вираз f(x,y,…) значення x=a, y=b,…

{f1(x),f2(x),…} /. x->a – здійснює підстановку в вирази f1(x), f2(x),… значення x=a.

{f1(x,y,..),f2(x,y,…),…} /. {x->a,y->b,…} – здійснює підстановку в вирази f1(x,y,…), f2(x,y,…),… значення x=a, y=b,…

f(x) /. x->{x0,x1,…} – табулювання функції f(x).

Приклад використання підстановок показано у лістингу 3.

```
z1=x E^x+Log[x]-1.;
z1/.x→1.
 1.71828
 z2=x^2+y^2+Sin[x]^2+Cos[x]^2+1.5;
z2/.{x→1,y→2}
 7.5
 z2/.y→2.
 5.5 + x^2 + Cos[x]^2 + Sin[x]^2
 z1/.x→a
 -1. + a ℯ^a + Log[a]
 z2/.{x→a,y→b}
 1.5 + a^2 + b^2 + Cos[a]^2 + Sin[a]^2
 z2/.{x→a,y→a}
 1.5 + 2 a^2 + Cos[a]^2 + Sin[a]^2
```

Лістинг 4

Перетворення виразів

Функцій перетворення виразів:

Simplify[f] – спрощує вираз f;

FullSimplify[f] - спрощує вираз f, який в своєму складі містить спеціальні функції;

Expand[f] – розкриває і розширює вираз f;

Collect[f,x] – представляє вираз f по степеням x;

TrigExpand[f] – перетворює тригонометричні вирази;

Factor[f] – розкладає вираз на множники.

Приклади використання функції Simplify показано в лістингу 4, FullSimplify – в лістингу 5.

```
Simplify[957828/3831312]
```

$$\frac{1}{4}$$

```
Simplify[(315 + 297 x - 62 x^2 - 42 x^3 + 3 x^4 + x^5)/(105 - 41 x - x^2 + x^3)]
```

$$3 + 4\,x + x^2$$

```
Simplify[(Sin[x]+Cos[x])^2-1]
```

$$\text{Sin}[2\ x]$$

```
Simplify[a^4 + a^3 x + a^2 x^2 + a x^3 + x^4 + a (-1 + a^4)/(-a + x)]
```

$$\frac{a - x^5}{a - x}$$

$$\frac{2 + a + b - c}{a + b - c} - \frac{b + a b + c + a c}{b + c} + a$$

$$a + \frac{2 + a + b - c}{a + b - c} - \frac{b + a b + c + a c}{b + c}$$

```
Simplify[%]
```

$$\frac{2}{a + b - c}$$

```
Simplify[7.59375 + 25.3125 x + 33.75 x^2 + 22.5 x^3 + 7.5` x^4 + x^5]
```

$$1.\ (1.5 + x)^5$$

```
Simplify[(2 x+a^Log[a,x]+3)/3]
```

$$1+x$$

Лістинг 5

```
Simplify[x *Sin[x]* Cos[y]+(x^2+1)/x*Sin[y] Cos[x]]
```

$$x \, \text{Cos}[y] \, \text{Sin}[x] + \frac{(1 + x^2) \, \text{Cos}[x] \, \text{Sin}[y]}{x}$$

```
FullSimplify[x*Sin[x]*Cos[y]+(x^2+1)/x*Sin[y]*Cos[x]]
```

$$\frac{\text{Cos}[x] \, \text{Sin}[y]}{x} + x \, \text{Sin}[x + y]$$

```
Simplify[(Sin[x]+Cos[x])^2]
```

$$(\text{Cos}[x] + \text{Sin}[x])^2$$

```
FullSimplify[(Sin[x]+Cos[x])^2]
1+Sin[2 x]
FullSimplify[(1-Cos[2 x])/2+Cos[x]^2+a]
1+a
```

Лістинг 6

Функція Expand

Модифікації функції Expand:

Expand[f] – розкриває дужки добутків і підносить вирази в цілий додатний степінь;

ExpandAll[f] – розкриває дужки добутків і підносить вирази в цілий додатний степінь у будь-якому місці виразу f;

ExpandNumerator[f] - розкриває дужки добутків і підносить вирази в цілий додатний степінь у чисельнику виразу f;

ExpandDenominator[f] - розкриває дужки добутків і підносить вирази в цілий додатний степінь у знаменнику виразу f;

PowerExpand[f] - розкриває дужки добутків і підносить вирази в цілий додатний степінь у вкладених виразах функції f;

ComplexExpand[f] - розкриває дужки добутків і підносить вирази в цілий додатний степінь, за умови що змінні виразу f є комплексними числами;

ComplexExpand[f,{x1,x2,…}] - розкриває дужки добутків і підносить вирази в цілий додатний степінь, за умови що змінні x1,x2,.. виразу f є комплексними числами;

FunctionExpand[f] – використовується для спрощення виразу f, якщо до його складу входять спеціальні функції.

В лістингу 6 показані приклади використання модифікацій функції Expand[].

```
Expand[(a+b) (a-b) (a+c)/(a+1)^3]
```

$$\frac{a^3}{(1+a)^3} - \frac{ab^2}{(1+a)^3} + \frac{a^2c}{(1+a)^3} - \frac{b^2c}{(1+a)^3}$$

```
ExpandAll[(a+b) (a-b) (a+c)/(a+1)^3]
```

$$\frac{a^3}{1+3a+3a^2+a^3} - \frac{ab^2}{1+3a+3a^2+a^3} + \frac{a^2c}{1+3a+3a^2+a^3} - \frac{b^2c}{1+3a+3a^2+a^3}$$

```
ExpandAll[Sqrt[x+2^3]/((a+b) (a-b)) (a-b)+(c+2 I)^2]
```

$$-4 + 4ic + c^2 + \frac{\sqrt{8+x}}{a+b}$$

```
ExpandNumerator[(x^2-1)^5/(x+1)^3]
```

$$\frac{-1+5x^2-10x^4+10x^6-5x^8+x^{10}}{(1+x)^3}$$

```
ExpandDenominator[(x^2-1)^5/(x+1)^3]
```

$$\frac{(-1+x^2)^5}{1+3x+3x^2+x^3}$$

Лістинг 7

Функція Collect

Модифікації функції Collect:

Collect[f,x] – представляє вираз f по степеням x;

Collect[f,{x1,x2,…}] – представляє вираз f по степеням змінних x1,x2...

В лістингу 7 показані приклади використання модифікацій функції Collect[].

```
Collect[(x+a)^3-(x-a)^2,x]
```

$$-a^2 + a^3 + (2a + 3a^2)\,x + (-1 + 3a)\,x^2 + x^3$$

```
Collect[(x+1)^5-EulerE[5,x]-(x+a)^3,x]
```

$$\frac{3}{2} - a^3 + (5 - 3a^2)\,x + \left(\frac{15}{2} - 3a\right)x^2 + 9x^3 + \frac{15x^4}{2}$$

```
Collect[(x-y)(x+y)^2 Sqrt[(x+y)^3],x]
```

$$x^3\sqrt{(x+y)^3} + x^2y\sqrt{(x+y)^3} - xy^2\sqrt{(x+y)^3} - y^3\sqrt{(x+y)^3}$$

```
Collect[(x+y)^2+(x-1)^3-(y-1)^3,{x, y}]
```

$$-2x^2 + x^3 - 3y + 4y^2 - y^3 + x\,(3 + 2y)$$

```
Collect[1/(x+y)+1/(x-y)+x-y,{x, y}]
```

$$x + \frac{1}{x-y} - y + \frac{1}{x+y}$$

Лістинг 8

Функція Factor

Модифікації функції Factor:

Factor[f] – розклад функції f на прості доданки;

FactorList[f] – повертає перелік множників і число степенів кожного з них;

FactorTerms[f] – виносить за дужки спільний множник виразу f;

FactorTermsList[f] – виділяє спільний множник виразу f, представляючи функцію f в іншому виді;

FactorInteger[f] – повертає прості множники цілого числа f.

В лістингу 8 показані приклади використання модифікацій функції Factor[].

```
Factor[18 x^3+45 x^2+ x-14]
 (-1+2 x) (2+3 x) (7+3 x)
Factor[x^3+x^2 (3+a-b)+x (3 a-3 b-a b)-3 a b]
 -(b-x) (3+x) (a+x)
 Factor[a^2+a-1]
 -1 + a + a^2
 Factor[6 x^3+x^2 (13 I-6)-6 x (1+13/6 I)+6]
 (-1+x) (3 i+2 x) (2 i+3 x)
Factor[Sin[4 x],Trig→True]
 4 Cos[x] (Cos[x]-Sin[x]) Sin[x] (Cos[x]+Sin[x])
 Factor[1/16 (Cos[5 x]+5 Cos[3 x]+10 Cos[x]), Trig→True]
 Cos[x]^5
FactorList[x^5+x^4-6 x^3-x^2-x+6]
 {{1, 1}, {-2 + x, 1}, {-1 + x, 1}, {3 + x, 1}, {1 + x + x^2, 1}}
FactorTerms[3 x^3-3 x^2-51 x-45 ]
 3 (-15 - 17 x - x^2 + x^3)
 FactorTermsList[3 x^3-3 x^2-51 x-45]
 {3, -15 - 17 x - x^2 + x^3}
FactorInteger[1235]
 {{5,1},{13,1},{19,1}}
```

Лістинг 9

Функції перетворення тригонометричних виразів

TrigReduce[f] – спрощує вираз f, який містить тригонометричні функції;

TrigExpand[f] – здійснює розширення виразів тригонометричних функцій;

TrigFactor[f] – розкладає на множники тригонометричні функції;

TrigToExp[f] – перетворює тригонометричний вираз в експоненційну форму;

ExpToTrig [f] – перетворює експоненційний вираз в тригонометричну форму.

В лістингу 9 показані приклади використання функцій перетворення тригонометричних виразів.

```
TrigReduce[(Sin[x]+Cos[x])^2]
```

$$1+\text{Sin}[2\ x]$$

```
TrigReduce[2 Sin[1/2 (x+y)] Sin[1/2 (y-x)]]
```

$$\text{Cos}[x]-\text{Cos}[y]$$

```
TrigExpand[Sin[4 x]]
```

$$4\,\text{Cos}[x]^3\,\text{Sin}[x]-4\,\text{Cos}[x]\,\text{Sin}[x]^3$$

```
TrigExpand[Cosh[Cos[x y]]]
```

$$\text{Cosh}\left[\frac{1}{2}\,\text{Cos}[x\,y]-\frac{1}{2}\,i\,\text{Sin}[x\,y]\right]\text{Cosh}\left[\frac{1}{2}\,\text{Cos}[x\,y]+\frac{1}{2}\,i\,\text{Sin}[x\,y]\right]+$$
$$\text{Sinh}\left[\frac{1}{2}\,\text{Cos}[x\,y]-\frac{1}{2}\,i\,\text{Sin}[x\,y]\right]\text{Sinh}\left[\frac{1}{2}\,\text{Cos}[x\,y]+\frac{1}{2}\,i\,\text{Sin}[x\,y]\right]$$

```
TrigFactor[Cos[x]-Cos[y]]
```

$$-2\,\text{Sin}\left[\frac{x}{2}-\frac{y}{2}\right]\text{Sin}\left[\frac{x}{2}+\frac{y}{2}\right]$$

```
TrigToExp[Cos[x]+sin[x]]
```

$$\frac{e^{-ix}}{2}+\frac{e^{ix}}{2}+\text{sin}[x]$$

```
TrigToExp[1/I Sin[I x]]
```

$$-\frac{e^{-x}}{2}+\frac{e^{x}}{2}$$

```
ExpToTrig[E^x+E^(-x)]
```

$$2\ \text{Cosh}[x]$$

Лістинг 10

Лекція № 3. Таблиці, стилі та графічні об'єкти

Представлення векторів і матриць в табличному вигляді можливо за допомогою функцій TableForm і MatrixForm, що мають вигляд:

TableForm[f]

Out[n] // MatrixForm

де f - ім'я вектора або матриці;

n - номер рядка, в якій знаходиться вектор або матриця;

% - застосовується у разі, якщо функція % / / MatrixForm подання вектора або матриці в табличному вигляді розташовується слідом за вектором (матрицею).

Вектор або матрицю можна також створити за допомогою функції List:

List [a, b, c,...] — створює вектор {a,Ь,c,...};

List [{a, Ь, c,..}, {d, e, f, ..}, {d, h, k, ..}] - створює матрицю {{a, b, c,..}, {d, e, f, ..}, {g, У, k, ..}}.

В лістингу 2 в табличній формі наведені вектор f3 та матриці f4, f5, з лістингу 1, утворені за допомогою функцій TableForm і MatrixForm.

TableForm[f3]

2 + 3 i
1- 2 i
5
3 +7i
7

TableForm[f4]

1	2	3
4	7	0
-5	1	8

TableForm[f5]

Sin[X]	e^{-x}	$\frac{-1+x}{1+x}$	Log[x]	5
1 + 2 i	3	5	Tan[1+x]	-8
4	a	*B*	Cos[x]	2-i

Out [29]//MatrixForm

$$\begin{pmatrix} rSin[x] & e^{-x} & \frac{-1+x}{1+x} & Log[x] & 5 \\ 1+2\,i & 3 & 5 & Tan[\,1+x] & -8 \\ 4 & a & B & Cos[x] & 2-i \end{pmatrix}$$

Лістинг 5

Генерація векторів і матриць за допомогою функції Range

Функція Range застосовується для створення числових списків і має такі модифікації:

Range [n_{max}] - генерує вектор числових елементів виду {1, 2, ..., n_{max} };

Range [n_{mln}, n_{max}] - генерує вектор з числових елементів виду {n_{mln}, n_{max}};

Range [n_{mln}, n_{max}, dn] - генерує вектор числових елементів від n_{mln} до n_{max} з кроком dn.

Приклади реалізації функції наведено в лістингу 6.

Range [7]

{1,2,3,4,5,6,7}
Range[4,10]
{4,5,6,7,8,9,10}

Range[3,8,0.5]
{3,3.5,4.,4,5,5.,5.5,6.,6.5,7.,7.5,8.}

Лістинг 6

Генерація векторів і матриць за допомогою функцій Table

Для створення векторів і матриць можна використовувати функцію Table, що має вигляд:

Table [f, { n_{max} }] - створює n_{max} копій виразу f;

Table [f,{1, n_{max}}] - створює список функцій f при в діапазоні від 1 до n_{max};

Table [f, {n, n_{min}, n_{max}}] — список функцій f в діапазоні від n_{mln} до n_{max};

Table [f, {n, n_{min}, n_{max}, dn}]— створює список функцій f в діапазоні від n_{mln} до n_{max} з кроком dn.

Приклад використання функції Table показаний у лістингу 7.

Table [Log [x] ,{5}]
{Log[x] , Log[x] , Log[x] , Log[x] , Log[x] }
Table[Log[x] ,{x,5}]
{0,Log[2],Log[3],Log[4] ,Log[5]}
Table[Log[x] ,{x,5,10}]
{Log[5] ,Log[6] ,Log[7] ,Log[8] ,Log[9] ,Log[10] }
Table[Log[x] ,{x,l ,3 ,0 .5}]
{0,0.405465,0. 693 147,0.9162 91,1.09861}
Table [i+j ,{1,2 ,4} ,{j ,2,4}]
{{4,5, 6} ,{5, 6,7}, {6,7,8}}

Лістинг 7

Виділення та виведення елементів вектора і матриці

В системі Mathematica реалізовано такі методи виділення елементів векторів і матриць:

- використання подвійних квадратних дужок;
- використання функції Part;
- використання функції Select.

Використання подвійних квадратних дужок

У цьому випадку вираз, що виокремлює елементи вектора або матриці, представляється у вигляді:

f [[n]] f [[n_1, n_2, ...]],

де f - ім'я вектора або матриці;

n - елемент що виділяється;

n_i - і-й елемент із сукупності елементів що виділяються.

Приклади виділення елементів показано у лістингу 8.

f1={2 ,1,4,3,5}
{2,1,4,3,5}
f2={ {1,2,3}, {3,5,7}, {2,4,6}}
{{ 1,2,3} ,{3,5,7}, {2,4,6}}
f3={a,1,b,2,3,c}

{a,1,b,2,3,c}
f4={2,a,Sin[x+y^2] ,b,1}
{2, a, Sin[x + y^2] , b, 1}
f1[[4]]
3
f2[[2,2]]
5
f1 [[{2,3}]]
{1,4}
{f2[[1,2]] ,f2[[3,1]]}
{2,2}

Лістинг 8

Виведення елементів векторів і матриць здійснюється за допомогою функцій MatrixForm і TableForm.

Приклади використання цих форм виведення показана у лістингу 10. Використання опцій TableAlignments і TableSpacing, що дозволяють розташувати вектор і матрицю на екрані в бажаному вигляді показано у лістингу 11.

F={{a,l,4},{b,2,5} ,{c,3,6}}
{{a, 1,4} ,{b,2,5}, {c,3,5}}
MatrixForm[F]

$$\begin{pmatrix} a & 1 & 4 \\ b & 2 & 5 \\ c & 3 & 6 \end{pmatrix}$$

TableForm[F]
a 1 4
b 2 5
c 3 6

Лістинг 10

s={ 5,73426813438765,34}
{5,73426813438755,34}
TableForm[s, TableAlignments->Left]
5
73426S13438765
34
TableForm[s, TableAlignments->Center]
5
73426813438765
34
s1={{1,3,4},{5,1,1} ,{3,2,1}}
{{1,3,4},{5,1,7},{3,2,1}}
TableForm[s1]
1 3 4
5 1 7
3 2 1
TableForm[s1, TableSpacing->{ 1,1}]
1 3 4
5 1 7
3 2 1
TableForm[s1, TableSpacing->{ 5,2}]

```
1     3     4

5     1     7

3     2     1
```

```
TableForm[s1, TableSpacing->{ 2,5}]
```

```
1           3           4
5           1           7
3           2           1
```

Лістинг 11

Лекція № 4. Робота з векторами і матрицями

Вектори і матриці в системі Mathematica є списками. Списком називається сукупність даних обмежена фігурними дужками. Вектор є одновимірним, матриця – двовимірним списком. Елементами векторів і матриць, можуть бути дійсні і уявні числа, функції, математичні вирази. В лістингу 1 наведені вектори і матриці з різних елементів.

f1= {1,2,3,4,5}
{1,2,3,4,5}
f2={a,b,c, d,e}
{a,b,c,d,e}
f3={2+3 I,1-2 1,5,3+7 1,7}
{2+3 i, 1-2 i,5,3+7 i, 7}
f4={{1,2,3},{4,7,0},{-5,1,8}}
{{ 1,2,3} ,{4,7, 0}, {-5, 1,3} }
f5={{Sin[x] ,E^-x, (x-1)/(x+1) ,Log[x] ,5} , {1+2 i,3,5,Tan[x+1],-8},{4,a,b,Cos[x] ,2-1}}
{{sin[x], e^{-x}, (-1+x)/(1+x),Log[x], 5}, {1 + 2I, 3, 5, Tan[1+x] , -8}, {4, a, b, Cos[x], 2-i}}

Лістинг 1

Створення векторів і матриць здійснюється за допомогою наступних функцій.

Array [f, n]- створює вектор з n елементів f [1], f [2],…, f [n];

Array [f, n_1, n_2]- створює вектор з n_1 елементів, починаючи з f [n_2], при цьому n_2, може бути числом, функцією, виразом;

Array [f, { n_1, n_2}] - створює матрицю n_1 x n_2, з елементів функції f (n_1, n_2);

Array [f, n_1, n_2, h] - створює вектор з n_1 елементів, починаючи з f (n_2), з рівнем масиву h.

Приклади створення векторами за допомогою функції Array наведено у лістингу 2. Зверніть увагу на те, що ці функції дозволяють здійснювати підсумовування і множення елементів вектора за допомогою функцій Plus і Times.

Array [Tan,5]
{Tan[l] , Tan[2] ,Tan[3] ,Tan[4], Tan[5] }
Array [Sin, 5,3]
{Sin[3] , Sin[4] ,Sin[S] ,Sln[6] , Sin[7] }
Array[Sin,5,Sqrt[3]]
{Sin [$\sqrt{3}$] , Sin[1+ $\sqrt{3}$] , Sin [2 + $\sqrt{3}$], Sin[3 + $\sqrt{3}$], Sin [4 +$\sqrt{3}$]}
N[%]
{0.98702 7,0.398189,-0.556742,-0.999807,-0.523654}
Array[Sin,5,Sqrt[3] ,Plus]
Sin[$\sqrt{3}$] + Sin[1 + $\sqrt{3}$] + Sin[2 + $\sqrt{3}$] + Sin[3 + $\sqrt{3}$] + Sin [4 +$\sqrt{3}$]
N[%]
-0.694987
Array[Sin,5,Sqrt[3] ,Times]
Sin[$\sqrt{3}$] Sin[1+ $\sqrt{3}$] Sin[2 + $\sqrt{3}$] Sin[3 + $\sqrt{3}$] Sin[4+ $\sqrt{3}$]
N[%]
-0.11456
Array [Tan, {3,4}]
{{Tan[1,1] ,Tan[1,2] ,Tan[1,3] ,Tan[1,4] }, {Tan[2,1],Tan[2,2] ,Tan[2,3] ,Tan[2,4] } ,{Tan[3, 1] ,Tan[3,2],Tan[3,3] , Tan[3,4] }}
Array [Exp, 5,3,2]
2 [e^3 , e^4, e^5, e^6, e^7]

N[%]
2 [e^3 , e^4, e^5, e^6, e^7]
Array[Exp, 5, a, h]
h[e^a , e^{1+a}, e^{2+a}, e^{3+a}, e^{4+a}]

Лістинг 2

Виявлення структури вектора або матриці

Для виявлення структури вектора або матриці служать наступні функції:

vectorQ [V] - перевіряє, чи є V вектором, і видає True, якщо так, і False, якщо ні;

MatrixQ [M] - перевіряє, чи є м матрицею, і видає True, якщо так, і False, якщо ні;

Length [V] - повертає число елементів вектора V;

Length [M] - повертає число рядків матриці M;

MemberQ [V, n] - перевіряє, чи є у векторі V елемент n; якщо так, то видає True, якщо ні - False;

FreeQ [V, n] - перевіряє, чи має вектор V елемент n; якщо містить, то видає False, якщо ні - True;

FreeQ [M, n] - перевіряє, чи має матриця M елемент n; якщо містить, то видає False, якщо ні - True;

Dimensions [V] - повертає число елементів вектора V;

Dimensions [M] - повертає розмір матриці (кількість рядків і число стовпців);

Position [V, n] - повертає номери позицій елемента n вектора V;

Count [V, n] - повертає число елементів у векторі V, що мають значення n;

TensorRank [V] - видає ранг вектора V, якщо V є тензором;

TensorRank [M] - видає ранг матриці M, якщо м є тензором.

Перетворення і створення векторів і матриць

Система Mathematica має багаті можливості перетворення і створення нових векторів і матриць (і списків будь-якого рівня). Для цієї мети можуть використовуватись такі функції:

Drop [V, n] - видаляє перші n елементів вектора V;

Drop [V,-n]- видаляє останні n елементів вектора V;

Drop [V, {n}] – видаляє n-й елемент вектора V;

Drop [V, {m, n}]- видаляє елементи від m-го до n-го вектора V;

Last [f] - повертає останній елемент вектора (матриці) f;

Rest [V] - повертає вектор V з відкинутим першим елементом;

Take [V, n]- повертає вектор V з першими n елементами;

Take [V,-n]- повертає вектор V з останніми n елементами;

Take [V, {m, n}] - повертає вектор V з номерами елементів від m до n;

Append [V, a]- додає елемент a в кінець вектора V;

Prepend [V,-a]- додає елемент a в початок вектора V;

Insert [V, a, n] - вставляє елемент a в позицію n з відліком з початку вектора V;

Insert [V, a,-n]- вставляє елемент a в позицію n з відліком з кінця вектора V;

Delete [V, n]- видаляє елемент n з вектора V;

{Delete [f, n_1], Delete [f, n_2], Delete [f, n_3], ...} - видаляє з вектора (матриці) f елементи n_i та утворює новий вектор (матрицю).

Приклади використання функцій наведено у лістингу 3.

V={1,3,4,5,7,2,4}
{1,3,4,5,7,2,4}
M= {{1,3,5},{2,4,6},{1,2,3}}
{{ 1,3,5} ,{2,4, 6}, {1,2,3}}
{Drop[V,3] ,Drop[V,-4] ,Drop[V,{5}] ,Drop [V,{2,5}] }
{{5,7,2,4} ,{ 1,3,4}, {1,3,4,5,2,4}, {1,2,4} }
Delete [V,6]
{1,3,4,5,7,4}
{Delete[V,1] ,Delete[V,-3] ,Delete[V,7]}
{{3,4,5,7,2,4} ,{1,3,4,5,2,4}, {1,3,4,5,7, 2} }
Delete |M,{2,3}]
{{ 1,3,5} ,{2,4} ,{1,2,3}}
Delete |M,{3,1}]
{{ 1,3,5} ,{2,4, 6}, {2,3}}
Last [V]
4
Last [M]
{1,2,3}
Rest [V]
{3,4,5,7,2,4}
{Take[V,4] ,Take[V,-4] ,Take[V, {3 ,6}] }
{{ 1, 3, 4,5} ,{5,7,2,4}, {4,5,7,2}}
Append [V, Sin [x]]
{l,3,4,5,7,2,4,Sin[x] }
Prepend[V, Sin[x]]
{Sin[x] , 1,3,4,5,7,2 ,4}
Insert [V,a,5]
{1,3,4,5,a,7,2,4}
Insert [V,{3,a,Sin[x] } ,5]
{1,3 ,4,5, { 3, a. Sin [x] } , 7,2 ,4}

Лістинг 3

Утворення нових векторів і матриць також можливе шляхом зміни розташування вектора або матриці, для чого використовуються наступні функції.

Flatten [M] - утворює вектор з будь-якого списку, наприклад, матриці;

Flatten [M, n]- утворює вектор у рядку n матриці M;

Sort [f] - сортує елементи вектора (матриці) f в природному порядку;

Reverse [f] - повертає вектор (матрицю) f в зворотному порядку розташування елементів;

RotateLeft [f] - повертає елементи вектора (матриці) f після одноразового повороту вліво;

RotateLeft [f, n] - повертає елементи вектора (матриці) f після n-кратного повороту вліво;

RotateRight [f] - повертає елементи вектора (матриці) f після одноразового повороту вправо;

RotateRight [f, n] - повертає елементи вектора (матриці) f після n-кратного повороту вправо;

Transpose [M] - зміна рядків і стовпців матриці.

Приклади використання функцій наведено у лістингу 4.

V={1,3,6,2,4,5}
{1,3,6,2,4,5}
M={{2,1,4},{3,8,6},{1,2,3}}

{{2,1,4},{3,8,6},{1,2,3}}
Flatten [M]
{2,1,4,3,8,6,1,2,3}
FlatteiAt[M,2]
{{2, 1,4} ,3,8,6,(1,2,3}}
Sort [V]
{1,2,3,4,5,6}
Sort [M]
{{ 1, 2,3} ,{2, 1,4}, {3,8,6}}
Reverse [V]
{5,4,2,6,3,1}
Reverse [M]
{{ 1, 2,3} ,{3,8, 6}, {2, 1,4}}
RotateLeft [V,3]
{2,4,5,1,3,6}
RotateRight [V, 2]
{4,5,1,3,6,2}
RotateLeft [M,1]
{{3,8, 6} ,{ 1,2,3}, {2, 1,4}}
RotateRight [M,1]
{{ 1, 2,3} ,{2, 1,4}, {3,8,6}}
Transpose [M]
{{2,3, 1} ,{ 1,8,2}, {4,6,3}}

Лістинг 4

Виділення елементів вектора і матриці за допомогою функції Part

Функція Part представляється в наступному вигляді:

{Part[f, n_1], Part[f, n_2],...},

де f - ім'я вектора;

n_i - і-й елемент вектора f.

У разі виділення елементів матриці функція Part має вигляд:

{Part[f, n_1, m_1], Part[f, n_2 , m_2],...},

де n_i - і-й елемент рядка матриці;

m_i - і-й елемент стовпчика матриці.

У разі виділення елементів зі складних елементів вектора або матриці функція Part представляється в наступному вигляді:

Part[f, n, m, 1],

де f - ім'я вектора або матриці;

n - номер елемента вектора f;

m - рівень виразу (m = l у разі вектора, m = 2 у разі матриці);

1 - номер елемента у векторі або матриці.

Приклади використання функції Part показано у лістингу 9.

f3={a,1,b,2,3,c}
{a,1,b,2,3,c}
f4={2,a,Sin[x+y^2]
{2, a, Sin[x + y^2], b, 1}
{Part[f3 ,2] ,Part[f3 ,5] ,Part [f3,6] }
{1,3,c}
{Part[f3 ,-3] ,Part [f3,-5] ,Part [f3,-6] }
{2,1,a}
{Part [f4,4] ,Part [f4,3,1,1] ,Part [f4,3 ,1,2] }
{b, x, y^2}

Лістинг 9

Комбінування векторів і матриць

Комбінування векторів і матриць здійснюється за допомогою наступних функцій:

Union [F] - повертає новий вектор або матрицю, з якого вилучені повторювані елементи;

Union [f_1, f_2, ...] - об'єднує f_1, f_2 видаляючи повторювані елементи векторів і матриць;

Join [f_1, f_2...] - Об'єднує f_1, f_2 ... в єдиний ланцюг (конкатенація);

Complement [f_1, f_2, ...] - Повертає список f, елементи якого не містяться в f_1, f_2, ...;

Intersection [f_1, f_2, ...] - повертає впорядкований список елементів, загальних для всіх списків.

У лістингу 12 наведено приклади використання функцій.

```
Vl={a,3,l,7,2,4,6}
{a,3,1,7,2,4,6}
V2={1,»,3,4,5,8}
{1,b,3,4,5,8}
V3={4,5,c,7,8,l}
{4,S,c,7,8,1}
M1={{1,2,4},{3,5,8},{7,2,3}}
{{ 1,2,4} ,{3,5,8}, {7,2,3}}
M2={{a,Ь,c},{1,2,3},{4,5,6}}
{{ a,b,c},{1,2,3},{4,5,6}}
Union[Vl ,V2 ,V3 ,M1 ,M2]
{1, 2,3,4, 5, 6,7,8,a,Ь,c,{ 1,2,3},{1,2,4},{3,5,8},{4,5, 6}, {7,2,3} ,{a,b,c}
Union[{l,2,3,1,4,5,2,4,1,7}]
{1,2,3,4,5,7}
Join[Vl,V2 ,V3,M1,M2]
{a, 3, 1,7, 2, 4, 6, 1,b,3 ,4,5,8 ,4,5, c, 7,8, 1, {1,2,4}, {3 ,5,8} , {7, 2,3} ,{a,b, c} ,{1,2,3}, {4,5,6}}
Complement [Vl ,V2 ,V3]
{2,6,a}
Intersection [Vl ,V2, V3 ]
{1,4}
```

Лістинг 12

Математичні операції над векторами і матрицями

Найпростіші арифметичні операції над векторами і матрицями розглянемо на наступному прикладі.

Дано вектори V1, V2 і матриці Ml, M2:

V1 = {1, 3, 5, 2, 6, 4};

V2 = {2, 7, 5, 8, 1, 3};

M1 = {{1, 2, 3}, {6, 5, 4}, {1, 3, 5}};

M2 = {{3, 2, 1}, {4, 5, 6}, {5, 3, 1}}.

Необхідно:

- скласти, відняти, помножити і розділити вектор V1 і матрицю Ml на число 3;

- піднести в квадрат вектор V2 і матрицю M2;

- взяти квадратний корінь з вектора V1 і матриці Ml;

- обчислити e^{V1}, sin(V2), Ln(M1), cosh(M1).

Функції, що реалізують математичні операції наведено в лістингу 13.

V1={1,3,5 ,2,6,4}
{1,3,5,2,6,4}
V2={2,7,5,8,1,3}
{2,7,5,8,1,3}
M1={ {1,2,3}, {6,5,4}, {1,3, 5}}
{{ 1, 2,3} , { 6,5,4} ,{1,3,5}}
M2={{3,2,1},{4,5,6} ,{5,3,1}}
{{3,2, 1} , {4,5, 6} , {5,3,1}}
(V1+ 3 ,V1-3 ,V1***3**,V1/3**}**
{{4, 6, 8, 5, 9, 1}, {-2, 0, 2, -1, 3, 1}, {3, 9, 15, 6, 18, 12},{$\frac{1}{3}$,1,$\frac{5}{3}$,$\frac{2}{3}$,2,$\frac{4}{3}$}}
{M1+ 3 ,M1-3 ,M1*3, M1/3}
{{{4, 5, 6}, {9, 8, 1}, {4, 6, 8}}, {{-2, -1, 8}, {3, 2, 1}, {-2, 0, 2}}, {{3, 6, 9}, {18, 15, 12}, {3, 9, 15}},{{ $\frac{1}{3}$,$\frac{2}{3}$,1},{2,$\frac{5}{3}$,$\frac{4}{3}$},{$\frac{1}{3}$,1,$\frac{5}{3}$}}}
{V2^2,M2^2,Sqrt[V1] ,Sqrt[M1] }
{{4 , 49 , 25 , 64, 1, 9}, {{9, 4, 1}, {16, 25, 36}, {25, 9, 1}},
{1, $\sqrt{3}$, $\sqrt{5}$, $\sqrt{2}$, $\sqrt{6}$, 2}, {{1, $\sqrt{2}$, $\sqrt{3}$, 2}, {$\sqrt{6}$, $\sqrt{5}$, 2}, {1, $\sqrt{3}$, $\sqrt{5}$}}}
{E^Vl,Sin[V2] ,Log [Ml] ,Cosh[M2]}e
{ e, e^3 , e^5 , e^2 , e^6, e^4}, {Sln[2], Sin[7], Sln[5], Sin[8], Sin[1], Sin[3]}, {{0, Log[2], Log[3] }, {Log[6], Log[5], Log[4]}, {0, Log[3], Log[5] }}, {{Cosh[3], Cosh[2], Cosh[l]}, {Cosh[4], Cosh[5], Cosh[6]}, {Cosh[5], Cosh[3], Cosh[l]}}}
N[%]
{{2 .71828, 20.0855, 143.413, 7.33906, 403.429, 54.5982} ,{0.909297, 0.656987, 0.953924, 0.989353, 0.341471, 0.14112}, { {0 ., 0.693147, 1.09861}, {1.79176, 1.60944, 1.3862 9} , {0 ., 1.09861, 1.60944} }, {{ 10.0677, 3.7622, 1.54308} , { 27.3082, 74.2099, 201.716}, {74.2099, 10.0677, 1.54303}}}

Лістинг 13

Інші функції, призначені для роботи з векторами і матрицями наведені нижче:

Det [M] - повертає детермінант (головний визначник) матриці;

IdentityMatrix [M] - повертає одиничну матрицю: матрицю з діагональними елементами, рівними 1, і рештою нулями;

Transpose [M] - повертає транспоновану матрицю, у якої рядки замінені стовпцями, а стовпці рядками;

Inverse [M] - повертає зворотну матрицю, тобто таку, яка будучи помноженою на вихідну дає одиничну матрицю;

Tr [M] - повертає слід матриці (суму діагональних елементів);

LinearSolve [M, b] - повертає вектор невідомих матричного рівняння M * x = b, де M - матриця коефіцієнтів системи рівнянь, x - вектор невідомих, b - вектор вільних членів;

Eigensistem[M] - повертає список власних значень і власних векторів квадратної матриці M;

Eigenvalues [M] - повертає список власних значень квадратної матриці M;

Eigenvectors[M] - повертає список власних векторів квадратної матриці M;

Pseudoinverse[M] - шукає псевдоподібну матрицю для матриці M.

Всі перераховані вище матричні операції ілюструються у лістингу 14.

Ml={{1,2,3) ,{3,5,4),{7,2,1))
{{1,2,3},{3,5,4},{7,2,1}}
M2={{4,a,-2) ,{1,2,b),{3,5,c}}
{{4,a,-2}, {1,2,b} ,{3,5,c} }

{Det [Ml] ,Det [M2] }
{-40,2-20 b+3 a b-8 c-a c}
IdentityMatrix [3]
{{1,0,0},{0,1,0},{0,0,1}}
{Transpose [Ml] ,Transpose [112] }
{{{1,3,7},{2,5,2},{3,4,1}},{{4,1,3},{a,2,5},{-2,b,c}}}
{Inverse[M1],Inverse[M2]}
$\{\{\{\frac{3}{40},-\frac{1}{10},\frac{7}{40}\},\{-\frac{5}{8},\frac{1}{2},-\frac{1}{8}\},\{\frac{29}{40},-\frac{3}{10},\frac{1}{40}\}\},$
$\{\{\frac{-5b+2c}{2-20b+3ab+8c-ac},\frac{-10-ac}{2-20b+3ab+8c-ac},\frac{4+ab}{2-20b+3ab+8c-ac}\},$
$\{\frac{3b-c}{2-20b+3ab+8c-ac},\frac{6+4c}{2-20b+3ab+8c-ac},\frac{-2-4b}{2-20b+3ab+8c-ac}\},$
$\{-\frac{1}{2-20b+3ab+8c-ac},\frac{-20+3a}{2-20b+3ab+8c-ac},\frac{8-a}{2-20b+3ab+8c-ac}\}\}\}$
{Tr[M1],Tr [M2]}
{7, 6+c}
{LinearSolve [M1,{1,3,7)] ,LinearSolve [M2, {1,2 ,3}] }
$\{\{1,0,0\},\{\frac{-8-5b+3ab+2c+2ac}{2-20b+3ab+8c-ac},\frac{6-9b+7c}{2-20b+3ab+8c-ac},\frac{-17+3a}{2-20b+3ab+8c-ac}\}\}$
PseudoInverse [M1]
$\{\{\frac{3}{40},-\frac{1}{10},\frac{7}{40}\},\{-\frac{5}{8},\frac{1}{2},-\frac{1}{8}\},\{\frac{29}{40},-\frac{3}{10},\frac{1}{40}\}\}$
Eigenvalues [M1]
{Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 3], Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 1 Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 2]}
N[%]
{9.14592, -3.42345, 1.27752}
Eigenvectors [M1]
{{ $\frac{17}{71}+\frac{13}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 3]- $\frac{1}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 3]^2 ,
$-\frac{95}{71}-\frac{10}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 3]+ $\frac{7}{142}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 3]^2, 1},
{ $\frac{17}{71}+\frac{13}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 1]- $\frac{1}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 1]^2 ,
$-\frac{95}{71}-\frac{10}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 1]+ $\frac{7}{142}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 1]^2, 1}
{ $\frac{17}{71}+\frac{13}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 2]- $\frac{1}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 2]^2 ,
$-\frac{95}{71}-\frac{10}{71}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 2]+ $\frac{7}{142}$ Root[40 - 24 #1 - 7 #1^2 + #1^3 &, 2]^2, 1}}
N[%]
{{0.735903,1.4973,1.},{-0.552462,-0.278106,1.},{0.450362,-1.43751,1.}}

Лістинг 14

Векторний добуток реалізується за допомогою функції Dot [V1, V2] або знаком множення у вигляді точки: V1. V2 або Ml. M2, див. лістинг 15.

V1={1,3,5,2,6,4}
{1,3,5,2,6,4}
V2={2,7,5,8,1,3)
{2,7,5,8,1,3}
M1={{1,2,3},{6,5,4},{1,3,5}}

{{ 1,2,3} ,{6,5,4}, {1,3,5}}
M2={{3,2,1},{4,5,6},{5,3,1}}
{{3,2, 1} ,{4,5, 6}, {5,3,1}}
V1*V2
{2,21,25,16, 6, 12}
Vl .V2
82
Ml .M2
{{26,21, 16}, {58,49,40}, {40,32,24} }
Dot [Vl ,V2]
82
Dot [V1 ,V2]
{{26,21, 16}, {58,49,40}, {40,32,24} }

Лістинг 15

Лекція № 5. Графічні функції системи Mathematica

Функція Plot

Функція Plot дозволяє будувати графіки, задані аналітично, в двовимірному просторі у прямокутній системі координат. На одному графіку може бути зображено декілька функцій. За замовчуванням на екран виводиться координатна сітка.

Формат запису функції Plot

Plot[f,{x,xmin,xmax}];

Plot[{f1,f2,...},{x,xmin,xmax}],

де f – функція, графік якої будується,

fi – i-а функція, графік якої будується, i=1,2,…

x – аргумент функції,

xmin, xmax – інтервал зміни аргументу x.

Опції функції Plot

Опції функції Plot задаються в наступному вигляді:

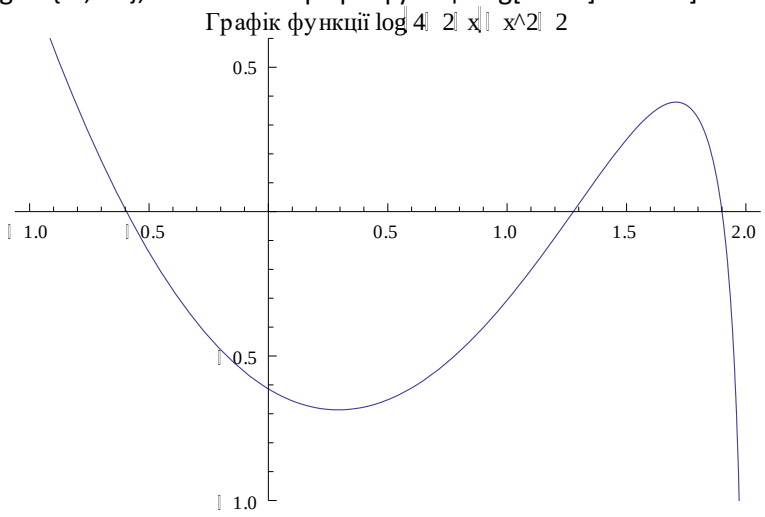
Ім’я опції -> значення опції.

Основними опціями функції Plot є:

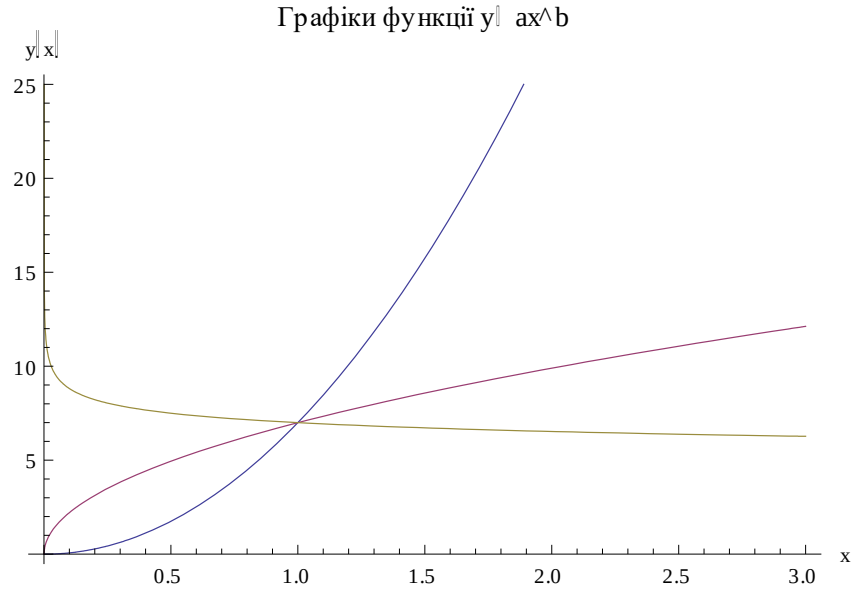
- встановлення масштабу по осі:
  PlotRange -> {ymin,ymax} – встановлює масштаб по осі y від ymin до ymax з автоматичним вибором кроку;
- визначення імені осі:
  AxesLabel -> {“Tx”, “Ty”} – встановлює надписи Tx і Ty по осям x і y відповідно;
- визначення імені графіка:
  PlotLabel -> “T” – встановлює ім’я графіка;
- вибір стилю графіка:
  Axes -> None – графік будується без осей.

Приклади використання функції Plot.

```
f=Log[4-2*x]+x^2-2
Plot[f,{x,-1,2},PlotRange->{-1,0.6},PlotLabel->“Графік функції log[4-2*x]+x^2-2”]
```




```
Plot[{7*x^2,7*x^0.5,7*x^(-0.1)},{x,0,3},AxesLabel->{“x”,“y(x)”},PlotLabel-> “Графіки функції y=a*x^b”]
```



Лістинг 1

Функція ListPlot

Використовується для побудови графіків, заданих у вигляді масиву точок.

Формат запису функції ListPlot

ListPlot[{y1,y2,…}];

ListPlot[{x1,y1},{x2,y2},…}],

де yi – i-е значення функції y(x),

xi - i-е значення аргументу функції y(x).

Призначення функції ListPlot

ListPlot[{y1,y2,…}] – виводить точкові значення функції y(x) з позначенням номеру точок по осі x.

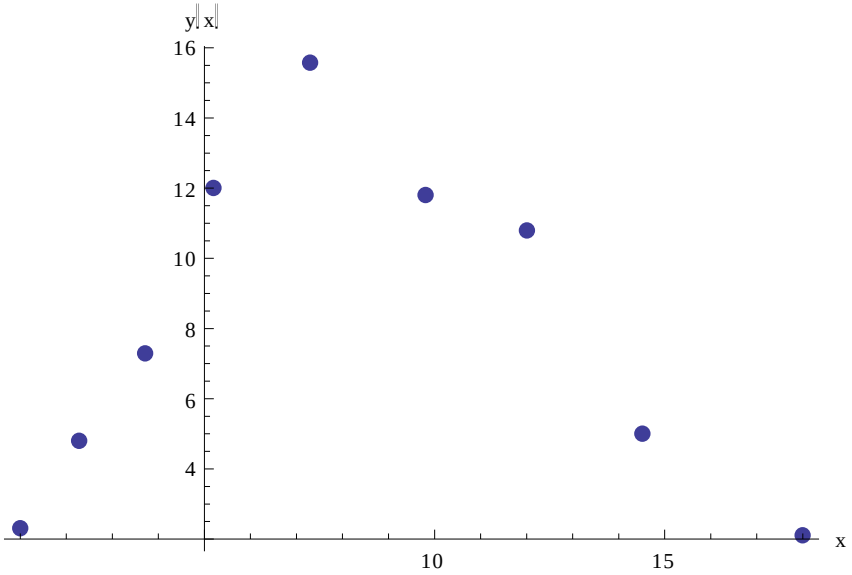
ListPlot[{x1,y1},{x2,y2},…}] – виводить точкові значення функції y(x) з позначенням аргументу функції по осі x.

Функція ListPlot має дві опції:

- визначення імені осі:
  AxesLabel -> {“Tx”, “Ty”} – встановлює надписи Tx і Ty по осям x і y відповідно;
- визначення розміру точок:
  PlotStyle -> PointSize[d] – встановлює діаметр точки, рівний d.

Приклад використання функції ListPlot.

```
f = {{1,2.3},{2.3,4.8},{3.7,7.3},{5.2,12},{7.3,15.6},{9.8,11.8},{12,10.8},
{14.5,5},{18,2.1}}
ListPlot[f,AxesLabel->{“x”,“y(x)”},PlotStyle -> PointSize[0.02]]
```



Лістинг 2

Функція Show

Використовується для побудови точкових і аналітичних графіків в одній площині.

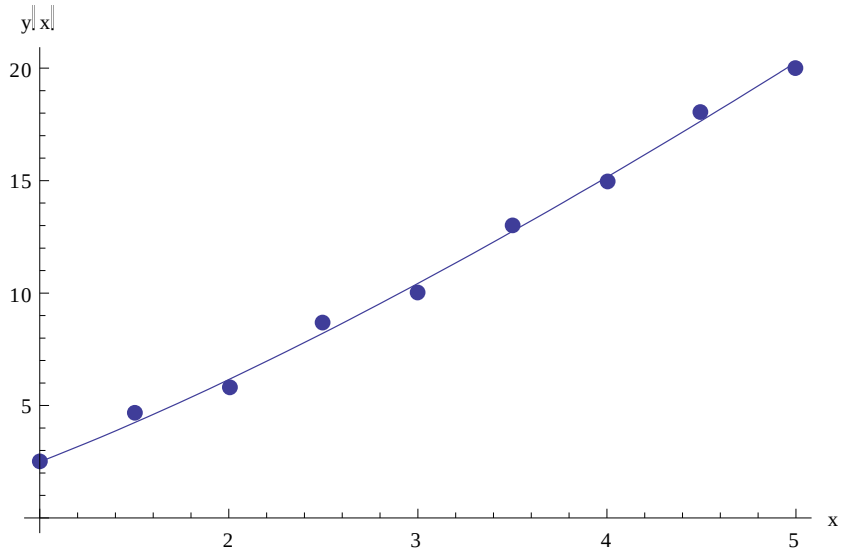
Формат запису функції Show

Show[r1,r2],

де r1, r2 – змінні, які використовуються для позначення графіків.

Приклад використання функції ListPlot.

```
f = {{1,2.5},{1.5,4.7},{2,5.8},{2.5,8.7},{3,10},{3.5,13},{4,15},{4.5,18},{5,20}}
r1:=ListPlot[f,AxesLabel->{“x”,“y(x)”},PlotStyle->PointSize[0.02]]
r2:=Plot[2.5x^1.3,{x,1,5}]
Show[r1,r2]
```



Лістинг 3

Вибір стилю графіка

Опція PlotStyle

Опція PlotStyle дозволяє обрати колір ліній і їх товщину.

Директиви опції PlotStyle

- колір лінії:

PlotStyle ->{GrayLevel[k1],GrayLevel[k2],…},

де k1,k2,… - коди кольору ліній у відтінках сірого відповідних функцій, обираються з діапазону 0..1;

PlotStyle ->{Hue[c1],Hue[c2],…},

де c1,c2,… - табличні коди кольору ліній відповідних функцій, обираються з діапазону 0..1;

PlotStyle ->{ RGBColor[ч1,з1,с1],Hue[ч2,з2,с2],…},

де ч1,з1,с1… - яскравості червоної, зеленої і синьої компонент кольору, обираються з діапазону 0..1;

- товщина лінії:

PlotStyle ->Thickness[d] – встановлює товщину ліній графіка, як долю його повної ширини;

PlotStyle ->AbsoluteThickness[d] – встановлює товщину ліній графіка в пікселях;

- стиль штрихування:

PlotStyle ->Dashing[{d1,d2,… }] – встановлює довжину штриха ліній графіка, де di задається в частках від ширини лінії графіка;

PlotStyle->AbsoluteDashing[d1] - встановлює довжину штриха ліній графіка, де di задається в пікселях;

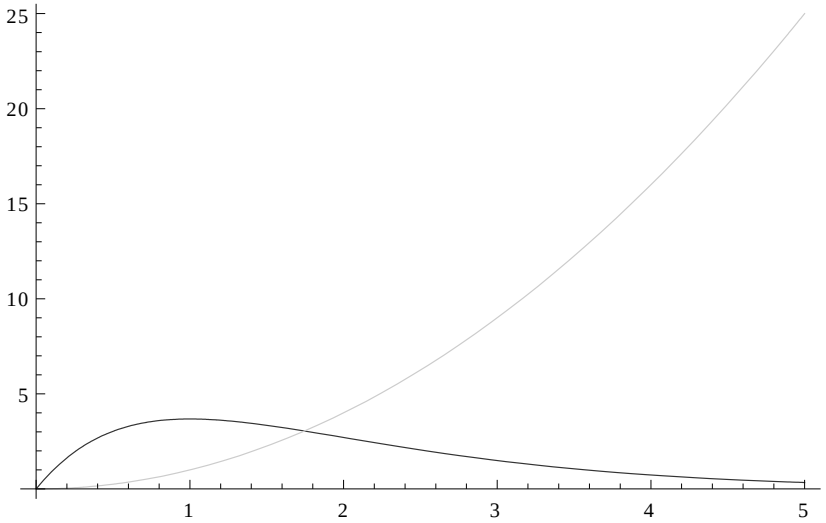
- точковий графік:

PlotStyle->PointSize[d] – графік у вигляді кіл діаметром d, які вимірюються в частках від загальної ширини графіка;

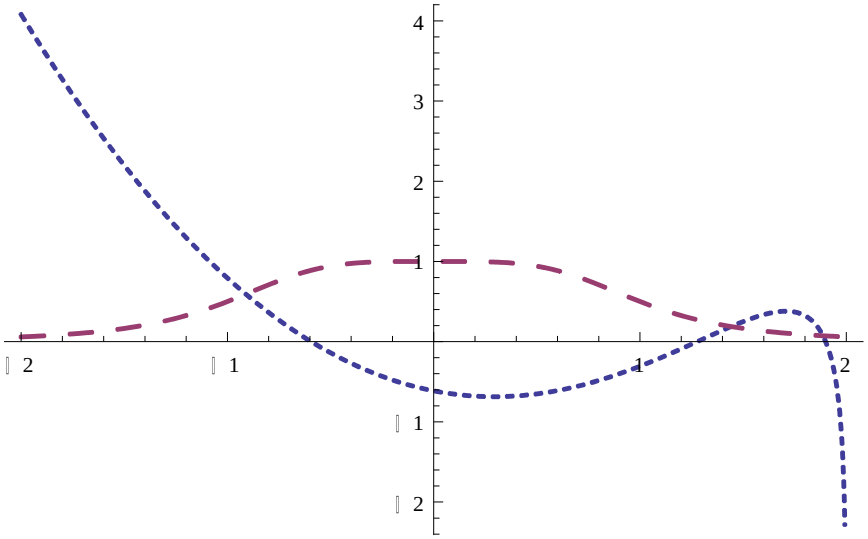
PlotStyle->AbsolutePointSize[d] – графік у вигляді кіл діаметром d, які вимірюються в пікселях.

Приклади використання опції PlotStyle.

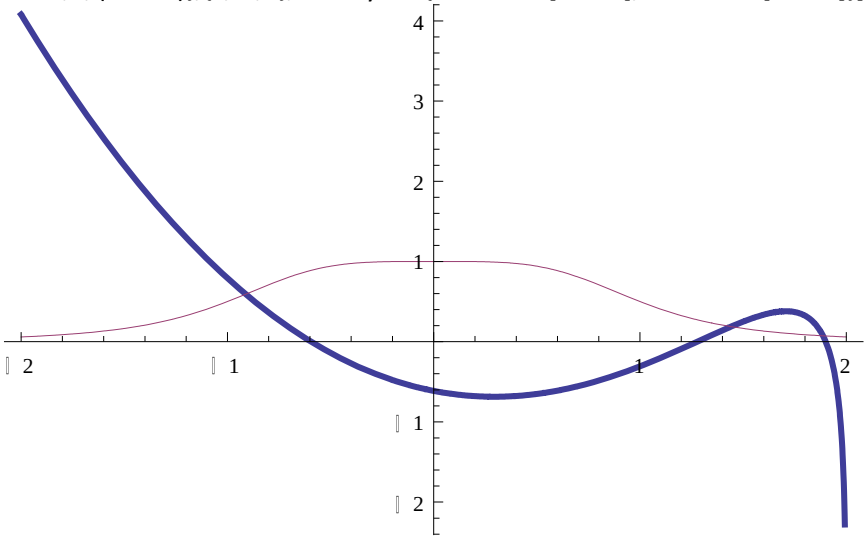
```
Plot[{10*x*Exp[-x],x^2},{x,0,5}, PlotStyle->{ GrayLevel[0.2],GrayLevel[0.8]}]
```



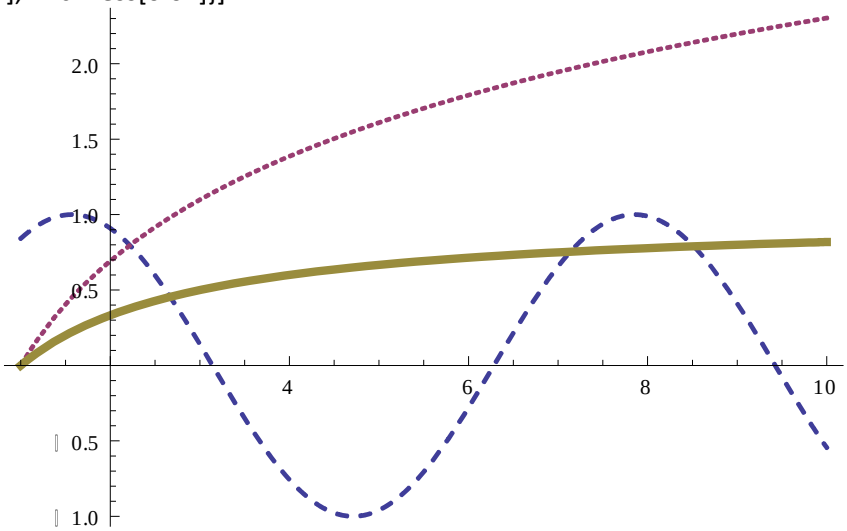

```
Plot[{Log[4-2*x]+x^2-2,1/(1+x^4)},{x,-2,2},PlotStyle->{AbsoluteDashing[3], AbsoluteDashing[10]}]
```




```
Plot[{Log[4-2*x]+x^2-2,1/(1+x^4)},{x,-2,2},PlotStyle->{Thickness[0.007], Thickness[0.001]}]
```




```
Plot[{Sin[x],Log[x],(x-1)/(x+1)},{x,1,10},PlotStyle->{AbsoluteDashing[5],
AbsoluteDashing[2],Thickness[0.01]}]
```



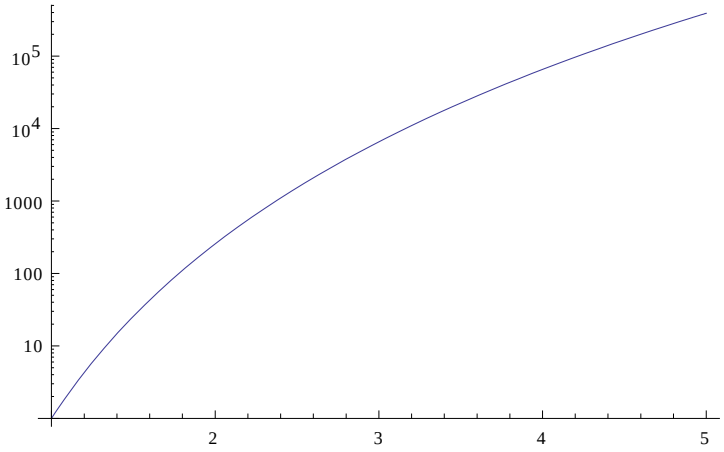
Лістинг 4

Графіки спеціальних типів

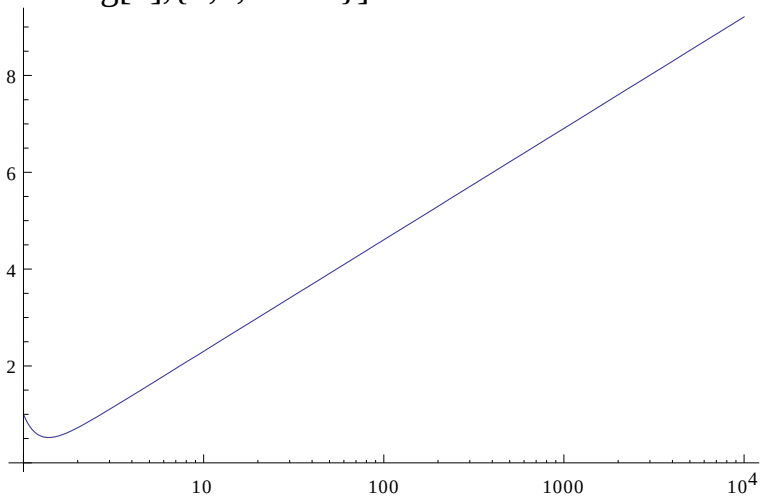
Функції побудови графіків в логарифмічному масштабі

LogPlot[f,{x,xmin,xmax}] – створює лінійно-логарифмічний графік функції f з логарифмічною шкалою по осі y в діапазоні xmin..xmax.

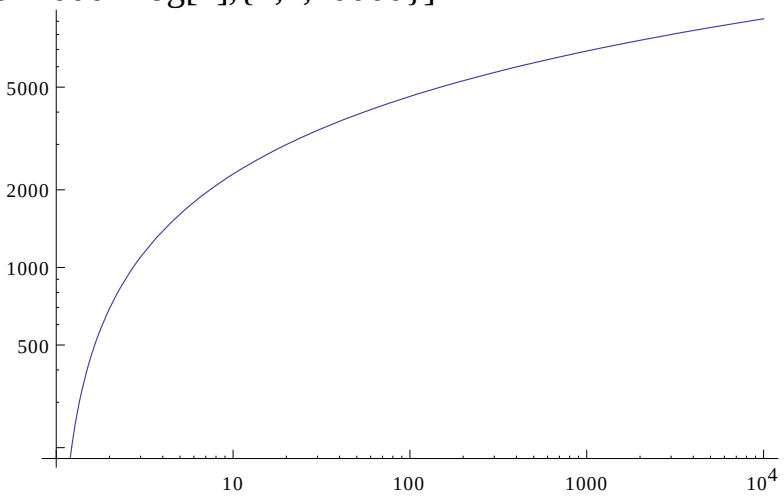
LogPlot[x^8,{x,1,5}]



LogLinearPlot[f,{x,xmin,xmax}] – створює логарифмічно-лінійний графік функції f з логарифмічною шкалою по осі x в діапазоні xmin..xmax.
LogLinearPlot[x^-5+Log[x],{x,1,10000}]



LogLogPlot[f,{x,xmin,xmax}] – створює графік функції f з логарифмічною шкалою по двом осям в діапазоні xmin..xmax.
LogLogPlot[x^-5+1000*Log[x],{x,1,10000}]



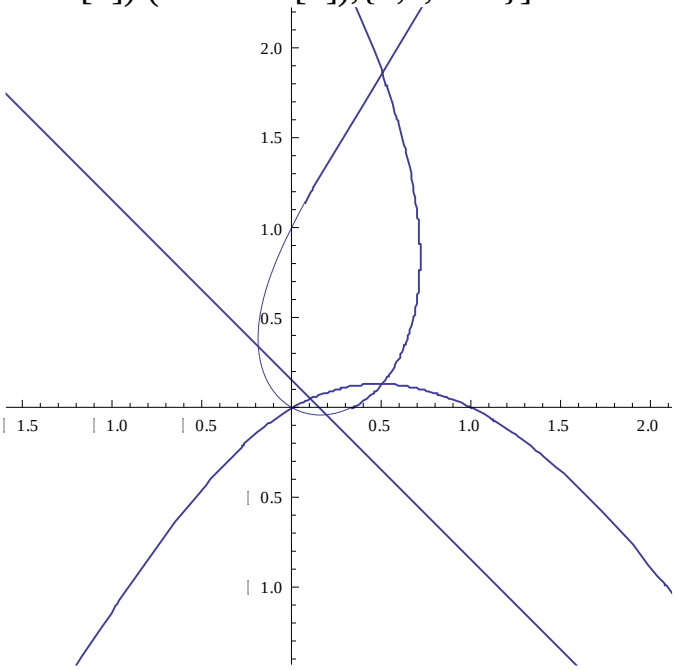
Функції
LogListPlot[{x1,y1},{x2,y2},…]
LogLinearListPlot[{x1,y1},{x2,y2},…]

LogLogListPlot[{x1,y1},{x2,y2},…]
аналогічні трьом попереднім і використовуються для побудови точкових графіків.

Функція побудови графіків в полярній системі координат

PolarPlot[f,{t,tmin,tmax}] – будує графік положення кінця вектора f при зміні кута t від tmin до tmax.

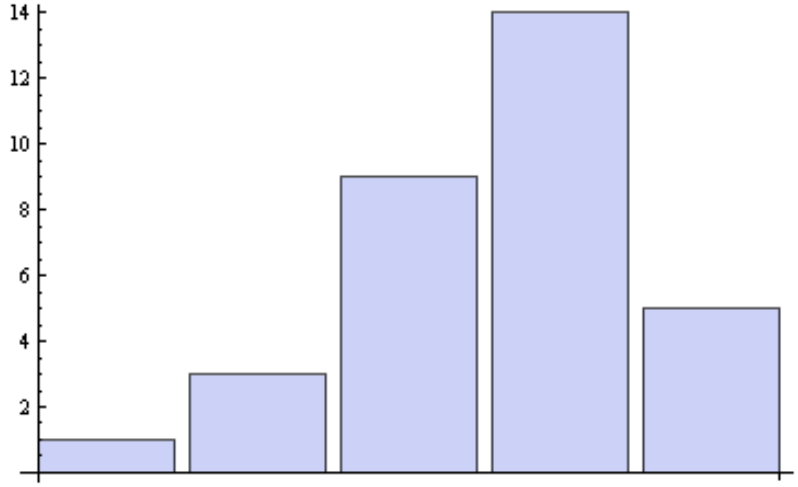
PolarPlot[(1+2*Sin[x])/(1+2*Cos[x]),{x,0,2*Pi}]



Функція побудови діаграм

BarChart[{c1,c2,…}] – будує діаграму у вигляді стовпців за даними списку.

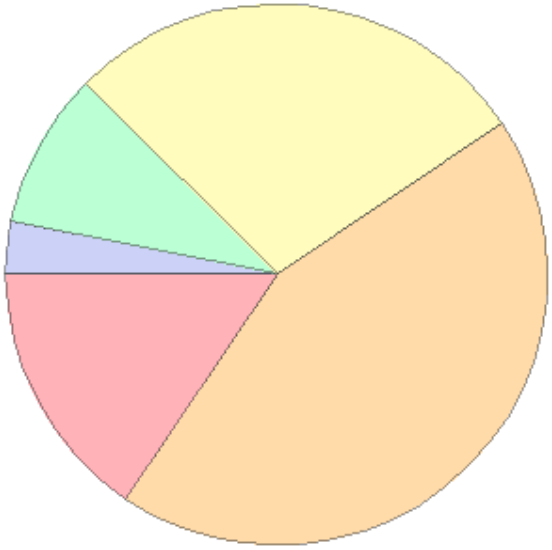
BarChart[{1,3,9,14,5}]



PieChart[{c1,c2,…}] – будує кругову діаграму за даними списку.

Разом з функцією PieChart використовується ряд опцій, опис яких доступний за командою Options[PieChart].

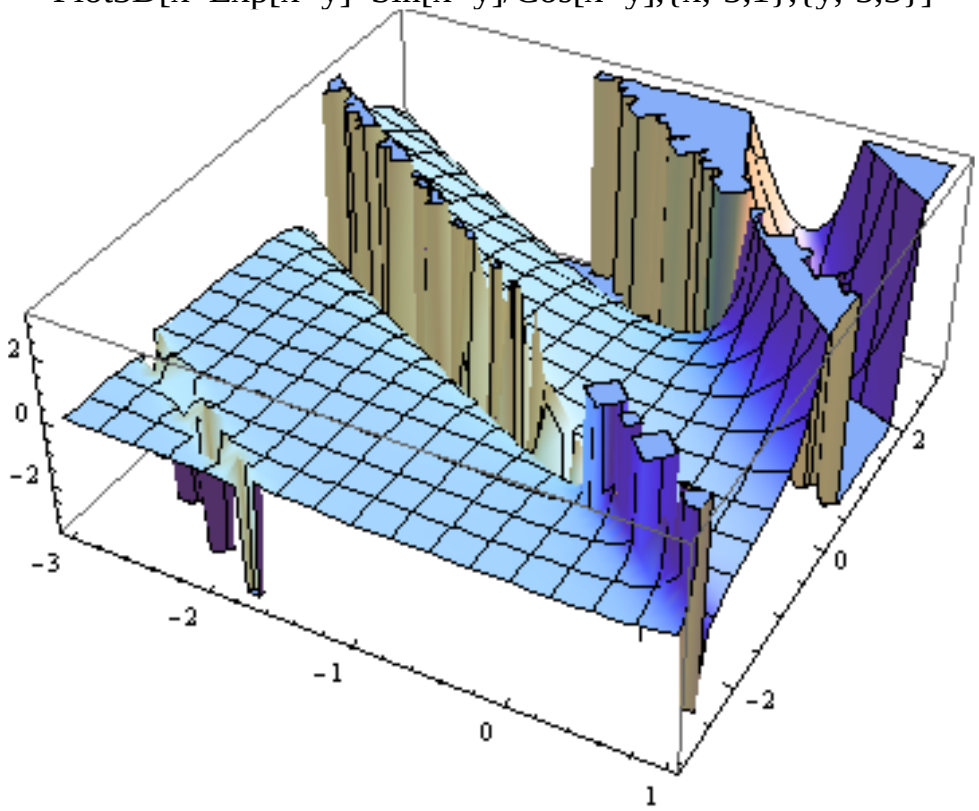
PieChart[{1,3,9,14,5}]



Функції тривимірної графіки

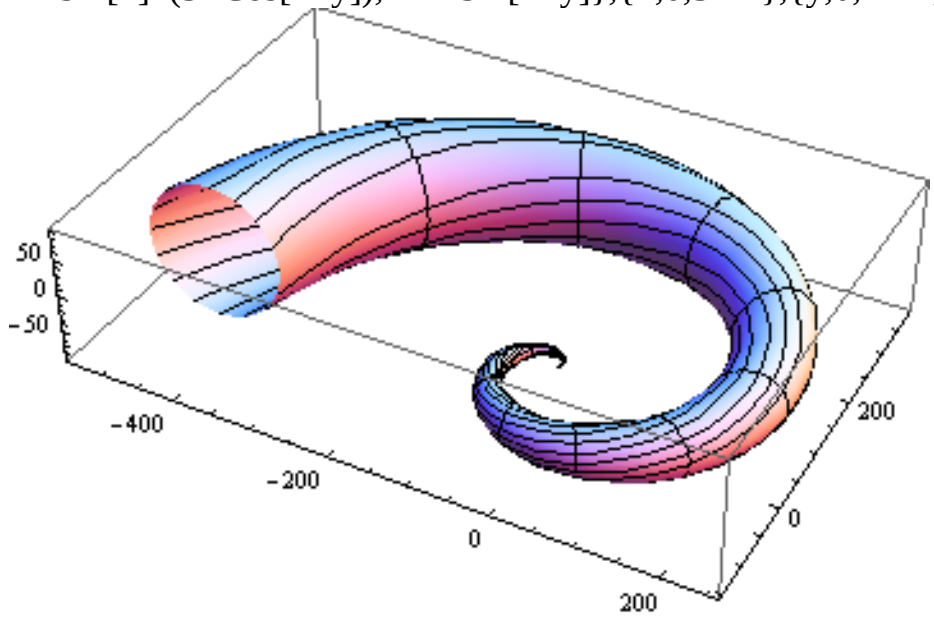
Plot3D[f,{x,xmin,xmax},{y,ymin,ymax}] – будує графік функції f=f(x,y). Опції функції Plot3D доступні за командою Options[Plot3D].

Plot3D[x*Exp[x+y]*Sin[x+y]/Cos[x+y],{x,-3,1},{y,-3,3}]



ParametricPlot3D[{f1,f2,f3},{t1,t1min,t1max},{t2,t2min,t2max}] – будує тривимірний графік параметрично заданої функції z(t1,t2)=f(x(t1,t2),y(t1,t2)). Опції функції Plot3D доступні за командою Options[ParametricPlot3D].

ParametricPlot3D[{x^2*Cos[x]*(5+Cos[x+y]),
x^2*Sin[x]*(5+Cos[x+y]),x^2*Sin[x+y]},{x,0,3*Pi},{y,0,2*Pi}]



Лекція № 6. Можливості роботи системи Mathematica з різнорідними даними

Позиція Graphics меню і графічний редактор.

При створенні складних ноутбуків в колишніх версіях Mathematica явно бракувало коштів для підготовки хоча б простих малюнків і діаграм, якими часто супроводжуються математичні і науково-технічні розрахунки. Подібні малюнки і діаграми, зрозуміло, можна створювати засобами програмування систем Mathematica, але це вимагає багато часу і вміння добре програмувати графічні завдання. Врахувавши це, розробники Mathematica ввели засіб побудови за допомогою миші простих малюнків за типом добре відомого графічного редактора Paint. Доступ до нього забезпечений з нової позиції Graphics меню. Вона містить наступні команди:

- New Graphic - висновок вікна для побудови графіка;
- Drawing Tool - висновок вікна графічного редактора;
- Graphics Inspector - висновок вікна інспектора графіки;
- Rendering - висновок підміню операцій рендеринга;
- Opetations - висновок підміню додаткових операцій.

Робота з зазначеними графічними засобами проста і очевидна. Її ілюструє рис. 1. На ньому показані вікна малюнка, графічного редактора і інспектора графіки. Відзначимо, що у графічного редактора немає коштів для побудови незафарбовані еліпса, прямокутника і полігону. Однак установкою кольорів ці фігури нескладно отримати.

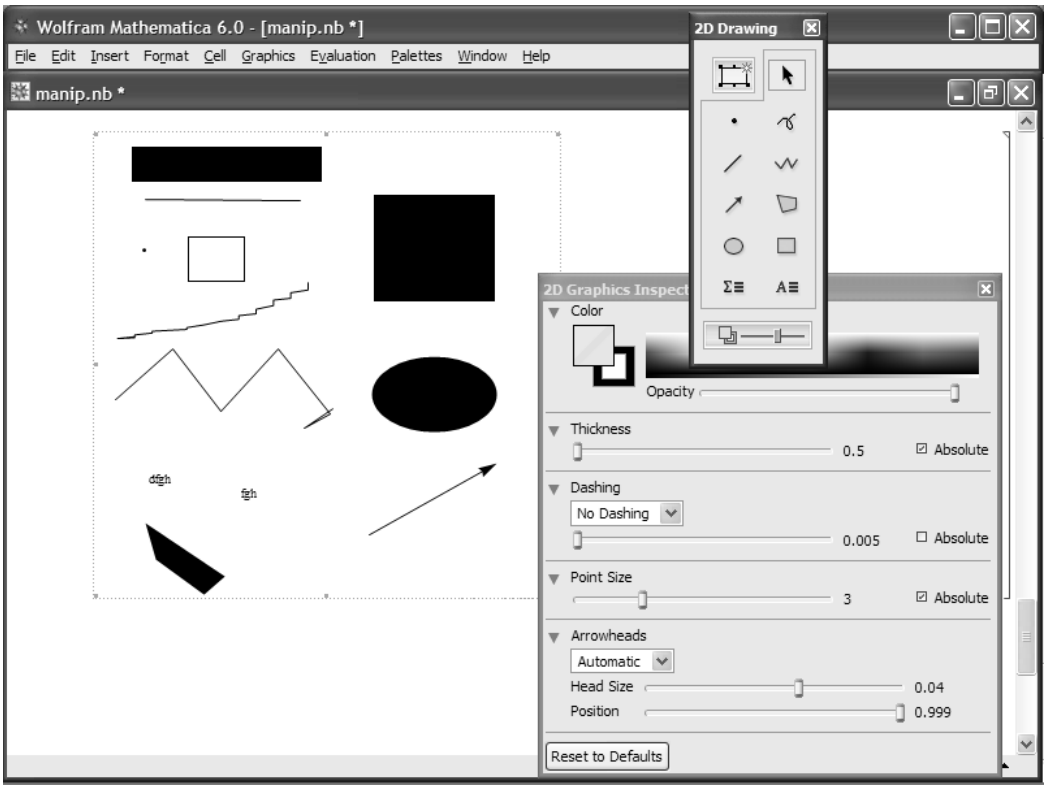


Рис. 1.

Використання опцій зафарбовування областей двовимірних графіків.

З нових опцій функції **Plot** в Mathematica найбільш ефектно виглядають опції зафарбовування областей двовимірних графіків **Filling** (Зафарбування) і **FillingStyle** (Стиль зафарбовування). Перша за замовчуванням відключена, друга має значення **Auto** (Автоматичний вибір стилю).

Продемонструємо дію опції Filling (рис2). На ньому функцією **Plot** будується графік функції **Sin [x] / x** в інтервалі зміни x від -4*Pi до +4*Pi з 4ма типами зафарбовування. Вони представлені значеннями опції **Filling**: **Axis** (зафарбовування йде від кожної точки кривої до осі абсцис), **Top** (зафарбовування області від вершини вікна графіка до його кривої), **Bottom** (зафарбовування від кривої до низу вікна графіка) і 0.5 (зафарбовування від лінії графіка до горизонталі з вертикальною координатою, що дорівнює 0.5).

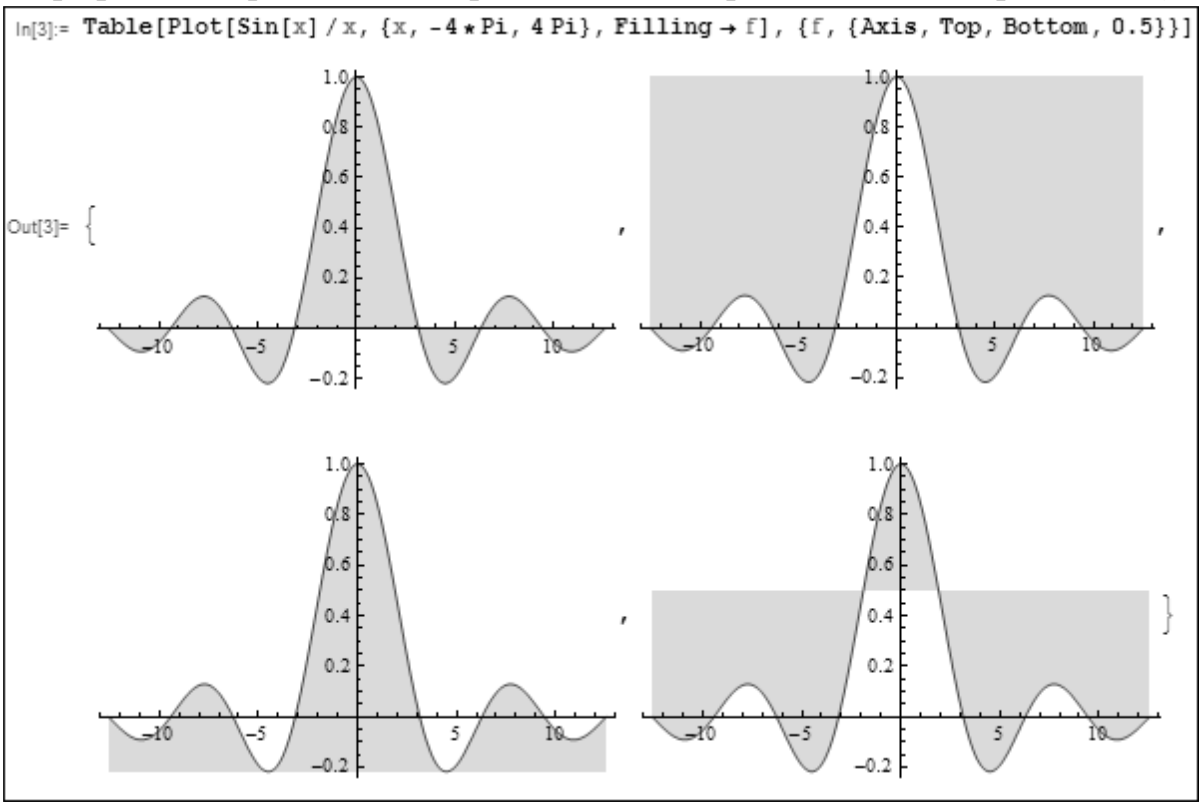


Рис. 2.

Ця опція може використовуватися і з функцією **ListPlot [list]**, яка будує точки з координатами, узятими зі списку list. Вона дає можливість побудови вертикалей, що з'єднують точки графіка з віссю абсцис (рис. 3). На цьому малюнку показано, як змінюється значення простих чисел від їх номера. Цікаво, що ця залежність близька до лінійної.

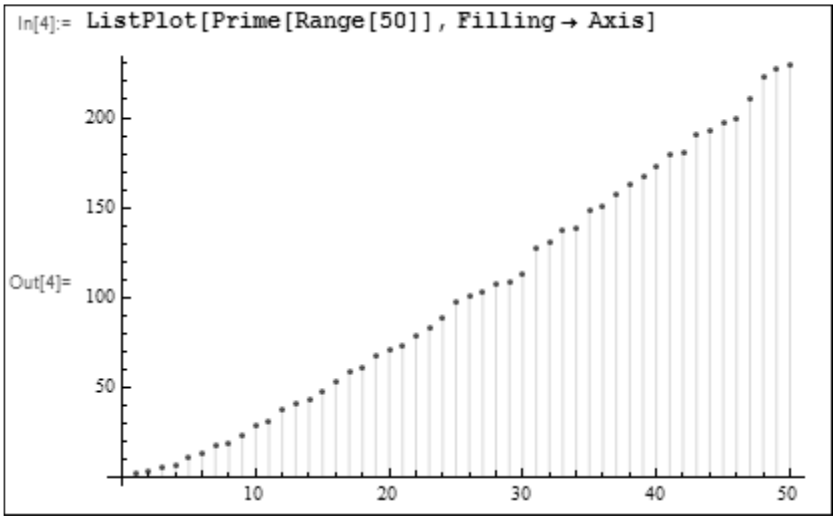


Рис. 3.

Рельєфна графіка.

Функція системи Mathematica **ReliefPlot [array]** служить для побудови реалістичних графіків рельєфу, який задається координатами точок масиву. Прикладом застосування цієї функції служить рис. 8.61, на якому побудований рельєф поверхні, заданої математичною формулою і + cos (i3 + j3), де і і j змінюються з дискретністю 0,03 в інтервалі від -4 до 4. Форма рельєфу залежить від значення опції **ColorFunction**.

Ще один приклад застосування функції **ReliefPlot** представлений на рис. 4. Тут задаються три масиву випадкових результатів ряду арифметичних операцій, що включають норми матриць. Отримані три рельєфу є в значній мірі випадковими і змінюються від пуску до пуску представленого модуля.

```
Table[
 ReliefPlot[
  Table[
   Evaluate[Sum[Product[Norm[{x, y} - RandomReal[{-3, 3}, {2}]], {i, j}] /
      Product[Norm[{x, y} - RandomReal[{-3, 3}, {2}]], {i, j}], {j, 1, 10}]],
   {x, -5, 5, .05}, {y, -5, 5, .05}], PlotRange → Automatic, ColorFunction → "Rainbow",
  ClippingStyle → Darker[Red]], {3}]
```

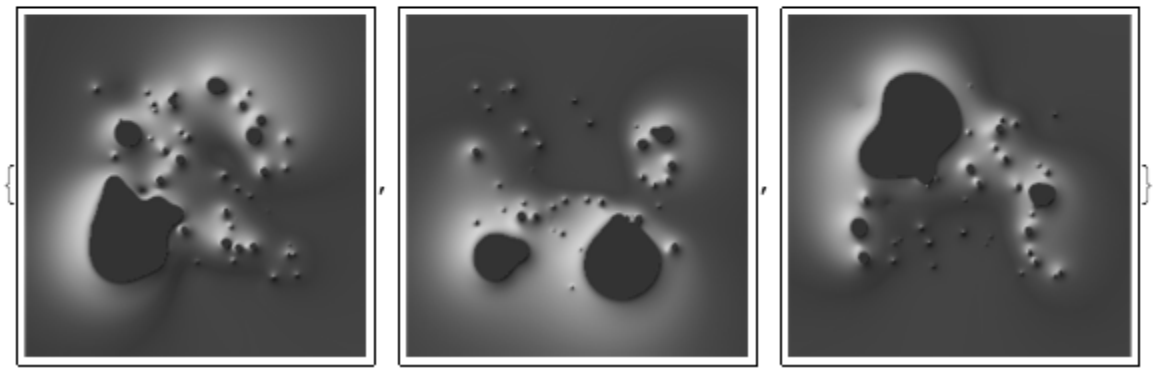


Рис. 4

Рис. 5 будує рельєф уявної частини функції sec (i + I * j) 2 для двох значень опції **PlotRange**, рівних **All** і **Automatic**. Неважко помітити серйозна

зміна характеру виявлення деталей одного і того ж рельєфу. Зрозуміло, масив для цієї функції може створюватися не тільки математичними виразами, а й будь-якими іншими способами - наприклад, завантаженням масивів малюнків. Безліч опцій функції **ReliefPlot** дозволяє створювати рельєфи з різним дозволом, різним забарвленням та іншими особливостями.

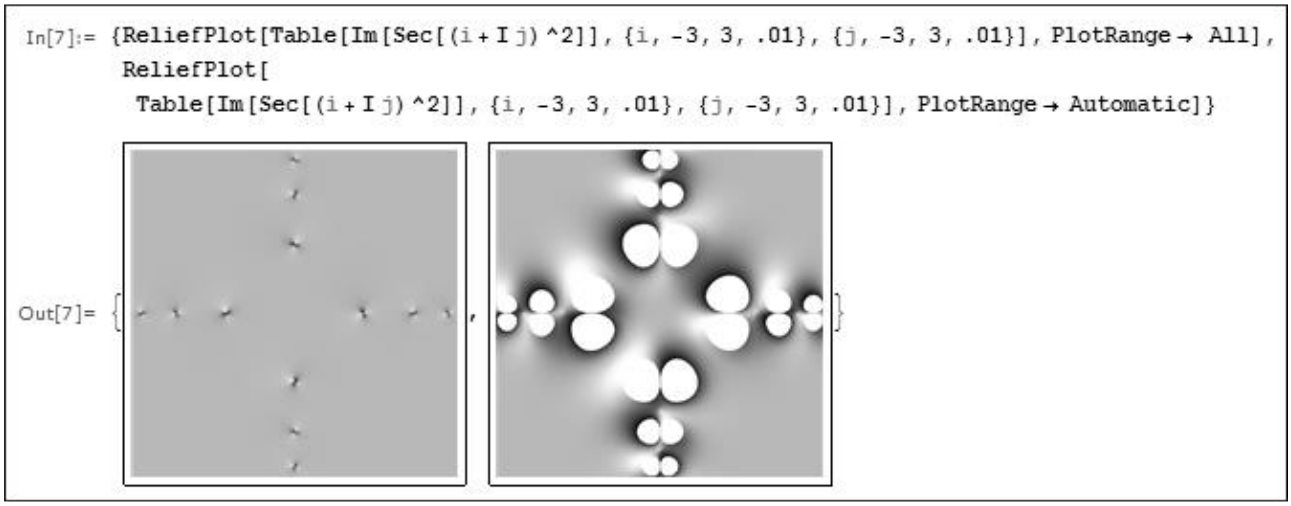


Рис. 5

Тривимірні об’єкти, отримані обертанням кривих.

Досить широко поширеними є тривимірні графічні об’єкти, отримані обертанням кривих щодо деякої осі. наприклад, повертаючи окружність на кут π, можна отримати поверхню кулі. Змінюючи межі зміни кута повороту, можна будувати замкнуті або незамкнуті фігури. Для побудови таких поверхонь (фігур) в Mathematica служить функція:

RevolutionPlot3D[fz,{t,tmin,tmax},...]

RevolutionPlot3D [{fx,fy,fz},{t,tmin,tmax},...]

На рис. 6 показано застосування цієї функції для побудови поверхні половинки бублика. Фігура виглядає досить реалістично.

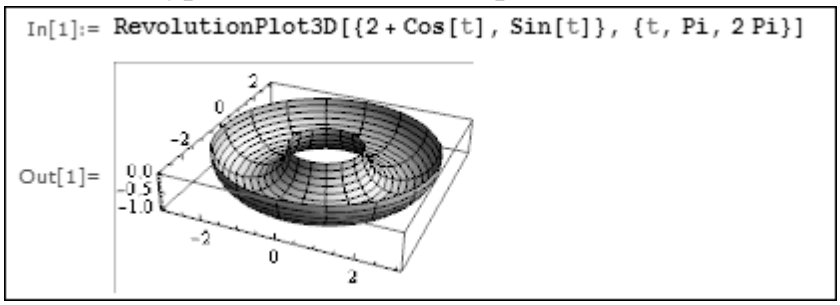


Рис. 6.

Більш кумедна фігура будується застосуванням цієї функції, показаним на рис. 7. Тут параметрами є два змінюються кута, і для обертання використовується параметрична крива.

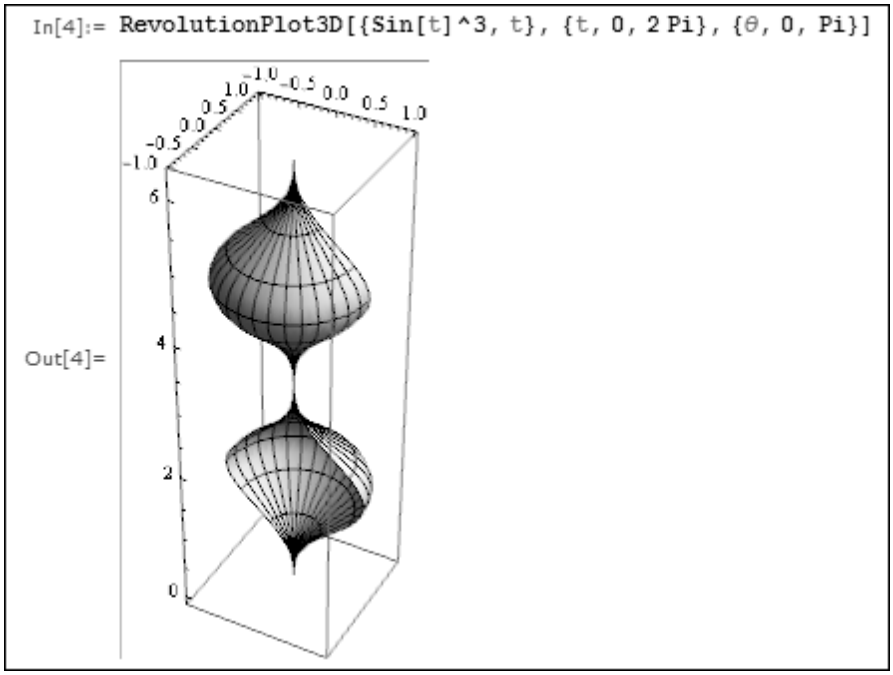


Рис. 7.

Приклад застосування опцій функції RevolutionPlot3D представлений на рис. 8. Тут крива обертання задається за допомогою шести нерівностей, з урахуванням яких будуються 6 фігур. На жаль, як і раніше, колірна забарвлення фігур відтворюється лише відтінками сірого кольору.

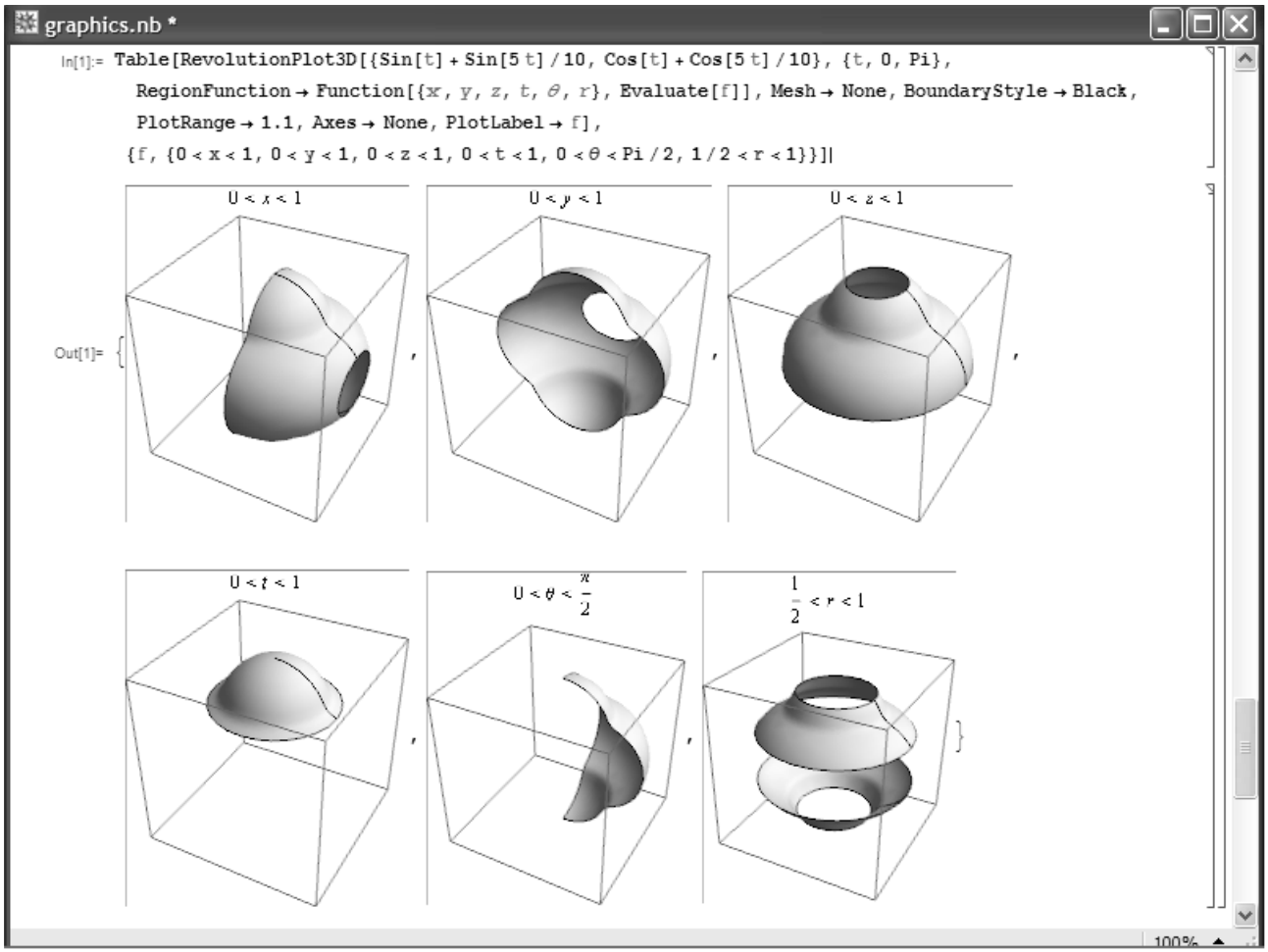


Рис. 8.

Лекція № 7. Функції розв'язання алгебраїчних рівнянь і систем рівнянь в програмі Mathematica

Аналітичні методи розв'язку алгебраїчних і трансцендентних рівнянь
Функція Solve

Для розв'язку рівнянь у аналітичному вигляді застосовується функція Solve, формат запису якої є наступним:

Solve[f,x],

де f – рівняння, яке записується у довільному вигляді,

x – ім'я змінної.

Для запису рівняння використовується символ рівності «==», наприклад $ax^2 + bx + c == 0$. Необхідно мати на увазі, що функція Solve не завжди формує рішення у найкомпактнішому виді, тому після використання цієї функції іноді потрібно використовувати функції Simplify, Expand або FullSimplify. Приклади використання функції Solve наведено у лістингу 1.

```
f1:=x^5+b*x^4-a^4*x-a^4*b==0
Solve[f1,x]
{{x->-a},{x->-ia},{x->ia},{x->a},{x->-b}}
fs:=Sin[a*x]+Cos[a*x]==0
Solve[f2,x]
{{x->-π/4a},{x->3π/4a}}
```

Лістинг 1

Слід мати на увазі, що розв'язок в аналітичному вигляді не завжди представляється системою в найбільш короткому вигляді. Для його оптимізації доцільно застосовувати команди спрощення виразів, наприклад, такі як Simplify, FullSimplify, Expand.

Функція Roots

Ця функція призначена для визначення коренів полінома, вона має вигляд:

Roots[f, x],

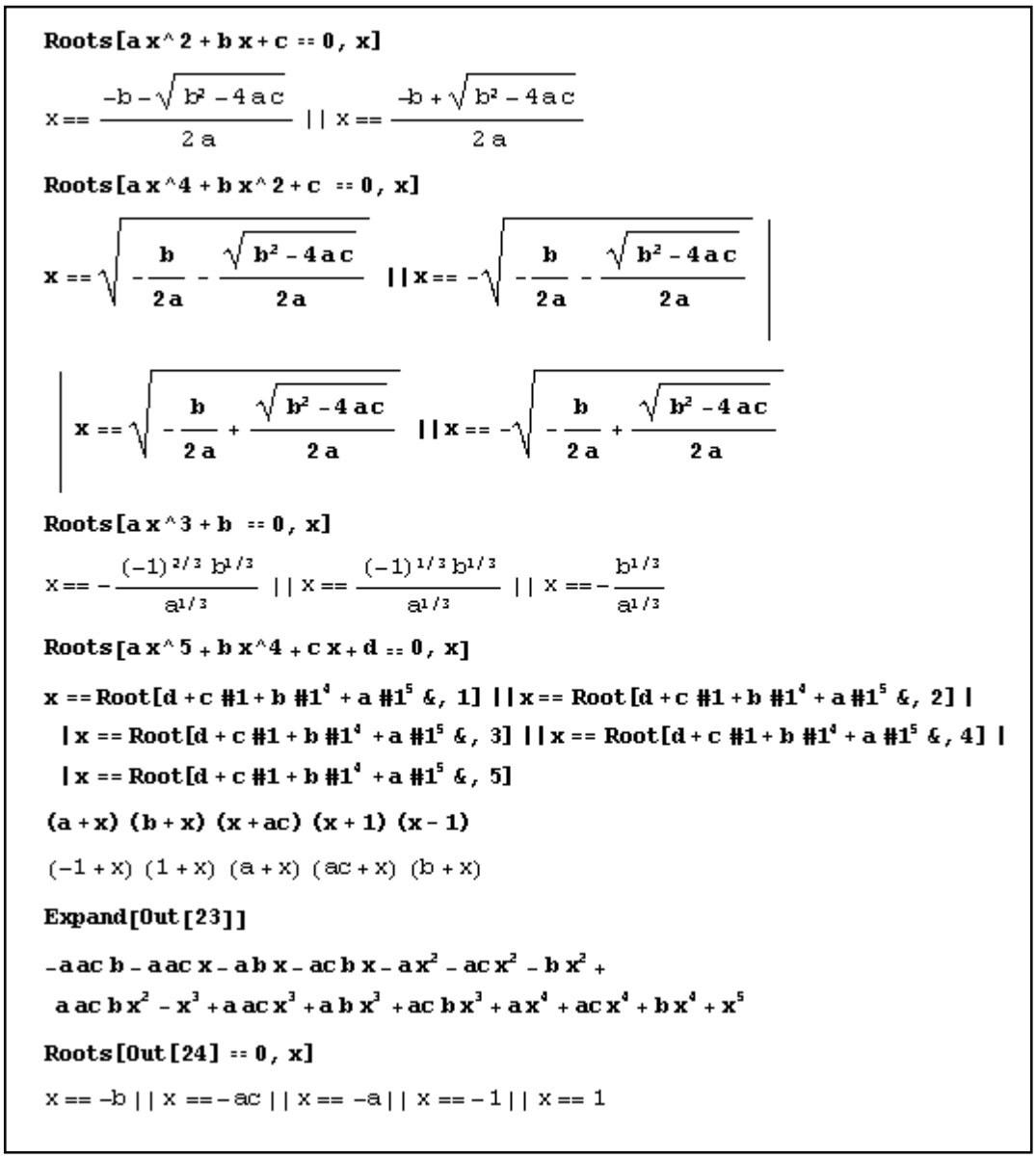
де f – поліном, корені якого необхідно знайти (може бути представлений у вигляді рівняння),

x – аргумент полінома f.

Результатом застосування функції Roots[f, x] є дійсні і комплексні корені рівняння $f(x) = 0$. При цьому розв'язок може бути отриманий в аналітичному і чисельному вигляді. Розв'язок в аналітичному вигляді в загальному випадку може бути отриманий для полінома не вище четвертого ступеня. Розв'язок $f(x) = 0$ не існує, якщо $f(x)$ – поліном п'ятого і вище ступеня. Однак якщо поліном може бути розкладений на множники, то функція Roots[f, x] знайде всі корені відповідного рівняння.

Розв'язки рівнянь у зазначених випадках показано на прикладах в лістингу 2 для рівнянь: $ax^2 + bx + c = 0$, $ax^4 + bx^2 + c = 0$, $ax^3 + b = 0$, $ax^5 + bx^4 + cx + d = 0$, $(a + x)(b + x)(x + ac)(x + 1)(x - 1)$. З лістингу 2 видно, що в перших трьох прикладах корені знайдено в аналітичному вигляді. У четвертому прикладі наведено поліном п'ятого ступеня, тому його корені не знайдені. В останньому прикладі

наведено поліном п'ятого ступеня, при цьому рішення отримано в аналітичному вигляді. Це пояснюється тим, що многочлен $f(x)$ може бути розкладений на множники. Після цього за допомогою функції Expand розкрито дужки многочлена, і знайдено всі його корені за допомогою функції Roots[Out[24], x]. Тут Out[24] – номер рядка, в якій знаходиться поліном в розкритому вигляді.



Лістинг 2

Якщо коефіцієнти полінома задані у вигляді чисел, то функція Roots[f, x] видає розв'язок у вигляді точного або наближеного значення коренів. Точне значення коренів представляється числами в раціональній формі, наближене – у формі дійсних чисел. В лістингу 3 наведено приклади вирішення таких рівнянь: $x^2 + 13/28x - 3/14 = 0$, $2x^7 + 3x^6 - 12x^2 + 7x + 5 = 0$, $x^{10} - 1 = 0$. З лістингу 3 видно, що функція визначила точне значення коренів першого рівняння, не змогла знайти в явному вигляді корені другого рівняння і знайшла корені третього рівняння, але у незручному для користувача вигляді: відсутні корені в комплексній формі. Для отримання розв'язку другого і третього рівнянь довелося

використовувати команду N[%].

```
Roots[x^2 + 13/28 x - 3/14 == 0, x]

x == 2/7 || x == -3/4

Roots[2 x^7 + 3 x^6 - 12 x^2 + 7 x + 5 == 0, x]

x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 1] |
  | x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 2] |
  | x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 3] |
  | x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 4] |
  | x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 5] |
  | x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 6] ||
   x == Root[5 + 7 #1 - 12 #1^2 + 3 #1^6 + 2 #1^7 &, 7]

N[%]

x == -0.4173393384413579` |
  | x == -1.6147152813957173` - 0.7294064649822042` ⅈ |
  | x == -1.6147152813957173` + 0.7294064649822042` ⅈ |
  | x == 0.1147195547572578` - 1.3716602453531235` ⅈ |
  | x == 0.1147195547572578` + 1.3716602453531235` ⅈ |
  | x == 0.9586653958591383` - 0.2968262106007846` ⅈ |
  | x == 0.9586653958591383` + 0.2968262106007846` ⅈ

Roots[x^10 - 1 == 0, x]

x == 1 || x == (-1)^(1/5) || x == (-1)^(2/5) |
  | x == (-1)^(3/5) || x == (-1)^(4/5) || x == -1 |
  | x == -(-1)^(1/5) || x == -(-1)^(2/5) |
  | x == -(-1)^(3/5) || x == -(-1)^(4/5)

N[%]

x == 1.` || x == 0.8090169943749475` + 0.5877852522924731` ⅈ |
  | x == 0.3090169943749474` + 0.9510565162951535` ⅈ ||
   x == -0.30901699437494734` + 0.9510565162951535` ⅈ ||
   x == -0.8090169943749473` + 0.5877852522924731` ⅈ |
  | x == -1.` |
  | x == -0.8090169943749475` - 0.5877852522924731` ⅈ ||
   x == -0.3090169943749474` - 0.9510565162951535` ⅈ ||
   x == 0.30901699437494734` - 0.9510565162951535` ⅈ ||
   x == 0.8090169943749473` - 0.5877852522924731` ⅈ
```

Лістинг 3

Застосування функції Roots[f, x] до трансцендентних рівнянь дає помилковий результат, тому для цього типу рівнянь її застосування нераціональне.

Чисельні методи розв’язку алгебраїчних і трансцендентних рівнянь

Існує велика кількість чисельних методів розв'язання алгебраїчних і трансцендентних рівнянь. Алгоритм будь-якого з цих методів є сукупністю

умов вибору початкового наближення, розрахункових співвідношень і ознаки закінчення обчислювального процесу.

Система Mathematica має багато вбудованих функцій розв'язання алгебраїчних і трансцендентних рівнянь в чисельному вигляді. Основними з них є: NSolve, NRoots, FindRoot. Розглянемо детально ці функції і наведемо приклади.

Функція NSolve

Функція NSolve має вигляд:

NSolve[f, x],

де f – вирішуване рівняння, x – шукане невідоме.

Результатом виконання цієї функції є корені рівняння $f(x)=0$. Коренями можуть бути дійсні та комплексні числа. Функція NSolve[f, x]може вирішувати всі рівняння, які вирішує функція Solve. Її відмінність лише у формі подання відповідей. Правила використання функції NSolve показані в лістингу 4 при вирішенні таких рівняннях: $x^3+9/4x^2-3/4x+5/16=0$, $2^x-4x+1=0$, $e^{-2x}+3/x-1=0$, $ax^3-1=0$, $2x^5+3.2x^3-7.3x-14=0$.

Функція NRoots

Функція NRoots має вигляд:

NRoots[f, x],

де f – вирішуване рівняння, x – шукане невідоме.

Результатом використання цієї функції є корені полінома $f(x)=0$. Правила використання функції NRoots показані в лістингу 5 при вирішенні таких рівняннях: $x^2+13/28x-3/14=0$, $2x^7+3x^6-12x^2+5=0$, $x^{10}-1=0$, $2^x-4x+1=0$. З лістингу 5 видно, що дійсні та комплексні корені знайдені в чисельному вигляді. Спроба вирішити трансцендентне рівняння $2^x-4x+1=0$ до успіху не привела – розв'язок не отримано.

Функція FindRoot

Функція FindRoot має вигляд:

FindRoot[f, {x, x_0}],

де f – рівняння, x – шукане невідоме (корінь (корені)) рівняння, x_0 – початкове наближення.

Функція FindRoot[f, {x, x_0}] знаходить корінь рівняння $f(x)=0$ з області значень x, близьких до x_0. Рекомендується наступна методика визначення коренів алгебраїчних і трансцендентних рівнянь за допомогою функції FindRoot[f, {x, x_0}]:

1. Визначення області ізоляції шуканого кореня і вибір значення наближення x_0.
2. Введення рівняння $f(x)=0$ з присвоєнням йому унікального імені.
3. Введення функції FindRoot[f, {x, x_0}] з обраним значенням x_0.
4. Отримання розв’язку натисканням комбінації клавіш <Shift>+<Enter>.

В якості приклада розглянута послідовність дій при знаходженні коренів рівняння $3^x-9x+1=0$. За графіком функції, зображеного рис. 1, обрано перші наближення для коренів функції.

```
Solve[x^3 - 2.25x^2 - 0.75x + 0.3125 == 0, x]
```

$$\{\{x \to -0.5\}, \{x \to 0.25\}, \{x \to 2.5\}\}$$

```
Solve[x^3 - 9/4 x^2 - 3/4 x + 5/16 == 0, x]
```

$$\left\{\left\{x \to -\frac{1}{2}\right\}, \left\{x \to \frac{1}{4}\right\}, \left\{x \to \frac{5}{2}\right\}\right\}$$

```
NSolve[x^3 - 9/4 x^2 - 3/4 x + 5/16 == 0, x]
```

$$\{\{x \to -0.5\}, \{x \to 0.25\}, \{x \to 2.5\}\}$$

```
Solve[2^x - 4 x + 1 == 0, x]
```

$$\left\{\left\{x \to \frac{\text{Log}[2] - 4\,\text{ProductLog}\left[-\frac{\text{Log}[2]}{2\,2^{3/4}}\right]}{4\,\text{Log}[2]}\right\}, \left\{x \to \frac{\text{Log}[2] - 4\,\text{ProductLog}\left[-1, -\frac{\text{Log}[2]}{2\,2^{3/4}}\right]}{4\,\text{Log}[2]}\right\}\right\}$$

```
NSolve[2^x - 4 x + 1 == 0, x]
```

$$\{\{x \to 0.639428\}, \{x \to 3.84666\}\}$$

```
Solve[E^(-2 x) + 3/x - 1, x]
```

$$\text{Solve}\left[-1 + e^{-2x} + \frac{3}{x}, x\right]$$

```
NSolve[E^(-2 x) + 3/x - 1 == 0, x]
```

$$\text{NSolve}\left[-1 + e^{-2x} + \frac{3}{x} == 0, x\right]$$

```
Solve[a x^3 - 1 == 0, x]
```

$$\left\{\left\{x \to \frac{1}{a^{1/3}}\right\}, \left\{x \to -\frac{(-1)^{1/3}}{a^{1/3}}\right\}, \left\{x \to \frac{(-1)^{2/3}}{a^{1/3}}\right\}\right\}$$

```
NSolve[a x^3 - 1 == 0, x]
{{x → -0.5` + 0.8660254037844387` i / a^(1/3)}, {x → -0.5` - 0.8660254037844388` i / a^(1/3)},
 {x → 1.` / a^(1/3)}}
Solve[2 x^5 - 3.2 x^3 + 7.3 x - 14 == 0, x]
{{x → -1.430088700021456` - 0.9042760720039539` i},
 {x → -1.430088700021456` + 0.9042760720039539` i},
 {x → 0.6995754953748784` - 1.0881835477460546` i},
 {x → 0.6995754953748784` + 1.0881835477460546` i},
 {x → 1.4610264092931553`}}
NSolve[2 x^5 - 3.2 x^3 + 7.3 x - 14 == 0, x]
{{x → -1.430088700021456` - 0.9042760720039539` i},
 {x → -1.430088700021456` + 0.9042760720039539` i},
 {x → 0.6995754953748784` - 1.0881835477460546` i},
 {x → 0.6995754953748784` + 1.0881835477460546` i},
 {x → 1.4610264092931553`}}
```

Лістинг 4

```
NRoots[x^2 + 13 / 28 x - 3 / 14 == 0, x]

x == -0.75 || x == 0.285714

NRoots[2 x^7 + 3 x^6 - 12 x^2 + 7 x + 5 == 0, x]

x == -1.6147152813957173` - 0.7294064649822042` i |
  | x == -1.6147152813957173` + 0.7294064649822042` i |
  | x == -0.4173393384413579` |
  | x == 0.1147195547572578` - 1.3716602453531235` i |
  | x == 0.1147195547572578` + 1.3716602453531235` i |
  | x == 0.9586653958591383` - 0.2968262106007846` i |
  | x == 0.9586653958591383` + 0.2968262106007846` i

NRoots[x^10 - 1 == 0, x]

x == -0.9999999999999999` |
  | x == -0.8090169943749475` - 0.5877852522924731` i |
  | x == -0.8090169943749475` + 0.5877852522924731` i |
  | x == -0.3090169943749474` - 0.9510565162951535` i |
  | x == -0.3090169943749474` + 0.9510565162951535` i |
  | x == 0.3090169943749473` - 0.9510565162951535` i |
  | x == 0.3090169943749473` + 0.9510565162951535` i |
  | x == 0.8090169943749476` - 0.5877852522924732` i |
  | x == 0.8090169943749476` + 0.5877852522924732` i || x == 1.`

NRoots[2^x - 4 x + 1 == 0, x]

NRoots[1 + 2^x - 4 x == 0, x]
```

Лістинг 5

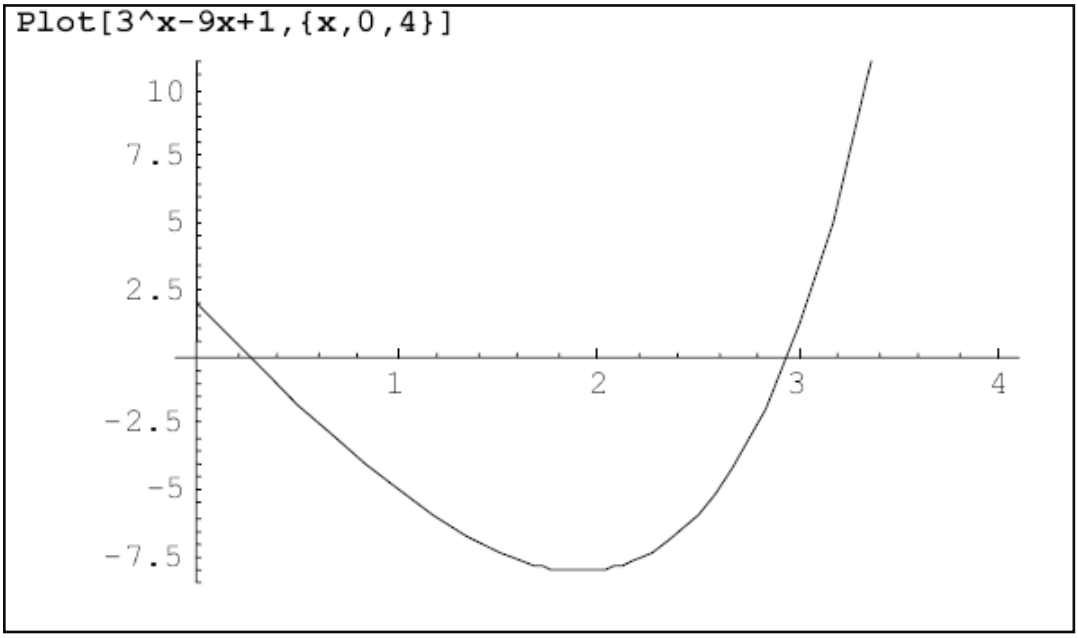


Рис. 1. Графік функції $f(x) = 3^x - 9x + 1$

З рис. 1 видно, що рівняння має два кореня, областями ізоляції яких можуть бути інтервали x_0 – [0, 1] і x_0 – [2.5, 3]. Розв'язок рівняння наведено у лістингу 6. При виборі початкового наближення потрібно бути уважним,

особливо в тих випадках, коли рівняння містить кілька коренів. Може виявитися, що при встановленому користувачем наближенні x_0 буде визначений не той корінь, який очікувався. У нашому прикладі незначна зміна x_0 призвела до іншого розв'язку: при x_0=1.9 функція FindRoot знайшла корінь x =0.258755, а вже при x_0=2 – x=2.94964.

```
F := 3^x - 9x + 1 == 0
FindRoot[F, {x, 0}]
{x → 0.258755}
FindRoot[F, {x, 3}]
{x → 2.94964}
FindRoot[F, {x, 1.9}]
{x → 0.258755}
FindRoot[F, {x, 2}]
{x → 2.94964}
FindRoot[F, {x, -50}]
{x → 0.258755}
FindRoot[F, {x, 20}]
{x → 6.37646}
```

Лістинг 6

Методи розв'язування систем рівнянь в системі Mathematica

Система Mathematica має багаті можливості розв'язування систем алгебраїчних рівнянь. Вбудовані функції дозволяють вирішувати системи лінійних та нелінійних рівнянь в аналітичному і чисельному вигляді. Дають можливість досить ефективно і оригінально перевіряти достовірність результатів. У процесі функціонування система видає коментарі, що дозволяють користувачеві приймати рішення щодо отриманих відповідей. Основними функціями розв'язання систем рівнянь є: Solve[F,X], Solve[F,X,Y], N[Solve[F,X]], FindRoot[F,X]. Розглянемо ці функції, опишемо технологію їх реалізації, наведемо приклади і задачі для самостійного розв'язання.

Функція Solve[F,X]

Функція Solve[F,X] дозволяє розв'язувати системи лінійних та нелінійних рівнянь в аналітичному вигляді. Вона записується наступним чином:

Solve[{f_1,f_2,...}, {x_1,x_2,...}]

def$_i$ – *i*-е рівняння, представлене в довільному вигляді, x_i – *i*-е невідоме.

Рівняння f_1, f_2, … можуть також представлятися через об'єднуючий знак &&. Приклади представлення функції Solve[F,X]:

```
Solve[{a*x^2+y==b,x+2*y==a+b},{x,y}]
Solve[a*x^2+y==b&&x+2*y==a+b,{x,y}]
```

При введенні рівнянь знак множення (*) можна замінити натисканням клавіші <Пробіл>. При вирішенні практичних завдань буває зручно, а в ряді випадків навіть доцільно, вводити рівняння окремо від функції Solve,

присвоюючи їм імена, які потім вводити в функцію Solve замість рівнянь. Наприклад:

f_1=a*x^2+y==b; f_2=x+2*y==a+b

Solve[{f_1, f_2},{x,y}]

або

Solve[f_1&&f_2,{x,y}]

Така форма запису спрощує перевірку достовірності рішення системи рівнянь.

Системи лінійних алгебраїчних рівнянь

Методика розв'язування рівнянь:

1. Введення рівнянь з унікальним іменем, яке задається за допомогою знака присвоєння (=).
2. Запис функції Solve[{f_1,f_2,...},{x,y,...}] або Solve [f_1&&f_2&&...,{x,y,...}].
3. Перевірка достовірності розв'язку системи рівнянь.

Приклади розв'язання лінійних алгебраїчних рівнянь наведені в лістингу 7.

Необхідно розв'язати наступні системи лінійних рівнянь:

$$\begin{cases} x_1 + ax_2 - bx_3 = y_1; \\ (2+a)x_1 + x_2 + cx_3 = y_2; \\ ax_1 + bx_2 + cx_3 = y_3. \end{cases} \quad \begin{cases} 3x_1 - 4x_2 + 2x_3 = 1; \\ x_1 + 7x_2 - 2x_3 = -4; \\ 2x_1 + 7x_2 + 3x_3 = 3. \end{cases} \quad \begin{cases} x_1 + 7x_2 - x_3 = 3.5; \\ -1.6x_1 + 3.7x_2 = 12; \\ x_1 + 2x_2 + 5x_3 = 7.5. \end{cases}$$

```
f1 = x1 + a * x2 + b * x3 == y1
```

$$x1 + a\,x2 + b\,x3 == y1$$

```
f2 = (2 + a) * x1 + x2 + c * x3 == y2
```

$$(2 + a)\,x1 + x2 + c\,x3 == y2$$

```
f3 = a * x1 + b * x2 + c * x3 == y3
```

$$a\,x1 + b\,x2 + c\,x3 == y3$$

```
Solve[{f1, f2, f3}, {x1, x2, x3}]
```

$$\left\{\left\{x1 \to -\frac{-c\,y1 + b\,c\,y1 - b^2\,y2 + a\,c\,y2 + b\,y3 - a\,c\,y3}{-a\,b + 2\,b^2 + a\,b^2 + c - 2\,a\,c - b\,c},\right.\right.$$
$$x2 \to -\frac{2\,c\,y1 + a\,b\,y2 - c\,y2 - 2\,b\,y3 - a\,b\,y3 + c\,y3}{-a\,b + 2\,b^2 + a\,b^2 + c - 2\,a\,c - b\,c},$$
$$\left.\left.x3 \to -\frac{a\,y1 - 2\,b\,y1 - a\,b\,y1 - a^2\,y2 + b\,y2 - y3 + 2\,a\,y3 + a^2\,y3}{-a\,b + 2\,b^2 + a\,b^2 + c - 2\,a\,c - b\,c}\right\}\right\}$$

```
f3 = 3 x1 - 4 x2 + 2 x3 == 1
```

$$3\,x1 - 4\,x2 + 2\,x3 == 1$$

```
f4 = x1 + 7 x2 - 2 x3 == -4
```

$$x1 + 7\,x2 - 2\,x3 == -4$$

```
f5 = 2 x1 + 7 x2 + 3 x3 == 3
```

$$2\,x1 + 7\,x2 + 3\,x3 == 3$$

```
Solve[f3 && f4 && f5, {x1, x2, x3}]
```

$$\left\{\left\{x1 \to -\frac{87}{119},\ x2 \to -\frac{3}{119},\ x3 \to \frac{184}{119}\right\}\right\}$$

```
Solve[{x1 + 7 x2 - x3 == 3.5, -1.6 x1 + 3.7 x2 == 12,
  x1 + 2 x2 + 5 x3 == 7.5}, {x1, x2, x3}]
{{x1 → -4.31818, x2 → 1.37592, x3 → 1.81327}}
```

Лістинг 7

З лістингу 7 видно, що система розв'язала першу систему рівнянь в аналітичному вигляді, другу – у вигляді точних значень невідомих, представлених в раціональній формі. Третя система рівнянь також розв'язана, але розв'язок представлено у вигляді дійсних чисел. Це пояснюється тим, що система рівнянь точного розв'язку не має. Таким чином, функція Solve може розв'язувати системи рівнянь також в чисельному вигляді. З прикладу видно, що розв'язок представляється у вигляді підстановок: x_1->, x_2->, x_3->. Це не дає можливості перевірити достовірність розв'язку, а також використовувати значення x_1, x_2, x_3 в подальших розрахунках.

Для перевірки достовірності розв'язку користувач повинен представити невідомі в явному вигляді з присвоєнням їм імені: x_1=-4.31818, x_2=1.37592, x3=1.81327, потім використовувати їх за призначенням. Отримати рішення в явному вигляді можна за допомогою виразу виду {x_1, x_2, x_3}/., яке ставиться перед функцією Solve: {x_1, x_2, x_3}/.Solve {f_1, f_2, f_3}, {x_1, x_2, x_3}]. Тепер x_1,x_2,x_3 можна використовувати за призначенням, у тому числі і для перевірки достовірності розв'язку системи рівнянь.

Системи нелінійних алгебраїчних рівнянь

Методика розв'язання систем нелінійних рівнянь та ж сама, що і лінійних. Це демонструється на прикладі розв’язку таких систем нелінійних рівнянь:

$$\begin{cases} x + ay = b; \\ x + by^2 = a + b. \end{cases} \qquad \begin{cases} xy = a; \\ x^2 + y^2 = ab. \end{cases}$$

Запис розв’язку та перевірки його достовірності наведено в лістингу 8.

```
f1 = x + a y == b
```

$$x + a y == b$$

```
f2 = x + b y^2 == a + b
```

$$x + b y^2 == a + b$$

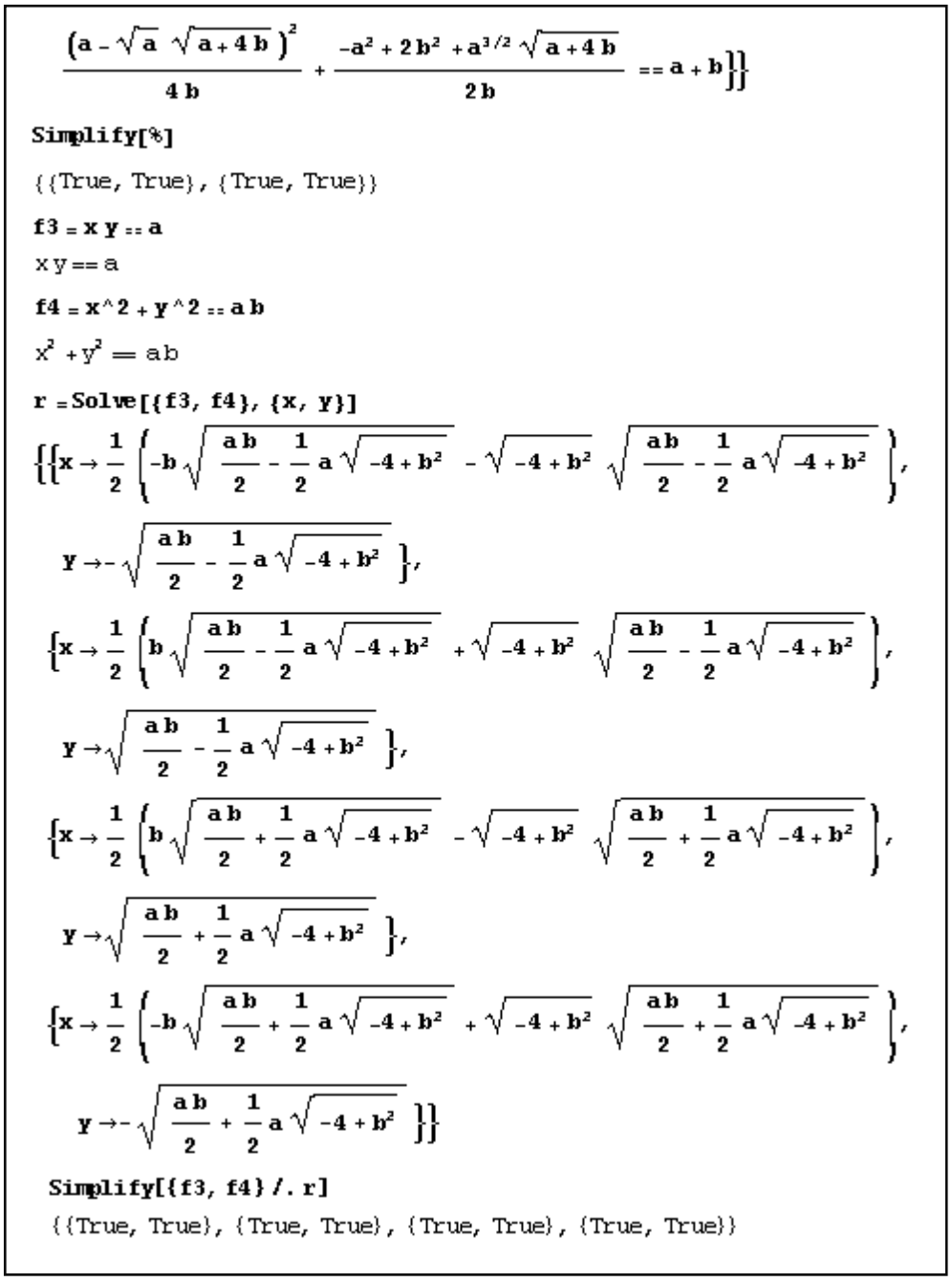
```
w = Solve[{f1, f2}, {x, y}]
```

$$\left\{\left\{x \to \frac{-a^2 + 2b^2 - a^{3/2}\sqrt{a+4b}}{2b},\ y \to \frac{a + \sqrt{a}\sqrt{a+4b}}{2b}\right\},\right.$$
$$\left.\left\{x \to \frac{-a^2 + 2b^2 + a^{3/2}\sqrt{a+4b}}{2b},\ y \to \frac{a - \sqrt{a}\sqrt{a+4b}}{2b}\right\}\right\}$$
$$\left\{\left\{x \to \frac{-a^2 + 2b^2 - a^{3/2}\sqrt{a+4b}}{2b},\ y \to \frac{a + \sqrt{a}\sqrt{a+4b}}{2b}\right\},\right.$$
$$\left.\left\{x \to \frac{-a^2 + 2b^2 + a^{3/2}\sqrt{a+4b}}{2b},\ y \to \frac{a - \sqrt{a}\sqrt{a+4b}}{2b}\right\}\right\}$$

```
{f1, f2} /. w
```

$$\left\{\left\{\frac{a\left(a + \sqrt{a}\sqrt{a+4b}\right)}{2b} + \frac{-a^2 + 2b^2 - a^{3/2}\sqrt{a+4b}}{2b} == b,\right.\right.$$
$$\left.\frac{\left(a + \sqrt{a}\sqrt{a+4b}\right)^2}{4b} + \frac{-a^2 + 2b^2 - a^{3/2}\sqrt{a+4b}}{2b} == a + b\right\},$$
$$\left\{\frac{a\left(a - \sqrt{a}\sqrt{a+4b}\right)}{2b} + \frac{-a^2 + 2b^2 + a^{3/2}\sqrt{a+4b}}{2b} == b,\right.$$

Лістинг 8 (Частина 1 з 2)



```
(a - Sqrt[a] Sqrt[a + 4 b])^2/(4 b) + (-a^2 + 2 b^2 + a^(3/2) Sqrt[a + 4 b])/(2 b) == a + b}}

Simplify[%]

{{True, True}, {True, True}}

f3 = x y == a

x y == a

f4 = x^2 + y^2 == a b

x^2 + y^2 == a b

r = Solve[{f3, f4}, {x, y}]

{{{x -> 1/2 (-b Sqrt[a b/2 - 1/2 a Sqrt[-4 + b^2]] - Sqrt[-4 + b^2] Sqrt[a b/2 - 1/2 a Sqrt[-4 + b^2]])},
  y -> -Sqrt[a b/2 - 1/2 a Sqrt[-4 + b^2]]},
 {{x -> 1/2 (b Sqrt[a b/2 - 1/2 a Sqrt[-4 + b^2]] + Sqrt[-4 + b^2] Sqrt[a b/2 - 1/2 a Sqrt[-4 + b^2]])},
  y -> Sqrt[a b/2 - 1/2 a Sqrt[-4 + b^2]]},
 {{x -> 1/2 (b Sqrt[a b/2 + 1/2 a Sqrt[-4 + b^2]] - Sqrt[-4 + b^2] Sqrt[a b/2 + 1/2 a Sqrt[-4 + b^2]])},
  y -> Sqrt[a b/2 + 1/2 a Sqrt[-4 + b^2]]},
 {{x -> 1/2 (-b Sqrt[a b/2 + 1/2 a Sqrt[-4 + b^2]] + Sqrt[-4 + b^2] Sqrt[a b/2 + 1/2 a Sqrt[-4 + b^2]])},
  y -> -Sqrt[a b/2 + 1/2 a Sqrt[-4 + b^2]]}}

Simplify[{f3, f4} /. r]

{{True, True}, {True, True}, {True, True}, {True, True}}
```

Лістинг 8 (Частина 2 з 2)

Функція Solve[F,X,Y]

Функція Solve[F,X,Y], так само, як і функція Solve[F,X], дозволяє розв'язувати системи лінійних та нелінійних рівнянь в аналітичному вигляді, але тільки з обмеженням: розв'язування здійснюються по змінним X і виключаються по змінним Y. Наприклад, функція Solve[{x+2*y-a==3, 2*x+y^2+b==7}, x, y] визначить X і виключить з розв'язку Y. Використання функції Solve[F,X,Y] демонструється в лістингу 9 при вирішенні таких систем рівнянь:

$$\begin{cases} ax^2 + bxy = 1; \\ xy + c = 7. \end{cases} \qquad \begin{cases} 7a + 3b + c = 1; \\ -5a + 12b = 3; \\ a + b + 2c = -7. \end{cases}$$

Першу систему розв'язано відносно x, другу – відносно a, потім відносно a і b.

```
f1 = a x^2 + b x y == 1
```
$$ax^2 + b\,x\,y == 1$$
```
f2 = x y + c == 7
```
$$c + x\,y == 7$$
```
z = Solve[{f1, f2}, x, y]
```
$$\left\{\left\{x \to -\frac{\sqrt{1-7b+bc}}{\sqrt{a}}\right\}, \left\{x \to \frac{\sqrt{1-7b+bc}}{\sqrt{a}}\right\}\right\}$$
```
x /. z
```
$$\left\{-\frac{\sqrt{1-7b+bc}}{\sqrt{a}}, \frac{\sqrt{1-7b+bc}}{\sqrt{a}}\right\}$$
```
f3 = 7 a + 3 b + c == 1
```
$$7a + 3b + c == 1$$
```
f4 = -5 a + 12 b == 3
```
$$-5a + 12\,b == 3$$
```
f5 = a + b + 2 c == -7
```
$$a + b + 2c == -7$$
```
s = Solve[{f3, f4, f5}, {a, b, c}]
```
$$\left\{\left\{a \to \frac{93}{181}, b \to \frac{84}{181}, c \to -\frac{722}{181}\right\}\right\}$$
```
s1 = Solve[{f3, f4, f5}, a, {b, c}]
```
$$\left\{\left\{a \to \frac{93}{181}\right\}\right\}$$
```
a /. s1
```
$$\left\{\frac{93}{181}\right\}$$
```
s2 = Solve[{f3, f4, f5}, {a, b}, c]
```
$$\left\{\left\{a \to \frac{93}{181}, b \to \frac{84}{181}\right\}\right\}$$
```
{a, b} /. s2
```
$$\left\{\left\{\frac{93}{181}, \frac{84}{181}\right\}\right\}$$

Лістинг 9

Функція NSolve[F,X]

Функція NSolve[F,X] дозволяє розв'язувати системи лінійних та нелінійних рівнянь в чисельному вигляді. Вона записується наступним чином:

Solve[{f_1, f_2,...}, {x_1, x_2,...}],

де f_i – *i*-е рівняння, представлене в довільному вигляді, x_i – *i*-е невідоме.

Рівняння f_1, f_2, … можуть також бути представлені через об'єднуючий знак &&. Методика розв'язання систем рівнянь за допомогою функції NSolve[F,X] практично не відрізняється від технології вирішення за допомогою функції Solve{F,X}. За її допомогою в лістингу 10 вирішуються наступні системи рівнянь:

$$\begin{cases} 2x_1 + 7x_2 - x_3 = 5; \\ x_1 - 2x_2 + 5x_3 = 2; \\ 4x_1 + x_2 + 3x_3 = -7. \end{cases} \qquad \begin{cases} 2y + 3x^2 = 5; \\ x + 7y^2 = 7.5. \end{cases} \qquad \begin{cases} \sin x_1 - x_2 = 1.3 \\ \cos x_2 - x_1 = -0.82. \end{cases}$$

З лістингу 10 видно, що функція NSolve[F,X] не вирішила останню систему рівнянь. Це пояснюється тим, що система складається з

трансцендентних рівнянь. Для її вирішення необхідні методи, які нереалізовані у функції NSolve[F,X].

```
f1 = 2 x1 + 7 x2 - x3 == 5
2 x1 + 7 x2 - x3 == 5
f2 = x1 - 2 x2 + 5 x3 == 2
x1 - 2 x2 + 5 x3 == 2
f3 = 4 x1 + x2 + 3 x3 == -7
4 x1 + x2 + 3 x3 == -7
s = NSolve[{f1, f2, f3}, {x1, x2, x3}]
{{x1 → -3.75, x2 → 2.06818, x3 → 1.97727}}
{f1, f2, f3} /. s
{{True, True, True}}
f1 = 2 y + 3 x^2 == 5
3 x^2 + 2 y == 5
f2 = x + 7 y^2 == 7.5
x + 7 y^2 == 7.5
r = NSolve[{f1, f2}, {x, y}]
{{x → 1.5110628030619`, y → -0.9249661921959316`},
 {x → -0.9661774325244309`, y → 1.0997517533207495`},
 {x → 1.0123543818272729`, y → 0.962707908392686`},
 {x → -1.5572397523647399`, y → -1.1374934695175007`}}
{f1, f2} /. r
{{True, True}, {True, True}, {True, True}, {True, True}}
f3 = Sin[x1] - x2 == 1.3
-x2 + Sin[x1] == 1.3
f4 = Cos[x2] - x1 == -0.82
-x1 + Cos[x2] == -0.82
r1 = NSolve[{f3, f4}, {x1, x2}]
NSolve[{-x2 + Sin[x1] == 1.3, -x1 + Cos[x2] == -0.82}, {x1, x2}]
```

Лістинг 10

Функція FindRoot[F,{X,x_0}]

Функція FindRoot[F, {X, x0}] розв'язує системи лінійних та нелінійних рівнянь чисельними методами ітерацій. Для її реалізації необхідно знати початкові наближення невідомих. Функція має вигляд:

FindRoot [{f_1, f_2, ...}, {x_1, x_{10}}, {x_2, x_{20}}, ...],

Де f_i – *i*-е рівняння, представлене в довільному вигляді, x_i – *i*-е невідоме,
x_{i0} – початкове наближення *i*-го невідомого.

Рівняння f_1, f_2, … можуть також представлятися через об'єднувальний знак &&.

Методика розв'язання систем рівнянь за допомогою функції FindRoot[F, {X, x0}] істотно відрізняється від технології розв'язання рівнянь за допомогою

функції NSolve[F,X]. Відмінність полягає в необхідності визначення початкових наближень. Приклади розв'язку систем рівнянь функцією FindRoot[F,{X,x0}] наведено в лістингу 11. В лістингу 11 розв’язуються такі системи рівнянь:

$$\begin{cases} \sin x_1 - x_2 = 1.3; \\ \cos x_2 - x_1 = -0.82; \\ x_{10} = 1.8; \\ x_{20} = -0.35. \end{cases} \qquad \begin{cases} tg(y_1 y_2 + 0.2) = y_1^2; \\ 0.5 y_1^2 + 2 y_2^2 = 1; \\ y_{10} = 0.9; \\ y_{20} = 0.5. \end{cases}$$

```
f1 = Sin[x1] - x2 == 1.3
-x2 + Sin[x1] == 1.3
f2 = Cos[x2] - x1 == -0.82
-x1 + Cos[x2] == -0.82
r = FindRoot[{f1, f2}, {x1, 1.8}, {x2, -0.35}]
{x1 → 1.76935, x2 → -0.319646}
{f1, f2} /. r
{False, False}
x1 := 1.76935; x2 := -0.319646
Sin[x1] - x2
1.3
Cos[x2] - x1
-0.820003
{x1, x2} /. r
{1.76935, -0.319646}
f3 = Tan[y1 y2 + 0.2] == y1^2
Tan[0.2 + y1 y2] == y1^2
f4 = 0.5 y1^2 + 2 y2^2 == 1
0.5 y1^2 + 2 y2^2 == 1
z = FindRoot[{f3, f4}, {y1, 0.9}, {y2, 0.5}]
{y1 → 0.910994, y2 → 0.540854}
{y1, y2} /. z
{0.910994, 0.540854}
y1 := 0.910994; y2 := 0.540854
Tan[y1 y2 + 0.2]
0.82991
y1^2
0.82991
0.5 y1^2 + 2 y2^2
1.
```

Лістинг 11

Функція Eliminate[F,x]

Функція Eliminate[F,x] призначена для скорочення числа рівнянь системи шляхом виключення заданих змінних x. Вона має вигляд:

Eliminate[{f_1, f_2, ...}, x],

де f_i – *i*-е рівняння, представлене в довільному вигляді,

x_i – *i*-е невідоме, яке підлягає виключенню.

Рівняння f_1, f_2, ... можуть також представлятися через об'єднувальний знак &&.

Ця функція здійснює перетворення вихідної системи рівнянь так, що число рівнянь і змінних скорочується. Граничним є одне рівняння з одним невідомим. Приклад використання функції Eliminate наведено в лістингу 12, в якому система з трьох рівнянь почергово зводиться до системи з двох рівнянь, а потім до одного рівняння.

$$\begin{cases} x^2 + ay^2 - z = 2; \\ x + y - bz = 7; \\ 2x + 3y + 9z = 1. \end{cases}$$

```
f1 = x^2 + a y^2 - z == 2
x^2 + a y^2 - z == 2
f2 = x + y - b z == 7
x + y - b z == 7
f3 = 2 x + 3 y + 9 z == 1
2 x + 3 y + 9 z == 1
Eliminate[{f1, f2, f3}, z]
9 a y^2 == 19 - 2 x - 9 x^2 - 3 y && b (-1 + 2 x + 3 y) == 63 - 9 x - 9 y
Eliminate[{f1, f2, f3}, {y, z}]
b^2 (-18 + a - 4 a x + 9 x^2 + 4 a x^2) + b (-48 + 126 a - 3 x - 270 a x + 54 x^2 + 36 a x^2) ==
 -18 - 3969 a + 9 x + 1134 a x - 81 x^2 - 81 a x^2
FullSimplify[Out[5]]
a (9 (-7 + x) + b (-1 + 2 x))^2 + 3 (3 + b) (2 + x (-1 + 9 x) + 3 b (-2 + x^2)) == 0
```

Лістинг 12

Матричні методи розв'язання систем лінійних рівнянь

Систему лінійних алгебраїчних рівнянь можна представити в наступному вигляді: A*X=B, де A – матриця коефіцієнтів, X – вектор невідомих, B – вектор вільних членів (правих частин) системи рівнянь. Методика розв'язання рівнянь в системі Mathematica проста і полягає в наступному:

1. Введення матриці коефіцієнтів з присвоєнням їй імені, наприклад A.
2. Введення вектора невідомих з ім'ям X.
3. Введення вектора вільних членів з ім'ям *B*.
4. Створення виразу *z = A. X ==B.*
5. Введення функції *Solve [z, X]*.

Приклад розв’язання системи лінійних рівнянь

$$\begin{cases} 2x_1 + 3x_2 - 7x_3 = 1; \\ -3x_1 + x_2 + 5x_3 = 7.5; \\ 5x_1 + 3x_3 = 2.5, \end{cases}$$

матричним методом показана у лістингу 13.

```
A = {{2, 3, -7}, {-3, 1, 5}, {5, 0, 3}}

{{2, 3, -7}, {-3, 1, 5}, {5, 0, 3}}

X = {x1, x2, x3}

{x1, x2, x3}

B = {1, 7.5, 2.5}

{1, 7.5, 2.5}

Z = A.X == B

{2 x1 + 3 x2 - 7 x3, -3 x1 + x2 + 5 x3, 5 x1 + 3 x3} == {1, 7.5, 2.5}

Solve[%]

{{x1 → -0.0664336, x2 → 2.58042, x3 → 0.944056}}

{{2, 3, -7}, {-3, 1, 5}, {5, 0, 3}}.{x1, x2, x3} == {1, 7.5, 2.5}

{2 x1 + 3 x2 - 7 x3, -3 x1 + x2 + 5 x3, 5 x1 + 3 x3} == {1, 7.5, 2.5}

Solve[%]

{{x1 → -0.0664336, x2 → 2.58042, x3 → 0.944056}}
```

Лістинг 13

Крім наведеного існують в системі Mathematica наступні два матричні способи розв'язання систем алгебраїчних рівнянь.

Спосіб 1. Визначення вектора невідомих *X* за формулою: $X=A^{-1}B$.

При цьому операція множення записується функцією *Dot*, а операція інвертування матриці *A*-функцією *Inverse*. Тоді рішення записується в наступному вигляді:

X: = Dot [Inverse [A], B]

Спосіб 2. Застосування функції *LinearSolve*.

Функція *LinearSolve* записується в наступному вигляді:

X: = LinearSolve [A, B]

У лістингу 14 наведено приклад вирішення системи лінійних рівнянь

$$\begin{cases} a_1x_1 + 2x_2 - 3x_3 = b_1; \\ -7x_1 + cx_2 + x_3 = b_2; \\ x_1 + x_2 + dx_3 = b_3, \end{cases}$$

двома способами.

Особливі випадки вирішення систем рівнянь

Система рівнянь може мати число розв’язків, рівне числу невідомих, така система називається *сумісною*. Якщо число рівнянь нескінченно велике, то систему називають *сумісною* та *невизначеною*. Якщо ж система не має жодного розв’язку, то її називають *несумісною*.

```
A = {{a, 2, -3}, {-7, c, 1}, {1, 1, d}}
{{a, 2, -3}, {-7, c, 1}, {1, 1, d}}
B = {b1, b2, b3}
{b1, b2, b3}
X = Dot[Inverse[A], B]
```

$$\left\{\frac{b3\,(2+3c)}{23-a+3c+14d+acd}+\frac{b2\,(-3-2d)}{23-a+3c+14d+acd}+\frac{b1\,(-1+cd)}{23-a+3c+14d+acd},\right.$$
$$\frac{(21-a)\,b3}{23-a+3c+14d+acd}+\frac{b1\,(1+7d)}{23-a+3c+14d+acd}+\frac{b2\,(3+ad)}{23-a+3c+14d+acd},$$
$$\left.\frac{(2-a)\,b2}{23-a+3c+14d+acd}+\frac{b1\,(-7-c)}{23-a+3c+14d+acd}+\frac{b3\,(14+ac)}{23-a+3c+14d+acd}\right\}$$

```
Simplify[%]
```

$$\left\{\frac{b3\,(2+3c)-b2\,(3+2d)+b1\,(-1+cd)}{23+3c+14d+a\,(-1+cd)},\ \frac{b1+3b2+21b3-ab3+7b1d+ab2d}{23-a+3c+14d+acd},\right.$$
$$\left.\frac{-(-2+a)\,b2-b1\,(7+c)+b3\,(14+ac)}{23+3c+14d+a\,(-1+cd)}\right\}$$

```
X = LinearSolve[A, B]
```

$$\left\{\frac{-b1-3b2+2b3+3b3c-2b2d+b1cd}{23-a+3c+14d+acd},\ \frac{b1+3b2+21b3-ab3+7b1d+ab2d}{23-a+3c+14d+acd},\right.$$
$$\left.\frac{-7b1+2b2-ab2+14b3-b1c+ab3c}{23-a+3c+14d+acd}\right\}$$

Лістинг 14

Дві наступні системи рівнянь демонструють особливі випадки при розв'язку систем лінійних рівнянь:

$$\begin{cases} x_1+2x_2-x_3=1; \\ 2x_1+4x_2-2x_3=2; \\ 3x_1+2x_2+3x_3=5. \end{cases} \qquad \begin{cases} x_1+2x_2-x_3=1; \\ 2x_1+4x_2-2x_3=2; \\ 3x_1+6x_2-3x_3=3. \end{cases}$$

Їх розв'язок наведено у лістингу 15.

```
Solve[{x1 + 2 x2 - x3 == 1, 2 x1 + 4 x2 - 2 x3 == 2, 3 x1 + 2 x2 + 3 x3 == 5}, {x1, x2, x3}]
```

$$\left\{\left\{x1 \to 2-2\,x3,\ x2 \to -\frac{1}{2}+\frac{3\,x3}{2}\right\}\right\}$$

```
Solve[{x1 + 2 x2 - x3 == 1, 2 x1 + 4 x2 - 2 x3 == 2, 3 x1 + 6 x2 + -3 x3}, {x1, x2, x3}]
Solve[{x1 + 2 x2 - x3 == 1, 2 x1 + 4 x2 - 2 x3 == 2, 3 x1 + 6 x2 - 3 x3}, {x1, x2, x3}]
Det[{{1, 2, -1}, {2, 4, -2}, {3, 2, 3}}]
0
Det[{{1, 2, -1}, {2, 4, -2}, {3, 6, -3}}]
0
```

Лістинг 15

З лістингу 15 видно, що перша система рівнянь має розв'язок: $x_1 = 2 - 2x_3, x_2 = -1/2 + 3/2x_3$, тобто має нескінченне число розв'язків (при будь-якому значенні x_3). Система сумісна, але невизначена. Друга система не має жодного розв'язку - вона несумісна. Зазначимо, що в обох випадках головний визначник системи дорівнює нулю.

Лекція № 8. Функції математичного аналізу

Обчислення сум і добутків рядів

Обчислення сум рядів може здійснюватись в аналітичному або числовому вигляді. Для обчислення сум в аналітичному вигляді використовується функція Sum. Існують наступні форматФи запису функції Sum:

- Sum[f_i, {i, i_{max}}];
- Sum[f_i, {i, i_{min}, i_{max}}];
- Sum[f_i, {i, i_{min}, i_{max}, Δi}];
- Sum[$f_{i,j,..}$ {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}],

де f – елемент сумування,
i – змінна сумування,
i_{min}, i_{max} – елементи сумування,
Δi – крок зміни аргументу i.

Якщо необхідно розрахувати суму членів ряду, який представлений аналітичною функцією, від 1 до n, необхідно використовувати функцію Sum[f_i, {i, i_{max}}]. В лістингу 1 наведено використання функції для обчислення рядів 1/n, $1/n^2$, $1/n^3$.

```
Sum[n,{n,n}]
1/2n(1+n)
Sum[n^2,{n,n}]
1/6n(1+n)(1+2n)
Sum[n^3,{n,n}]
1/4n^2(1+n)^2
Sum[n,{n,1000}]
500500
Sum[n,{n,1000000}]
500000500000
```

Лістинг 1. Використання функції Sum[f_i, {i, i_{max}}]

Функція Sum[f_i, {i, i_{min}, i_{max}}] обчислює суму значень функції f в діапазоні значень аргументу i_{min} ..i_{max} з кроком 1. При використанні функції Sum[f_i, {i, i_{min}, i_{max}, Δi}] крок зміни аргументу задається параметром Δi.

Функція Sum[$f_{i,j,..}$ {i, i_{min}, i_{max}}, {j, j_{min}, j_{max}}] обчислює суму за декількома змінними. Приклад її використання ілюструється при обчисленні наступних рядів: $\sum_{i=1}^{50}\sum_{j=1}^{10}(x_i^2+y_j^2)$, $\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\frac{x^n}{n!}\frac{y^m}{m!}$, $\sum_{n=1}^{3}\sum_{m=1}^{3}\frac{x^n}{n!}\frac{y^m}{m!}$ показано у лістингу 2.

```
Sum[x^2+y^2,{x,1,50},{y,1,10}]
448500
Sum[(x^n/n!)*(y^m/m!),{n,0,∞},{m,0,∞}]
e^(n+m)
Sum[(x^n/n!)*(y^m/m!),{n,1,3},{m,1,3}]
xy+x^2y/2+x^3y/6+xy^2/2+x^2y^2/4+x^3y^2/12+xy^3/6+x^2y^3/12+x^3y^3/36
```

Лістинг 2

Чисельне обчислення сум виконується функцією NSum, яка має такі ж модифікації, як і функція Sum.

Обчислення добутків здійснюється аналогічно сумуванню. Для цього використовуються функції:

- Product – для обчислення добутків в аналітичному і чисельному вигляді;
- NProduct - для обчислення добутків тільки в чисельному вигляді.

Обчислення границі функції

Обчислення границі функції в системі Mathematica здійснюється за допомогою функції Limit. Синтаксис її запису такий:

Limit[f(x), x->x_0].

Функція Limit має опцію Direction. Опція Direction вказує на напрям наближення до границі. Її запис має два варіанти:

Direction -> +1;

Direction -> -1.

Значення +1 свідчить про наближення до границі з лівого боку, -1 – з правого боку. Приклад використання функції Limit показано у лістингу 3.

```
Limit[(1-Exp[-2*t])/Log[1-3*t],t->0]
-2/3
Limit[ArcTan[1/x],x->0,Direction->+1]
-Pi/2
Limit[ArcTan[1/x],x->0,Direction->-1]
Pi/2
```

Лістинг 3

Розклад функцій у степеневий ряд

Для розкладу в степеневий ряд в системі Mathematica використовуються наступні функції:

Series[f,{x,x0,n}] – розкладає функцію f в околі точки x = x_0 при використанні n членів ряду;

Series[f(x,y),{x,x0,nx}, {y,y0,ny}] – розкладає функцію f по двом змінним x і y в околі точки (x0, y0) з числом членів nx і ny відповідно.

Приклади використання функцій наведено у лістингу 4.

```
Series[Exp[x],{x,0,7}]
1 + x + x^2/2 + x^3/6 + x^4/24 + x^5/120 + x^6/720 + x^7/5040 +O[x]^8
Series[Exp[x+y],{x,0,3},{y,0,3}]
(1 + y + y^2/2 + y^3/6 + O[y]^4) + (1 + y + y^2/2 + y^3/6 + O[y]^4)*x + (1/2 + y/2 + y^2/4 + y^3/12 + O[y]^4)*x^2 +
(1/6 + y/6 + y^2/12 + y^3/36 + O[y]^4)*x^3 +O[x]^4
```

Лістинг 4

Розрахунок похідних функцій

Розрахунок похідних здійснюють за допомогою використання таких функцій:

D[f,x];

D[f,{x,n}];

D[f,x_1,x_2,…];

Dt[f,x];

Dt[f];

Derivative[n_1,n_2,..][f],

де f – диференційована функція,

x – змінна диференціювання,

x_1, x_2,.. – змінні диференціювання,

n – порядок похідної.

Методи обчислення інтегралів

Аналітичні методи

Інтеграл в аналітичному вигляді обчислюється за допомогою наступних вбудованих функцій :

- $Integrate\ [f(x), x]$- обчислює невизначений інтеграл функції $f(x)$ за аргументом x ;
- $Integrate\ [\ f(\ x\ )\ , \{\ x, x_н\ , x_к\}]$ - обчислює визначений інтеграл функції $f(\ x\ )$ за змінною x з нижньою $x_н$ і верхньою $x_к$ межами інтегрування. Межами інтегрування можуть бути символьні змінні, числа і навіть функції;
- $Integrate\ [\ f(\ x\ , y, \dots\ )\ , \{\ x\ , x_н, x_к\ \}\ , \{y\ , y_н\ \ , y_k, \dots\}]$ – Обчислює визначений інтеграл функції багатьох змінних $x\ , y, \dots$ з межами інтегрування $x_н$, $x_к$, $y_н$, $y_к$.

Приклади обчислення інтегралів наведено у лістингу 5. Обчислюються такі інтеграли:

$$\int \frac{ax-1}{bx+1} dx, \int_a^b \frac{2+x}{x} dx, \int_a^b (1 + 2xy + 4x^2y^2)\, dxdy\ .$$

```
f1:= (a*x - 1) / (b*x + 1)
Integrate[f1, x]
```

$$\frac{a\,x}{b} + \frac{(-a - b)\ Log[1 + b\,x]}{b^2}$$

```
f2:= (2 + x) / x
Integrate[f2, {x, a, b}]
-a + b - 2 Log[a] + 2 Log[b]
f3:= 1 + 2*x*y + 4*x^2*y^2
Integrate[f3, {x, a, b}, {y, a, b}]
```

$$a^2 + \frac{a^4}{2} + \frac{4\,a^6}{9} - 2\,a\,b + b^2 - a^2\,b^2 - \frac{8\,a^3\,b^3}{9} + \frac{b^4}{2} + \frac{4\,b^6}{9}$$

Лістинг 5

У попередньому прикладі межами інтегрування були символьні змінні a і b. У лістингу 6 показано обчислення інтегралів тих же функцій у випадку, коли межами інтегрування є числа : x_H = 1, x_K = 5, y_H = 0, y_K = 10.

```
Integrate[(2 + x) / x, {x, 1, 5}]
2 (2 + Log[5])
N[%]
7.21888
Integrate[1 + 2 x y + 4 x^2 y^2, {x, 1, 5}, {y, 0, 10}]
```

$$\frac{507160}{9}$$

```
N[%]
56351.1
```

Лістинг 6

З лістингу 6 видно, що в даному випадку отримані у вигляді точних рішень. Числові значення інтегралів отримано за допомогою функції N(%).

Чисельні методи

Обчислення інтегралів у чисельному вигляді необхідно в наступних випадках:

- первісна не виражається через елементарні функції;
- підінтегральна функція задана у вигляді таблиці;
- аналітичний вираз первісної занадто складний.

Як приклад у лістингу 7 наведені обчислення невизначених і визначених інтегралів від функцій: $y(x) = x^{\frac{1}{x}} e^x u\ y(x) = x^{20} e^{-x}$.

```
Integrate[x^(1 / x) * Exp[x], x]
∫ e^x x^(1/x) dx
NIntegrate[x^(1 / x) * Exp[x], {x, 1, 5}]
204.435
Integrate[x^20 Exp[-x], x]
Integrate[x^20 *Exp[-x],x]
  Integrate[x^20 *Exp[-x],x]
 e^-x (-2432902008176640000 - 2432902008176640000 x -
    1216451004088320000 x^2 - 405483668029440000 x^3 -
    101370917007360000 x^4 - 20274183401472000 x^5 -
    3379030566912000 x^6 - 482718652416000 x^7 -
    60339831552000 x^8 - 6704425728000 x^9 - 670442572800 x^10 -
    60949324800 x^11 - 5079110400 x^12 - 390700800 x^13 -
    27907200 x^14 - 1860480 x^15 - 116280 x^16 - 6840 x^17
    - 380 x^18 - 20 x^19 - x^20)
NIntegrate[x^20Exp[-x], {x, 1, 2}]
14860.3
```

Лістинг 7

З лістингу 7 видно, що перший інтеграл аналітично не вирішується, а другий є занадто громіздким. В цих випадках результат обчислення інтегралів чисельними методами можна отримати за допомогою функції NIntegrate.

Формат функції NIntegrate є таким:

NIntegrate$(f(x), \{x, x_H, x_K\})$,

де *f(x)* - підінтегральна функція ;

x - аргумент підінтегральної функції ;

x_H, x_K - нижня і верхня межі інтегрування.

Методика використання функції NIntegrate не відрізняється від методики обчислення визначеного інтеграла в аналітичному вигляді.

Обчислення кратних інтегралів

Обчислення кратних інтегралів в системі Mathematica здійснюється шляхом багаторазового застосування функції Integrate при аналітичному інтегруванні або NIntegrate при чисельному інтегруванні. У лістингу 8 наведено приклади обчислення кратних інтегралів від функції $f(x) = \frac{x-1}{x+1}$ при різних варіантах меж інтегрування:

- символьних змінних від a до b;
- чисельних значеннях меж інтегрування від 0 до 2;
- меж інтегрування , заданих функціями: ln 2 і $e^{1.2}$.

```
Integrate[Integrate[(x - 1) / (x + 1), x], {x, a, b}]
```

$$\frac{1}{2}\left(-4a - a^2 + 4b + b^2 + 4\,\text{Log}[1+a] + 4a\,\text{Log}[1+a] - 4\,\text{Log}[1+b] - 4b\,\text{Log}[1+b]\right)$$

$$\int_a^b \int (x-1)/(x+1)\,dx\,dx$$

$$\frac{1}{2}\left(-4a - a^2 + 4b + b^2 + 4\,\text{Log}[1+a] + 4a\,\text{Log}[1+a] - 4\,\text{Log}[1+b] - 4b\,\text{Log}[1+b]\right)$$

$$\int_a^b \int\int (x-1)/(x+1)\,dx\,dx\,dx$$

$$\frac{1}{6}\left(-6a - 9a^2 - a^3 + 6b + 9b^2 + b^3 + 6\,\text{Log}[1+a] + 12a\,\text{Log}[1+a] + 6a^2\,\text{Log}[1+a] - 6\,\text{Log}[1+b] - 12b\,\text{Log}[1+b] - 6b^2\,\text{Log}[1+b]\right)$$

$$\int_0^2 \int\int (x-1)/(x+1)\,dx\,dx\,dx$$

$$\frac{1}{3}\left(28 - 27\,\text{Log}[3]\right)$$

```
N[%]
-0.554177
```

$$\int_{\text{Log}[2]}^{\text{Exp}[1.2]} \int\int (x-1)/(x+1)\,dx\,dx\,dx$$

```
-1.31499
```

Лістинг 8

Обчислення невласних інтегралів

Система Mathematica дозволяє обчислювати інтеграли з нескінченними границями. При цьому використовуються ті ж функції, що й у випадку обчислення інтегралів з кінцевими межами. Значення нескінченності позначається або символом ∞ або службовою константою Infinity. Приклади обчислення невласних інтегралів наведено у лістину 9.

```
Integrate[Sin[x] / x, {x, 0, Infinity}]
```

$$\frac{\pi}{2}$$

```
Integrate[(Sinh[x] + Cosh[x]) / Exp[x^2], {x, 0, Infinity}]
```

$$\frac{1}{2} e^{1/4} \sqrt{\pi} \left(1 + \text{Erf}\left[\frac{1}{2}\right]\right)$$

```
N[%]
```

```
1.73023
```

```
Integrate[(x + 1) / (x^3 + 1), {x, 0, Infinity}]
```

$$\frac{\pi + i \left(\text{Log}\left[3 - i\sqrt{3}\right] - \text{Log}\left[3 + i\sqrt{3}\right]\right)}{\sqrt{3}}$$

```
FullSimplify[%]
```

$$\frac{4\pi}{3\sqrt{3}}$$

```
Integrate[Exp[x] / (a + Exp[x])^2, {x, -Infinity, Infinity}]
```

$$\frac{1}{a}$$

```
Integrate[(x + 1) / (2 x + 1), {x, 0, Infinity}]
```

$$\int_0^\infty \frac{1 + x}{1 + 2x} \, dx$$

```
NIntegrate[1 / (2 x^2 + 1), {x, 0, Infinity}]
```

```
1.11072
```

Лістинг 9

З лістингу 9 можна зробити наступні висновки:

- вирішення невласного інтеграла отримуємо в аналітичному вигляді. Для отримання чисельного значення інтеграла застосовується команда N [%];
- вираз первісної функції в аналітичному вигляді може бути складним. Для його спрощення слід скористатися функціями Simplify , Expand, Factor або їх прототипами;
- якщо інтеграл не має первісної, то результатом буде початковий вираз інтеграла.

Табличне інтегрування

Підінтегральна функція f(x) може бути задана у вигляді таблиці. Найчастіше це необхідно при проведенні експериментальних досліджень. У таких випадках обчислення інтеграла можна здійснити за формулами прямокутників, трапецій або парабол . Рішення можна отримати також шляхом інтерполяції функції f (x) з подальшим її інтегруванням. Система Mathematica має наступні вбудовані функції, що дозволяють виконати табличне інтегрування :

- ListIntegrate[$\{y_1, y_2, \dots, y_n\}, h$];
- ListIntegrate[$\{y_1, y_2, \dots, y_n\}, h, k$];

- ListIntegrate [{ $\{x_1, y_1\}$, $\{x_2, y_2\}$, …, $\{x_n, y_n\}$ } , k].

У функціях використовуються наступні позначення:

- y_i - значення функції y = f(x) в x_i - мувузлі , 1 = 1 , 2 , ... , n;
- h - крок зміни аргументу;
- k - число точок кожного з підінтегралів.

Як приклад обчислимо значення інтеграла табличної функції y = f(x), значення якої наведені в табл. 1.

Таблиця 1. Табличне представлення функції y = f(x)

x	1	2	3	4	5	6	7	8
y	1	8	27	64	125	216	343	512

У даному випадку крок інтегрування постійний і дорівнює h = 1. З табл. 1 видно, що аналітичний вираз функції має вигляд y = x^3. У лістингу10 наведені процедури обчислення інтеграла в разі задання підінтегральної функції у вигляді таблиці і у вигляді аналітичного виразу y = x^3 .

```
f = {1, 8, 27, 64, 125, 216, 343, 512}

{1, 8, 27, 64, 125, 216, 343, 512}

ListIntegrate[f, 1]
 4095
  4
Integrate[x^3, {x, 1, 8}]


 4095
  4
ListIntegrate[{{1,1},{2,8},{3,27},
   {4,64},{5,125},{6,216},{7,343},
   {8,512}}]
 4095
  4
```

Лістинг 10

Розв’язання диференційних рівнянь у середовищі Mathematica

Система Mathematica дозволяє вирішувати в аналітичному і чисельному вигляді лінійні і нелінійні диференційні рівняння та системи. Розв’язок диференційних рівнянь та систем здійснюється за допомогою вбудованих функцій.

Аналітичні методи

Аналітичні методи розв'язання диференційних рівнянь в системі Mathematica реалізуються за допомогою двох вбудованих функцій:

DSolve $\left[f, y\left[x\right], x\right]$;

DSolve $\left[\left\{ f_1, f_2, ...\right\}, \left\{y\left[x_1\right], y\left[x_2\right], ...\right\}, \left\{x_1, x_2, ...\right\}\right]$.

Розглянемо докладно ці функції і наведемо приклади.

Функція *DSolve* $\left[f, y\left[x\right], x\right]$ призначена для розв’язку диференційного рівняння f щодо функції $y(x)$ з аргументом x. Функція дає загальний розв’язок рівняння з постійними інтегрування, які позначаються $c[i]$.

Диференційне рівняння представляється у довільному вигляді. Наприклад, $y'==2x\wedge 2-1,\ y'-2x\wedge 2==-1$ або $y'-2\,x\wedge 2+1==0$. Функція дозволяє вирішувати диференційне рівняння будь-якого порядку. Наведемо приклади. Дано наступні диференційні рівняння:

$y' = 2x^2 + 3x - 1,\ y' = 2\ln x + x - 2, y' = 3e^{-2x} - x^2 + x - 1.$

Розв'язок рівнянь наведено у лістингу 11.

```
F = y'[x] == 2 x^2 + 3 x - 1

y'[x] == -1 + 3 x + 2 x^2

DSolve[F, y[x], x]

{{y[x] → -x + 3 x^2/2 + 2 x^3/3 + C[1]}}

DSolve[y'[x] == 2 Log[x] + x - 2, y[x], x]

{{y[x] → -4 x + x^2/2 + C[1] + 2 x Log[x]}}

DSolve[y'[x] - 3 E^(-2 x) + x^2 - x + 1 == 0, y[x], x]

{{y[x] → -3/2 E^(-2 x) - x + x^2/2 - x^3/3 + C[1]}}
```

Лістинг 11

З лістингу 11 видно, що при розв'язанні першого рівняння йому присвоєно ім'я *F* і здійснено введення поза функцією *DSolve*. При вирішенні другого рівняння останнє введено безпосередньо у функцію *DSolve* . При вирішенні третього рівняння воно представлено у вигляді, відмінному від перших двох. З прикладу видно, що програма знайшла спільний розв'язок з довільними постійними інтегрування, а розв'язок зведено до простого інтегрування правої частини рівнянь. Функція *DSolve* дозволяє також вирішувати диференційні рівняння високого порядку, що демонструється на наступних прикладах:

$y''' = 3x^2 - 2x + 1,\ y''' = y'(x) - 5y(x) + x2 - 1.$

Розв'язок наведено у лістингу 12.

```
DSolve[y'''[x] == 3 x^2 - 2 x + 1, y[x], x]
```

$$\left\{\left\{y[x] \to \frac{x^3}{6} - \frac{x^4}{12} + \frac{x^5}{20} + C[1] + x\,C[2] + x^2\,C[3]\right\}\right\}$$

```
F1 = y'''[x] == y'[x] - 5 y[x] + x^2 - 1
```

$$y^{(3)}[x] == -1 + x^2 - 5\,y[x] + y'[x]$$

```
DSolve[F1, y[x], x]
```

$$\left\{\left\{y[x] \to \frac{1}{125}\left(-23 + 10\,x + 25\,x^2\right) + e^{\#1\,\text{Root}[5-\#1+\#1^3\&,1]}\,C[1] + e^{\#1\,\text{Root}[5-\#1+\#1^3\&,2]}\,C[2] + e^{\#1\,\text{Root}[5-\#1+\#1^3\&,3]}\,C[3]\right\}\right\}$$

```
N[%]
```

$$\{\{y[x] \to 0.008`\left(-23.` + 10.`\,x + 25.`\,x^2\right) + 2.718281828459045`^{-1.9041608591349206`\,\#1}\,C[1] + 2.718281828459045`^{(0.9520804295674603`-1.3112480440771224`\,i)\,\#1}\,C[2] + 2.718281828459045`^{(0.9520804295674603`+1.3112480440771224`\,i)\,\#1}\,C[3]\}\}$$

Лістинг 12

Розв’язок диференційних рівнянь за умови відомих початкових умов

Для отримання розв’язку диференційних рівнянь за умови відомих початкових умов використовується функція наступна модифікація функції DSolve:

$$DSolve\left[f(x,x_0),\ y\left[x\right],\ x\right],$$

де $f(x,x_0)$ - диференційне рівняння в сукупності з початковими умовами;

$y(x)$ - шукана функція;

x - незалежна змінна.

Розглянемо приклади окремого розв’язку диференційних рівнянь за допомогою функції *DSolve*:

$y'(x) = 2\ln x + x - 2$ при початковій умові: $y(0) = 1$;

$y''(x) = 3y(x) - e^{2x} + x - 1$ при $y(0) = 1, y'(0) = 0$;

$y''(x) = -5y(x) - 1$ при $y(1) = y'(1) = y''(1) = 0$.

Розв’язок наведено у лістингу 13.

```
DSolve[{y'[x]==2 Log[x]+x-2,y[0]==1},y[x],x]
```

$$\left\{\left\{y[x] \to \frac{1}{2}\left(2-8x+x^2+4x\,\text{Log}[x]\right)\right\}\right\}$$

```
DSolve[{y''[x]==3 y[x]-E^(2 x)+x-1,y[0]==1,y'[0]==0},y[x],x]
```

$$\left\{\left\{y[x] \to -\left(e^{-\sqrt{3}\,x}\left(15-7\sqrt{3}+6\,e^{\sqrt{3}\,x}+15\,e^{2\sqrt{3}\,x}+7\sqrt{3}\,e^{2\sqrt{3}\,x}-18\,e^{2x+\sqrt{3}\,x}-6\,e^{\sqrt{3}\,x}x\right)\right)\Big/ \left(18\left(-2+\sqrt{3}\right)\left(2+\sqrt{3}\right)\right)\right\}\right\}$$

```
Simplify[%]
```

$$\left\{\left\{y[x] \to \frac{1}{18}e^{-\sqrt{3}\,x}\left(15-7\sqrt{3}+\left(15+7\sqrt{3}\right)e^{2\sqrt{3}\,x}-18\,e^{\left(2+\sqrt{3}\right)x}-6\,e^{\sqrt{3}\,x}(-1+x)\right)\right\}\right\}$$

```
DSolve[{y'''[x] == -5 y[x] - 1, y[1] == 0, y'[1] == 0, y''[1] == 0}, y[x], x]
```

$$\left\{\left\{y[x] \to \left(e^{-\frac{5^{1/3}}{2}-5^{1/3}x}\left(e^{\frac{3\,5^{1/3}}{2}}\text{Cos}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]^2-3\,e^{\frac{5^{1/3}}{2}+5^{1/3}x}\text{Cos}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]^2+\right.\right.\right.$$

$$2\,e^{\frac{3}{2}5^{1/3}x}\text{Cos}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]\text{Cos}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}x\right]+e^{\frac{3\,5^{1/3}}{2}}\text{Sin}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]^2-$$

$$3\,e^{\frac{5^{1/3}}{2}+5^{1/3}x}\text{Sin}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]^2+$$

$$\left.\left.\left.2\,e^{\frac{3}{2}5^{1/3}x}\text{Sin}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]\text{Sin}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}x\right]\right)\right)\Big/\left(15\left(\text{Cos}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]^2+\right.\right.\right.$$

$$\left.\left.\left.\text{Sin}\left[\frac{1}{2}\sqrt{3}\,5^{1/3}\right]^2\right)\right)\right\}\right\}$$

Лістинг 13

Розв’язок систем диференційних рівнянь в аналітичному вигляді

Системи диференційних рівнянь в аналітичному вигляді вирішуються також за допомогою вбудованої функції *DSolve* , яка в даному випадку має вигляд:

$$DSolve\,[\{f_1, f_2,...\},\{y_1(x), y_2(x),...\}, x]$$

де f_i - і-те рівняння системи;

$y_i(x)$ - і-те шукане невідоме;

x - незалежна змінна.

Приклади використання функції *DSolve* для випадку розв’язання систем диференційних рівнянь наведено нижче:

$$\begin{cases} x'(t) = y(t) + z(t) \\ y'(t) = x(t) + 3z(t) \\ z'(t) = x(t) + y(t). \end{cases}$$

Розв’язок наведено у лістингу 14.

```
f = {x'[t] == y[t] + z[t], y'[t] == x[t] + z[t], z'[t] == x[t] + y[t]}
{x'[t] == y[t] + z[t], y'[t] == x[t] + z[t], z'[t] == x[t] + y[t]}
DSolve[f, {x[t], y[t], z[t]}, t]
{{x[t] → 1/3 e^-t (2 + e^3 t) C[1] + 1/3 e^-t (-1 + e^3 t) C[2] + 1/3 e^-t (-1 + e^3 t) C[3],
  y[t] → 1/3 e^-t (-1 + e^3 t) C[1] + 1/3 e^-t (2 + e^3 t) C[2] + 1/3 e^-t (-1 + e^3 t) C[3],
  z[t] → 1/3 e^-t (-1 + e^3 t) C[1] + 1/3 e^-t (-1 + e^3 t) C[2] + 1/3 e^-t (2 + e^3 t) C[3]}}
```

Лістинг 14

З лістингу 14 видно, що розв'язок отримано в загальному вигляді, так як початкові умови не були задані.

За умови підстановки початкових умов: $x(0)=1, y(0)=z(0)=0$ запис функції *DSolve* є наступним:

$$DSolve[\{x'[t]==y[t]+z[t], y'[t]==x[t]+z[t], z'[t]=x[t]+y[t],$$
$$x[0]=1, y[0]==z[0]==0\}, \{x[t], y[t], z[t]\}, t].$$

Чисельні методи розв'язку диференційних рівнянь

Чисельні методи розв'язку диференційних рівнянь у системі Mathematica реалізуються за допомогою наступних двох вбудованих функцій:

$$NDSolve\ [f, y[x], \{x, x_{\min}, x_{\max}\}]$$

$$NDSolve\ [\{f_1, f_2, ..., y_1(x_0), y_2(x_0), ...\}, \{y_1[x], y_2[x], ...\}, \{x, x_{\min}, x_{\max}\}]$$

де f - диференційне рівняння і початкові умови;

f_i - і-те рівняння системи диференційних рівнянь;

$y[x]$ - шукана функція;

$y_i[x]$- і-та шукана функція системи диференційних рівнянь;

$y_i(x_0)$ - і-та початкова умова;

$x_{\min}, x_{\max}$ - мінімальне і максимальне значення незалежної змінної;

x - аргумент шуканої функції.

Функції чисельного розв'язку диференційних рівнянь і систем мають опцію *StartingStepSize*, яка визначає величину початкового кроку інтегрування.

Чисельні методи найбільш часто застосовуються в тих випадках, коли рівняння в аналітичному вигляді системою не вирішується або зовсім не має аналітичного розв'язку. Такими є більшість нелінійних рівнянь. Далі детально викладаються вбудовані функції, способи їх реалізації та наводяться приклади.

Функція $NDSolve\ [f, y[x], \{x, x_{\min}, x_{\max}\}]$

Ця функція вирішує диференційне рівняння *n*-го порядку, обчислюючи шукану функцію *y(x)* в діапазоні незалежної змінної *x* від $x_{\min}$ до $x_{\max}$. Розв'язок можна отримати у вигляді таблиці або графіка.

Методику розв'язання задачі показано на прикладі такого диференційного рівняння:

$$y'(t) - xy(x) = 1,$$

при початкових умовах $y(0)=1$. Розв'язок слід отримати в табличному і графічному вигляді в діапазоні від 0 до 5 з кроком 0.5.

Методика вирішення диференційного рівняння з допомогою функції *NDSolve* складається з виконання наступних операцій:

1. Введення функції *NDSolve*, яка в нашому прикладі має вигляд:

$NDSolve[\{y'[x]==xy[x]+1, y[0]==1\}, y[x], \{x,0,5\}]$.

2. Отримання розв'язку шляхом одночасного натискання клавіш <Shilit> + <Enter>. Розв'язок буде отримано у вигляді повідомлення без виведення самого розв'язку на екран.

3. Введення функції *Table* для отримання розв'язку в табличній формі. У нашому прикладі ця функція матиме вигляд:

$Table[\{x, y[x]/.Out[8]\}, \{x,0,5,0.5\}]$.

Тут *Out [8]* - це 8 рядок, в якому знаходиться розв'язок рівняння. Результатом є вектор, представлений у вигляді рядка.

4. Введення функції *TableForm [%]* для отримання розв'язку у формі таблиці. Результатом є функція *y(x)*, представлена у вигляді таблиці.

5. Введення функції $Plot[\{y[x]/.Out[8]\}, \{x,0,5\}]$. Результатом є графік функції.

Розв'язок завдання наведено у лістингу 15. Спочатку наведено розв'язання рівняння в аналітичному вигляді.

У наступному прикладі необхідно вирішити рівняння вищого порядку:

$$y'''(x)-3x^2y''(x)-2xy'(x)-y'(x)=1,$$

при початкових умовах $y(1)=1, y'(1)=y''(1)=0$. Розв'язок отримати в діапазоні *x* від 0 до 3 з кроком 0.5 при поданні рішення в табличному вигляді та в діапазоні від 0 до 2 - у графічному. Розв'язок наведено у лістингу 16.

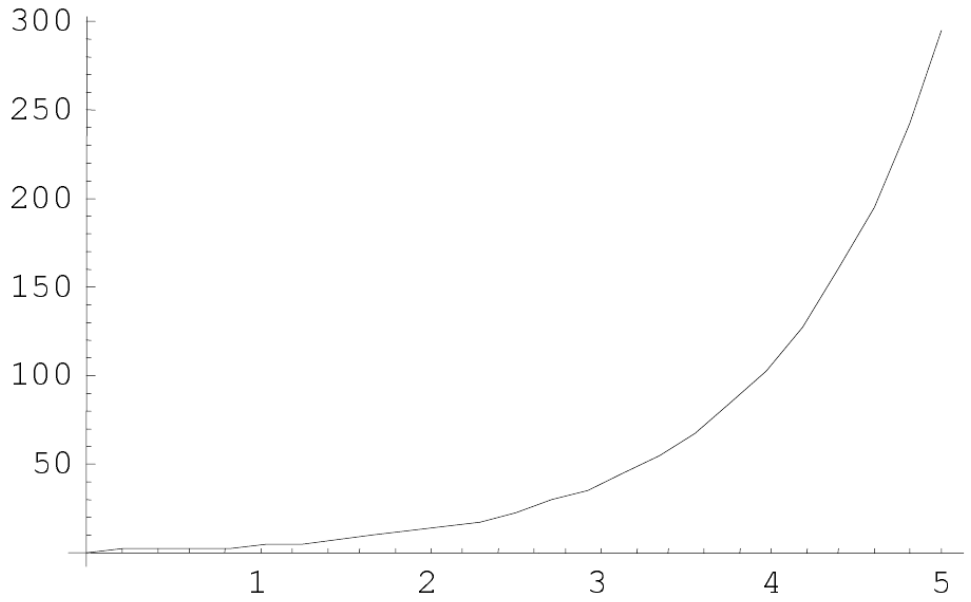
```
DSolve[{y'[x]==x* y[x]+1,y[0]==1},y[x],x]
```

$$\left\{\left\{y[x]\to \frac{1}{2} e^{\frac{x^2}{2}} \left(2+\sqrt{2\pi}\, \text{Erf}\left[\frac{x}{\sqrt{2}}\right]\right)\right\}\right\}$$

```
NDSolve[{y'[x]==y[x]+1,y[0]==1},y[x],{x,0,5}]
{{y[x]->InterpolatingFunction[{{0.,5.}},<>][x]}}
Table[{x,y[x]/.Out[8]},{x,0,5,0.5}]
{{0,{1.}},{0.5,{2.29744}},{1.,{4.43656}},{1.5,{7.9
6337}},{2.,{13.7781}},{2.5,{23.365}},{3.,{39.171}}
,{3.5,{65.2308}},{4.,{108.196}},{4.5,{179.034}},{5
.,{295.826}}}
TableForm[%]
```

0	1.
0.5	2.29744
1.	4.43656
1.5	7.96337
2.	13.7781
2.5	23.365
3.	39.171
3.5	65.2308
4.	108.196
4.5	179.034
5.	295.826

```
Plot[y[x]/.Out[8],{x,0,5}]
```

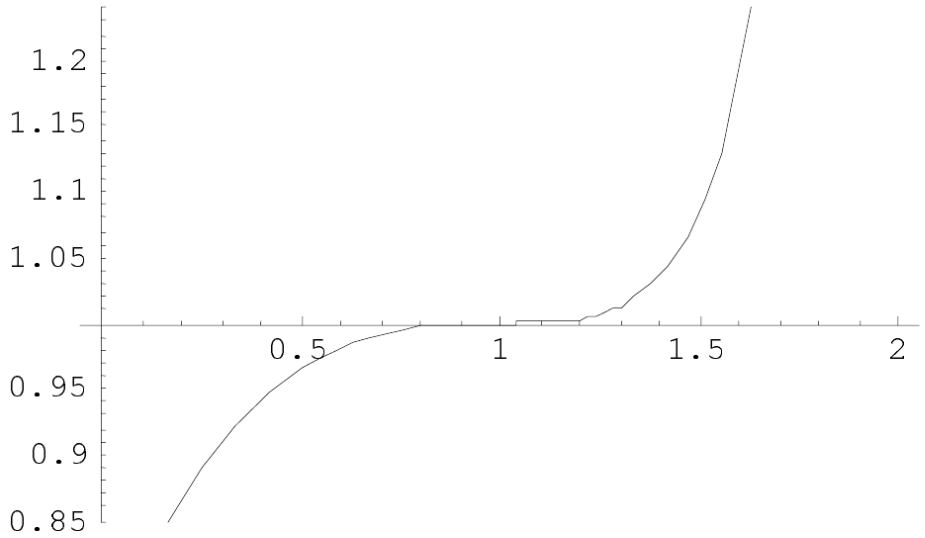


Лістинг 15

```
NDSolve[{y'''[x]==3 x^2 y''[x]+2 x
y'[x]+y[x]+1,y[1]==1,y'[1]==y''[1]==0},y[x],{x,0,5}]
 {{y[x]→InterpolatingFunction[{{0.,5.}},<>][x]}}
Table[{x,y[x]/.Out[1]},{x,0,3,0.5}]
{{0,{0.7363584435178943`}},{0.5`,{0.9662626674117316`}},
{1.`,{1.`}},{1.5`,{1.0834912468759925`}},
{2.`,{6.943463004953 99`}},{2.5`,{5233.080815734498`}},
{3.`,{2.368089715606595`*^8}}}
TableForm[%]
```

0	0.736358
0.5	0.966263
1.	1.
1.5	1.08349
2.	6.94346
2.5	5233.08
3.	2.36809×10^8

```
Plot[{y[x]/.Out[1]},{x,0,2}]
```



Лістинг 16

Функція NDSolve $[\{f_1, f_2,..., y_1(x_0), y_2(x_0),...\},\{y_1[x], y_2[x],...\},\{x, x_{\min}, x_{\max}\}]$

Ця функція вирішує систему диференційних рівнянь *n*-го порядку, обчислюючи шукані функції $y_1[x], y_2[x],...$ в діапазоні незалежної змінної x від $x_{\min}$ до $x_{\max}$. Розв'язок можна отримати у вигляді таблиці або графіків.

Методику використання функції показано на такому прикладі:

$$\begin{cases} p_0'(t) = -0.9\,p_0(t) + 2\,p_1(t) \\ p_1'(t) = 0.9\,p_0(t) - 2.9\,p_1(t) + 4\,p_2(t) \\ p_2'(t) = 0.9\,p_1(t) - 4\,p_2(t), \end{cases}$$

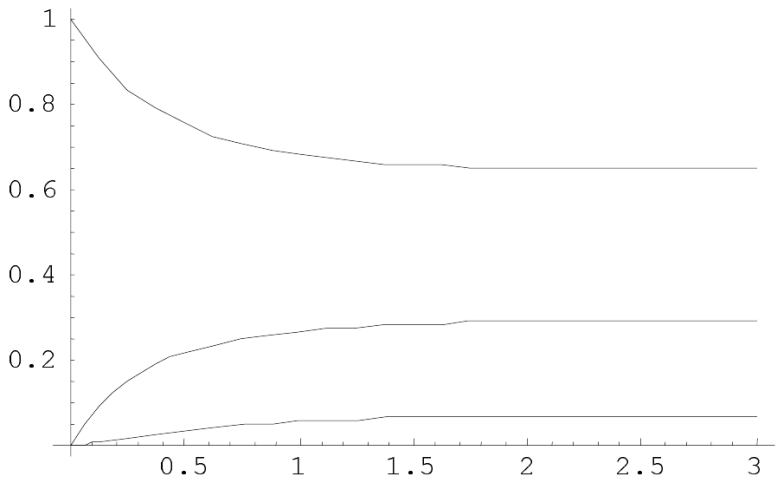
за наступних початкових умов: $p_0(0) = 1, p_1(0) = p_2(0) = 0$. Розв'язок отримати у вигляді таблиць і графіків.

У цьому випадку функція *NDSolve* буде мати вигляд:

NDSolve [{p_0 '[t]=-0.9 p_0 [t] +2p_1[t], p_1'[t]=0.9 p_0[t]-2.9p_1[t]+4p_2[t], p_2'[t]=0.9p_1[t]-4p_2[t], p_0[0]= 1,p_1[0]= p_2[0]=0}, {p_0[t], p_1[t], p_2[t]}, {t, 0, 100}].

Розв'язок наведено у лістингу 17. Таблиця представлена в діапазоні *t* від 0 до 1 з кроком 0.1, а графік - в діапазоні *t* від 0 до 3.

```
NDSolve[{p0'[t]==-0.9 p0[t]+2 p1[t],p1'[t]==0.9
p0[t]-2.9 p1[t]+4 p2[t],p2'[t]==0.9 p1[t]-4
p2[t],p0[0]==1,p1[0]==p2[0]==0},{p0[t],p1[t],p2[t]},
{t,0,100}]


{{p0[t]→InterpolatingFunction[{{0.,100.}},<>][t],p
1[t]→InterpolatingFunction[{{0.,100.}},<>][t],p2[t
]→InterpolatingFunction[{{0.,100.}},<>][t]}}

Table[{t,{p0[t],p1[t],p2[t]}/.Out[2]},{t,0,1,0.1}]

{{0, {{1., -2.38559×10^-21, -6.03826×10^-26}}},
 {0.1, {{0.921667, 0.0751894, 0.00314313}}},
 {0.2, {{0.862132, 0.127986, 0.00988277}}},
 {0.3, {{0.81629, 0.166013, 0.0176963}}},
 {0.4, {{0.780636, 0.194028, 0.0253356}}},
 {0.5, {{0.752693, 0.215063, 0.0322435}}},
 {0.6, {{0.730668, 0.231105, 0.038227}}},
 {0.7, {{0.713235, 0.243489, 0.0432764}}},
 {0.8, {{0.699393, 0.253141, 0.0474669}}},
 {0.9, {{0.688377, 0.260716, 0.0509062}}},
 {1., {{0.679598, 0.266694, 0.053708}}}}
TableForm[%]
0       1.  -2.38559×10^-21  -6.03826×10^-26
0.1     0.921667  0.0751894  0.00314313
0.2     0.862132  0.127986  0.00988277
0.3     0.81629  0.166013  0.0176963
0.4     0.780636  0.194028  0.0253356
0.5     0.752693  0.215063  0.0322435
0.6     0.730668  0.231105  0.038227
0.7     0.713235  0.243489  0.0432764
0.8     0.699393  0.253141  0.0474669
0.9     0.688377  0.260716  0.0509062
1.      0.679598  0.266694  0.053708
Plot[{{p0[t]/.Out[2]},{p1[t]/.Out[2]},{p2[t]/.Out[2]}}, {t,0,3}]
```



Лістинг 17

Система Mathematica, як і будь-яка інша система комп'ютерної алгебри, не є ідеальною щодо розв'язання диференційних рівнянь. Одержаний розв'язок рідко коли збігається з відповіддю, наявною в математичних довідниках. Не рідкі випадки, коли вбудована функція не дає розв'язку або він є помилковим, хоча рівняння досить просте. Наведемо такі приклади. Нехай потрібно вирішити наступні рівняння і системи рівнянь:

$y''(x) = -y(x) + tgx;$

$y''(x) = -y(x) + 4x\sin x;$

$x'(t) + y'(t) - tx(t) = t;$

$x'(t) + y'(t) + y(t) = t(t+2);$

$3z(x)z'(x) - 2x = 0.$

Систему рівнянь вирішити аналітичним і чисельним методами при початкових умовах, $x(0) = 1, y(0) = 0$ в діапазоні x від -1 до 1.

Розв'язки рівнянь наведено у лістингу 18.

```
DSolve[y''[x]+y[x]==Tan[x],y[x],x]
```

$$\left\{\left\{y[x] \to -2\,\text{ArcTanh}\left[\text{Tan}\left[\frac{x}{2}\right]\right]\text{Cos}[x] + C[1]\,\text{Cos}[x] + C[2]\,\text{Sin}[x]\right\}\right\}$$

```
DSolve[y''[x]+y[x]==4 x Sin[x],y[x],x]
```

$$\left\{\left\{y[x] \to C[1]\,\text{Cos}[x] + C[2]\,\text{Sin}[x] + \frac{1}{2}\left(-2x^2\,\text{Cos}[x] + \text{Cos}[x]\,\text{Cos}[2x] - 2x\,\text{Cos}[2x]\,\text{Sin}[x] + 2x\,\text{Cos}[x]\,\text{Sin}[2x] + \text{Sin}[x]\,\text{Sin}[2x]\right)\right\}\right\}$$

```
Simplify[%]
```

$$\left\{\left\{y[x] \to \left(\frac{1}{2} - x^2 + C[1]\right)\text{Cos}[x] + (x + C[2])\,\text{Sin}[x]\right\}\right\}$$

```
DSolve[{x'[t]+y'[t]-t x[t]==t,x'[t]+y'[t]+y[t]==t
(t+2),
x[0]==1,y[0]==0},{x[t],y[t]},t]
```

```
DSolve[{-t x[t] + x'[t] + y'[t] == t, y[t] + x'[t] + y'[t] == t (2 + t),
  x[0] == 1, y[0] == 0}, {x[t], y[t]}, t]
```

```
NDSolve[{x'[t]+y'[t]-t x[t]==t,x'[t]+y'[t]+y[t]==t
(t+2),
x[0]==1,y[0]==0},{x[t],y[t]},{t,-1,1}]
```

```
NDSolve[{-t x[t] + x'[t] + y'[t] == t, y[t] + x'[t] + y'[t] == t (2 + t),
  x[0] == 1, y[0] == 0}, {x[t], y[t]}, {t, -1, 1}]
```

```
DSolve[3 z[x]^2 z'[x]-2 x==0,z[x],x]
```

$$\left\{\left\{z[x] \to (x^2 + 3C[1])^{1/3}\right\}, \left\{z[x] \to -(-1)^{1/3}(x^2 + 3C[1])^{1/3}\right\}, \left\{z[x] \to (-1)^{2/3}(x^2 + 3C[1])^{1/3}\right\}\right\}$$

Лістинг 18

На основі лістингу можна дати наступні коментарі. Розв'язок першого і другого рівнянь є вірними, але не збігаються з довідковими даними, в яких наводяться такі відповіді:

$$y(x) = c_1 \cos x + c_2 \sin x - \cos x \ln(\mathrm{tg}(\pi/4 + x/2)),$$

$$y(x) = c_1 \cos x + c_2 \sin x + x(\sin x - x \cos x).$$

Отримані рішення можна істотно спростити за допомогою функції *Simplify [%]*.

Розв’язок системи рівнянь не отриманий ні аналітичним, ні чисельним методами, хоча такий розв’язок існує.

Цікавою є відповідь розв’язку третього рівняння-три рівноцінних результати.

При вирішенні диференційних рівнянь чисельними методами можуть виникати неприпустимо великі помилки за рахунок методичних помилок і помилок вибору кроку інтегрування. Завжди необхідно пам'ятати, що при комп'ютерних технологіях розв’язку диференційних рівнянь необхідна перевірка достовірності отриманих результатів.

Лекція № 9. Комп'ютерні технології інтерполяції в середовищі Mathematica

Інтерполяція, точна у вузлах

В середовищі Mathematica інтерполяція, точна у вузлах, може бути реалізована наступними методами:

- універсальним;
- за допомогою універсальних функцій InterpolatingPolynomial і Interpolation.

Універсальний метод

Універсальний метод потребує вирішення систем алгебраїчних рівнянь, які були отримані на основі даних функції $y = f(x)$, яка представлена у вигляді таблиці чи матриці.

Приклади інтерполяції універсальним методом наведені нижче.

Функція $y = f(x)$ задана у вигляді табл. 1.

Таблиця 1. Функція $y = f(x)$ в табличній формі

x	1	2	3	4
y	6,2	4,1	1,9	0,6

Необхідно розв'язати задачу інтерполяції, яка є точною у вузлах, якщо функція $y = \varphi(x)$ є поліномом. Так як кількість вузлів $n = 4$, то степінь полінома має бути не вищим за $n - 1$. Тобто $y = a_0 + a_1x + a_2x^2 + a_3x^3$.

Складемо систему рівнянь:

$$a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 + a_3 \cdot 1^3 = 6{,}2$$
$$a_0 + a_1 \cdot 2 + a_2 \cdot 2^2 + a_3 \cdot 2^3 = 4{,}1$$
$$a_0 + a_1 \cdot 3 + a_2 \cdot 3^2 + a_3 \cdot 3^3 = 1{,}9$$
$$a_0 + a_1 \cdot 4 + a_2 \cdot 4^2 + a_3 \cdot 4^3 = 0{,}6$$

Розв'язок отримаємо за допомогою функції $Solve\big(F, (a_0, a_1, a_2, a_3)\big)$, де $F -$ вихідна система рівнянь (див. лістинг 1).

```
F={a0+a1+a2+a3==6.2,
    a0+2 a1+2^2 a2+2^3 a3==4.1,
    a0+3 a1+3^2 a2+3^3 a3==1.9,
    a0+4 a1+4^2 a2+4^3 a3==0.6}
Solve[F,{a0, a1,a2,a3}]
 {a0+a1+a2+a3==6.2,a0+2 a1+4 a2+8 a3==4.1,a0+3 a1+9
a2+27 a3==1.9,a0+4 a1+16 a2+64 a3==0.6}
 {{a0→7.2,a1→-0.116667,a2→-1.05,a3→0.166667}}
```

Лістинг 1

У результаті отриманого розв'язку інтерполяційна формула буде мати вигляд:

$$y = 7{,}2 - 0.116667x - 1{,}05x^2 + 0{,}166667x^3.$$

Задачу інтерполяції, точної у вузлах також можна вирішити за допомогою матричного методу розв'язку системи лінійних рівнянь. Перевірка правильності розв'язку задачі виконують табуляцією отриманої формули й порівнянням результатів з вихідними даними, наприклад за допомогою функції $Table[f(x),\{x,x_H,x_K,h\}]$, де $f(x)-$ функція, що підлягає табуляції, $x-$аргумент функції $f(x)$, x_H,x_K- початкове й кінцеве значення аргументу, $h-$ крок таблиці, якщо $h=1$, то ним можна знехтувати.

У попередньому прикладі число рівнянь і число невідомих однакове. При розв'язуванні практичних задач майже завжди кількість значень табульованої функції перевищує степінь алгебраїчних рівнянь. У таких випадках обмежене число вузлів інтерполяції доводиться обирати з усього діапазону вихідних даних. Проілюструємо це на прикладі табульованої функції, навденої у табл. 2.

Таблиця 2. Табульована функція

x	3	6	9	12	15	18	21	24	27	30
y	26	90	180	300	500	700	1000	1200	1500	2000

Нехай функція інтерполяції є поліномом другого степеня $y=a_0+a_1x+a_2x^2$. Розрахунок коефіцієнтів інтерполяційного полінома наведено у лістингу 2. В результаті порівняння результатів інтерополяції з вихідними даними можна зробити висновок, про наявність похибки інтерполяції при деяких значеннях аргументу x. Для детального аналізу величини похибки на рис. 1 побудовано функції інтерполяції і даних.

```
F={a0+6 a1+6^2 a2==90,
    a0+18 a1+18^2 a2==700,
    a0+27 a1+27^2 a2==1500}
Solve[F,{a0,a1,a2}]
 {a0+6 a1+36 a2==90,a0+18 a1+324 a2==700,a0+27 a1+729
a2==1500}
```

$$\left\{\left\{a0\to -\frac{135}{7},\ a1\to \frac{925}{126},\ a2\to \frac{685}{378}\right\}\right\}$$

```
 N[%]
 {{a0->-19.2857,a1->7.34127,a2->1.81217}}


 y=-135/7+925/126 x+685/378 x^2
Table[y,{x,3,30,3}]
```

$$-\frac{135}{7}+\frac{925\,x}{126}+\frac{685\,x^2}{378}$$

$$\left\{\frac{400}{21},\ 90,\ \frac{1355}{7},\ \frac{6925}{21},\ \frac{3490}{7},\ 700,\ \frac{19615}{21},\ \frac{8405}{7},\ 1500,\ \frac{38470}{21}\right\}$$

```
 N[%]

{19.0476,90.,193.571,329.762,498.571,700.,934.048,1200.
71,1500.,1831.9}
```

Лістинг 2

```
f1={{3,26},{6,90},{9,180},{12,300},
    {15,500},{18,700},{21,1000},{24,1200},
    {27,1500},{30,2000}}
r1=ListPlot[f1,AxesLabel→{"x","y"},PlotStyle→PointSize[
0.02]]
 r2=Plot[-135/7+925/126 x+685/378 x^2,{x,2,40}]
Show[{r1,r2}]
```

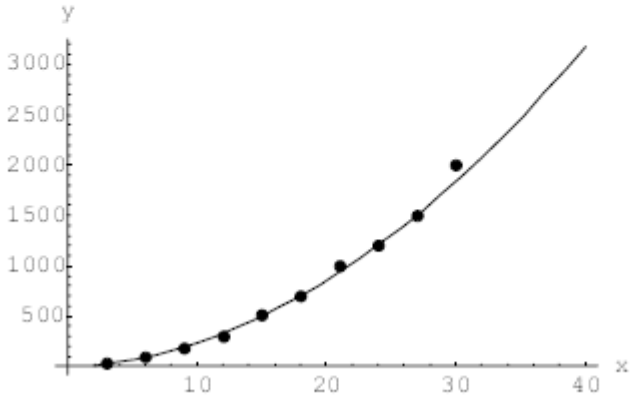


Рис. 1. Графіки вихідної функції та табуляції

Обчислимо абсолютну ε і відносну δ середньоквадратичну похибку інтерполяції за такими формулами:

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^{n} \Delta_i^2}{n}}, \delta = \frac{\varepsilon}{y_{\min}} \cdot 100\%, \qquad (1)$$

де $\Delta_i = y(x_i) - \varphi(x_i)$ – різниця між значеннями табульованої функції $y(x_i)$ і функції інтерполяції $\varphi(x_i)$,

n – кількість значень табульованої функції,

y_{min} – мінімальне значення функції y(x).

Розрахунок значень абсолютної і відносної і похибок наведено у лістингу 3.

```
v1={26,90,180,300,500,700,1000,1200,1500,2000}
v2={19.0476,90.,193.571,329.762,498.571,700.,
    934.048,1200.71,1500.,1831.9}
z=v1-v2
 {26,90,180,300,500,700,1000,1200,1500,2000}

{19.0476,90.,193.571,329.762,498.571,700.,934.048,1200.
71,1500.,1831.9}
 {6.9524,0.,-13.571,-29.762,1.429,0.,65.952,-
0.71,0.,168.1}
 z^2

{48.3359,0.,184.172,885.777,2.04204,0.,4349.67,0.5041,0
.,28257.6}

Plus[48.33586576000001`,0.`,184.17204099999995`,885.776
644`,2.0420409999999247`,0.`,4349.666303999999`,0.50410
00000000516`,0.`,28257.609999999968`]
 33728.1
 Sqrt[33728.1/10]
 58.0759
 58.0759/26*100
 223.369
 58.0759/2000*100
 2.9038
```

Лістинг 3

Функція InterpolatingPolynomial

Для інтерполяції поліноміальних функцій використовують функцією $InterpolatingPolynomial$, яка має такий формат:

$InterpolatingPolynomial\ [z, x]$,

де z −матриця вихідних даних,

x −аргумент функції z.

Приклад використання функції $InterpolatingPolynomial\ [z, x]$ продемонструємо на прикладі даних, наведених у табл. 3.

Таблиця 3. Табличні дані

X	1	2	3	4	5
Y	1	8	27	64	125

Методика використання функції показана у лістингу 4. Згідно з лістингом функція інтерполяції має вид $y = x^3$. Розв'язок є точним. Для спрощення виразу використано функцію $Simplify$.

```
z={{1,1},{2,8},{3,27},{4,64},{5,125}}
InterpolatingPolynomial[z,x]
 {{1,1},{2,8},{3,27},{4,64},{5,125}}
 1+(-1+x) (7+(-2+x) (3+x))
 Simplify[%]
 x^3
```

Лістинг 4

Функція $InterpolatingPolynomial\,[z,x]$ розв'язує задачі інтерполяції також у випадку не рівновіддалених вузлів. Елементами вектора z може бути набір функцій $f_1(x), f_2(x), \ldots$. Тоді розв'язок видаватиметься у вигляді багаточлена зі зберіганням вихідного виду виразу $f_1(x), f_2(x), \ldots$, лістинг 4.

```
z={1,Sin[x],1/x,Exp[-x]}
z1=InterpolatingPolynomial[z,x]
 {1, Sin[x], 1/x, E^-x}

1+(-1+x) (-1+Sin[x]+(-2+x) (1/2 (1+1/x-2 Sin[x])+
       1/3 (-3+x) (-1/2 (1+1/x-2 Sin[x])+1/2 (E^-x-2/x+Sin[x]))))
 Table[z1,{x,1,4}]
N[%]
 {1, Sin[2], 1+2 (-1+1/2 (4/3-2 Sin[3])+Sin[3]),
  3 (-1+Sin[4]+2 (1/2 (5/4-2 Sin[4])+
       1/3 (1/2 (-1/2+1/E^4+Sin[4])+1/2 (-5/4+2 Sin[4]))))}
 {1.,0.909297,0.333333,0.0183156}
```

Лістинг 5

Значення функції z показані в табл. 4.

Таблиця 4. Значення функції

x	1	2	3	4
y	1	$sinx$	$1/x$	e^{-x}
$y(x)$	1	0,909297	0,333333	0,0183156

Функція Interpolation[data]

Ця функція дозволяє розв'язувати задачу інтерполяції над даними (data) у діапазоні аргументів, які задані цими даними. При цьому функція апроксимації користувачу невідома. Дані наводяться у вигляді матриці функції $y = f(x)$.

При введені цієї функції Mathematica видає не функцію інтерполяції, а нову функцію:

$InterpolatingFunction[\{\{x_H, x_K\}\}, <>]$,

де x_H, x_K −діапазон аргументів функції інтерполяції.

Якщо тепер ввести значення аргументу з діапазону $x_H - x_K$, то відповідь буде значенням функції при заданому значенні аргументу.

Розглянемо методику використання функції на прикладі. Таблична функція $y = f(x)$ наведена в табл. 5. Необхідно визначити значення функції при x= 5.8 і x = 18.5 та перевірити достовірність отриманих розв'язків.

Таблиця 5. Функція $y = f(x)$ в табличній формі

X	2	3	8	12	20
Y	1	2,5	4,6	3,2	1,6

Розв'язання задачі наведено у лістингу 5.

```
f={{2,1},{3,2.5},{8,4.6},{12,3.2},{20,1.6}}
y=Interpolation[f]
 {{2,1},{3,2.5},{8,4.6},{12,3.2},{20,1.6}}
 InterpolatingFunction[{{2.,20.}},<>]
y[5.8]
4.56372
y[18.5]
1.18763
Table[y[{2,3,8,12,20}]]
{1.,2.5,4.6,3.2,1.6}
```

Лістинг 6

Відповідно до лістингу 6 функція $Interpolation[data]$ інтерполює дані точним методом у вузлах. Результатами табуляції є точні значення функції у вузлах інтерполяції.

Інтерполяція нелінійними функціями

Якщо інтерполяційна функція є нелінійною, то для визначення її коефіцієнтів точним методом у вузлах використовуються два способа:

- створення й розв'язування системи нелінійних рівнянь;
- лінеаризація нелінійної інтерполяційної функції шляхом перетворення координат.

Розглянемо обидва способи.

Спосіб 1. Розв'язування системи нелінійних рівнянь

Методика інтерполяції цим способом полягає в розв'язувані системи нелінійних рівнянь. Продемонструє її на основі даних, навдених у табл. 6.

Таблиця 6. Функція $y = f(x)$ у табличній формі

X	1	2	3	4	5	6	7	8
У	7,6	16	33	71	156	341	750	1650

Інтерполяційна функція має аналітичний вираз $y = ab^x + c$. Необхідно визначити невідомі a, b, c та похибку функції інтерполяції. Оберемо три наступні координати функції: (1,7.6), (4,71), (7, 750) і складемо систему рівнянь:

$$ab^1 + c = 7{,}6$$
$$ab^4 + c = 71$$
$$ab^7 + c = 750$$

Розв'яжемо цю систему нелінійних рівнянь за допомогою функції FindRoot за наступними початковими значеннями невідомих: $a = 2{,}5, b = 3, c = 1{,}5$. Розв'язок наведено у лістингу 7.

```
f={a b+c==7.6, a b^4+c==71,a b^7+c==750}
FindRoot[f,{a,2.5},{b,3},{c,1.5}]
 {a b + c == 7.6, a b^4 + c == 71, a b^7 + c == 750}
 {a→2.96224,b→2.20425,c→1.0705}
 y=2.96224*2.20425^x+1.0705
Table[y,{x,1,8}]
 1.0705 + 2.96224 2.20425^x

{7.60002,15.4632,32.7956,71.0005,155.214,340.841,75
0.009,1651.92}
 v1={7.5,16,33,71,156,341,750,1650}
v2={7.6,15.4632,32.7956,71,155.214,
    340.841,750,1651.92}
(v1-v2)^2
 {7.5,16,33,71,156,341,750,1650}

{7.6,15.4632,32.7956,71,155.214,340.841,750,1651.92
}

{0.01,0.288154,0.0417794,0,0.617796,0.025281,0,3.68
64}
Plus[0.009999999999999929`,0.28815423999999945`,0.0
4177935999999988`,0,0.6177960000000021`,0.025280999
999997396`,0,3.6864000000002792`]

 4.66941
 Sqrt[Out[33]/8]
 0.763987
 Out[34]/7.6 100
 10.0525
```

Лістинг 7

З лістингу 7 видно, що математичною моделлю об'єкту є наступна функція інтерполяції:

$y = 2.96 \cdot 2.2^{x} + 1.07.$

Абсолютна й відносна середньоквадратична похибка обчислені відповідно до формули (1). Відносна похибка дорівнює 10%.

Спосіб 2. Лінеаризація нелінійної функції

У системі Mathematica апроксимація нелінійними функціями може бути зведена до розв'язування лінійних рівнянь шляхом перетворення координат.

Вирівнювання функцій здійснюється шляхом їх перетворення у лінійну функцію методом заміни змінних. Найбільш просто ці перетворення здійснюються за умови використання степеневих, логарифмічних, дрібно-лінійних, показникових функцій. Методика інтерполяції методом лінеаризації нелінійних функцій здійснюється при виконанні наступних дій:

1. Перетворення функції інтерполяції до лінійної форми.
2. Перетворення матриці вихідних даних у матрицю нових змінних.
3. Утворення системи лінійних рівнянь.
4. Розв'язування системи лінійних рівнянь.
5. Утворення функції інтерполяції.
6. Перевірка адекватності отриманої моделі.

Розглянемо методику на прикладі даних, наведених у табл. 7.

Таблиця 7. Функція $y = f(x)$ в табличній формі

X	1	3	5	7	9	11	13	15
Y	2,5	7,8	18,7	28,5	39	50	61,7	73,8

Необхідно знайти математичну модель досліджуваного об'єкту, якщо відомо, що функція інтерполяції є показниковою функцією $y = ax^b$. Розв'язок задачі показано у лістингу 8.

```
f1={1.,3.,5.,7.,9.,11.,13.,15.}
f2={2.5,7.8,18.7,28.5,39.,50.,61.7,73.8}
Log[f1]
 {1.,3.,5.,7.,9.,11.,13.,15.}
 {2.5,7.8,18.7,28.5,39.,50.,61.7,73.8}

{0.,1.09861,1.60944,1.94591,2.19722,2.3979,2.56495,
2.70805}
 Log[f2]

{0.916291,2.05412,2.92852,3.3499,3.66356,3.91202,4.
12228,4.30136}
 Solve[{2.054==A+1.09861 b,3.912==A+2.398 b},{A,b}]
 {{A→0.483096,b→1.4299}}
 a=Exp[0.483096]
 1.62109
 y=1.62 x^1.43
Table[{x,y},{x,1,15,2}]
  1.43
 1.62 x
{{1,1.62},{3,7.79468},{5,16.1824},{7,26.1821},{9,37
.5044},{11,49.9697},{13,63.4533},{15,77.862}}
```

Лістинг 8

У перших двох рядках лістингу 8 знаходяться вектори вихідних даних із ім'ям $f1$ і $f2$. За ними йдуть ті самі вектори в логарифмічному маштабі. Для обчислення коефіцієнтів a і b лінійної функції $lny = lna + blnx$. За вихідні дані взято координати (3,7), (11,50), які в логарифмічному маштабі мають значення: (1.09861, 2.05412), (2.3979, 3.91202). Тоді система рівнянь має вигляд:

$2.05412 = A \cdot 1.098616^b;$

$3.91202 = A \cdot 2.3979^b,$

де $A = lna$.

Система розв'язується за допомогою функції $Solve$. У результаті розв'язування отримані наступні значення коефіцієнтів: $a = 0.483096$, $b = 1.4299$. Тоді $a = e^A = 1.62109$, а функція інтерполяції має вигляд: $y = 1.62x^{1.43}$. Коефіцієнти a і b функції інтерполяції округлені до двох значущих цифр після крапки. Адекватність моделі доведена табуляцією функції інтерполяції за допомогою функції $Table$. Порівнявши результати табуляції з вихідними даними, можна зробити висновки, що задача розв'язана правильно (значення функцій практично збігаються у вузлах інтерполяції $x = 3$ і $x = 11$), а функція інтерполяції є математичною моделлю об'єкту, що вивчається. З метою порівняння результатів двох способів інтерполяції була розв'язана задача інтерполяції нелінійної функції $y = ax^b$. Були обрані ті самі вузли інтерполяції $x = 3$ і $x = 11$ й розв'язана система нелінійних рівнянь за допомогою функції $FindRoot$ за початкових наближеннях $a_0 = 2,\ b_0 = 1$. Розв'язок має вигляд:

```
f={7.8==a 3^b,50==a 11^b}
FindRoot[f,{a,2},{b,1}]
  {7.8 == 3^b a, 50 == 11^b a}
  {a→1.62121,b→1.42994}
```

Відповідно до розв'язку, коефіцієнти функції інтерполяції практично збігаються у випадку лінеаризації рівняння.

Інтерполяція, наближена у вузлах

Інтерполяція, наближена у вузлах (апроксимація), здійснюється за критерієм мінімуму середньоквадратичної похибки (метод найменших квадратів). Реалізується системою Mathematica за допомогою функції Fit. Функція Fit має вигляд:

$Fit[\{\{M\}\}, \{X\}, x]$,

де M −матриця вихідних даних,

X − перелік базисних змінних,

x − аргумент функції.

Методика інтерполяції за допомогою функції $Fit[\{\{M\}\}, \{X\}, x]$ реалізується при виконанні наступних дій:

1. Введення матриці вихідних даних із присвоєнням їй унікальної назви, наприклад, M.
2. Введення базисних змінних *X*.
3. Введення функції $Fit[\{\{M\}\}, \{X\}, x]$.

Приклад розв'язування задач інтерполяції покажемо на даних, наведених у табл. 8.

Таблиця 8. Функція $y = f(x)$ в табличній формі

x	1	3	4	7	10
Y	3,5	6,7	4,2	2,8	1,2

Необхідно знайти функцію інтерполяції в базисі $a, x, x^2, \frac{x}{1+x}, e^x$. У даному прикладі функція Fit матиме вигляд:

$Fit[\{\{1,3.5\},\{3,6.7\},\{4,4.2\},\{7,2.8\},\{10,1.2\}\}, \left\{a, x, x^2, \frac{x}{1+x}, Exp[x]\right\}, x]$

Розв'язок показано у лістингу 9.

```
M={{1,3.5},{3,6.7},{4,4.2},{7,2.8},{10,1.2}}
X={1,x,x^2,x/(1+x),Exp[x]}
Fit[M,X,x]
 {{1,3.5},{3,6.7},{4,4.2},{7,2.8},{10,1.2}}
 {1, x, x^2, x/(1+x), E^x}

-33.1913 - 0.00090638 E^x - 16.2137 x + 1.22525 x^2 + 103.364 x/(1+x)
```

Лістинг 9

Паде апроксимація

Апроксимацію Паде використовують для інтерполяції функції, що задана в аналітичному вигляді дрібно-раціональною функцією. Вбудована функція Паде-апроксимація знаходиться в пакеті розширення

$\ll calculus\backslash pade.m$

Функція Паде має вигляд:

$Pade[f(x), \{x, r, n_1, n_2\}]$

де $f(x)$ −функція, що задана в аналітичному вигляді,

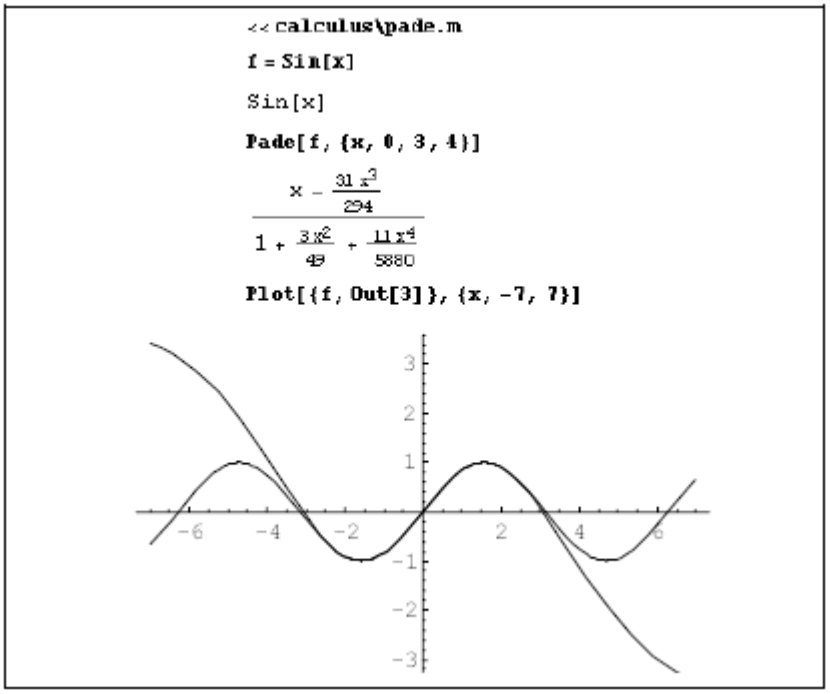
x −аргумент функції $f(x)$,

r −точка, поблизу якої справедлива функція апроксимації,

n_1 −степінь багаточлена числівника,

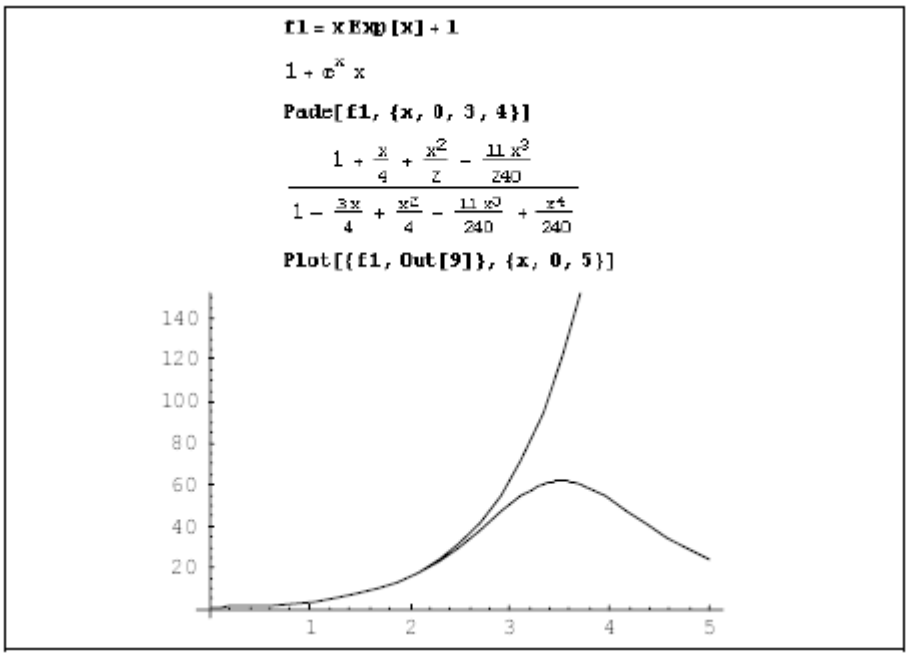
n_2 −степінь багаточлена знаменника.

Виконаємо Паде-апроксимацію функції $y = sinx$ поблизу $x = 0$ при $n_1 = 3$ і $n_2 = 4$. Процедура апроксимації в середовищі Mathematica наведена у лістингу 10.



Лістинг 10

Відповідно до лістингу 10 апроксимація виконана правильно. У діапазоні від $x = -3$ до $x = 3$ функції збігаються. Апроксимація Паде не має обмежень на вигляд вихідної функції, що показано у наступному прикладі. Необхідно виконати Паде-апроксимацію функції $y = xe^x + 1$ поблизу $x = 0$ при $n_1 = 3$ і $n_2 = 4$. Переконатися у певності апроксимації шляхом побудови графіків. Результат апроксимації наведено у лістингу 11.



Лістинг 11

Сплайн-інтерполяція

Функція реалізації сплайн-інтерполяції розміщена у підпакеті SplineFit пакету NumericalMath. Інтерполяція кубічними сплайнами забезпечує високу точність математичної моделі. Функція має такий формат:

SplineFit[F, type],

Де F – дані, представлені у вигляді матриці,

type - тип апроксимації, по замовчуванню – апроксимація кубічними сплайнами (Cube). Також можлива апроксимація сплайнами Bezier та CompositeBezier.

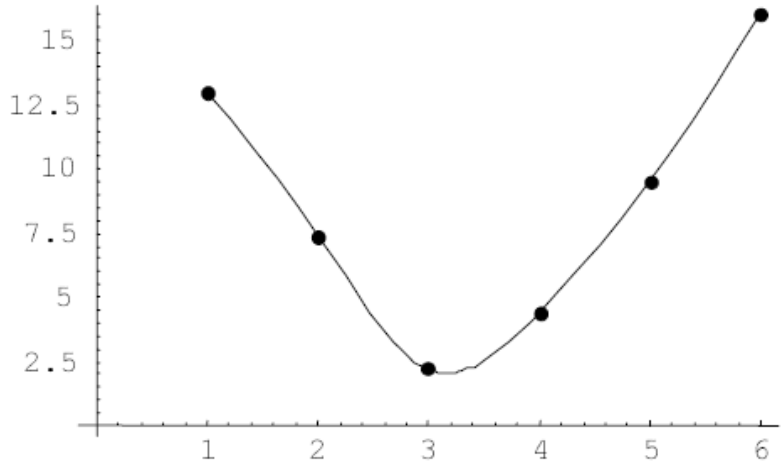
Розглянемо використання функції на прикладі матриці:

M=((1,13), (2, 7.4), (3, 2.2), (4, 4.4), (5, 9.5), (6, 16)).

Необхідно отримати математичну модель функції $y = f(x)$, використовуючи кубічну сплайн-інтерполяцію. Достовірність рішення перевірити графічно.

Розв'язок приведено у лістингу 12.

```
<<NumericalMath`SplineFit`
M = {{1, 13}, {2, 7.4}, {3, 2.2}, {4, 4.4}, {5, 9.5}, {6, 16}}
{{1, 13}, {2, 7.4}, {3, 2.2}, {4, 4.4}, {5, 9.5}, {6, 16}}
Z1 = SplineFit[M, Cubic]
SplineFunction[Cubic, {0., 5.}, <>]
Z1[0]
{1, 13}
Z1[1]
Z1[5]
{6., 16.}
Z1[3.53]
{4.53, 6.92269}
r1 = ParametricPlot[Z1[x], {x, 0, 5}]
r2 = ListPlot[M, PlotStyle → {PointSize[0.02]}]
Show[r1, r2]
```



Лістинг 12

З лістингу 11 видно, що розв’язок отриманий не в явному вигляді, тобто аналітичний вираз функції інтерполяції відсутній. Це є суттєвим недоліком функції SplineFit[F, type]. З графіку можна зробити висновок, що інтерполяція виконана достатньо точно.

Лекція № 10. Функції спектрального аналізу системи Mathematica

Система Mathematica має багаті ожливості спектрального аналізу та снтезу сигналів. Велика кількість вбудованих функцій дозволяє виконувати:

- спектральний аналіз сгналів;
- гармонічний аналіз сигналів;
- фильтраціюсигналів.

Перетворення Фур'є виконується в системі Mathematica за допомогою наступних функцій:

- **FourierTransform[F(t), t, w]** – повертає результат прямого перетворення Фур'є функції **F(t)**, що представлена параметром **w**;
- **InverseFourierTransform[F(w), w, t]** – повертає результат зворотного перетворення Фур'є виразу **F(w)**, що предсталений параметром **t**;
- **FourierSinTransform[F(t), t, w]** – повертає результат синусного перетворення Фур'є функції **F(t)**, що представлена параметром **w**;
- **FourierCosTransform[F(t), t, w]** – повертає результат косинусного перетворення Фур'є функції **F(t)**, що представлена параметром **w**;
- **FourierTransform[F, {t_1, t_2, …}, {w_1, w_2, …}]** – повертає результат прямого перетворення Фур'є функції **F{t_1, t_2, …}**, що представлена параметрами **{w_1, w_2, …}**;
- **InverseFourierTransform[F, {w_1, w_2, …}, {t_1, t_2, …}]** – повертає результат зворотного перетворення Фур'є виразу **F{w_1, w_2, …}**, що предсталений параметрами **{t_1, t_2, …}**.

Необхідно отримати пряме та зворотнє перетворення Фур'є наступних функцій:

$F(t) = te^{\alpha t}$;

$F(t) = t^2 \cos(bt)$;

$F(t) = t^2 \sin(at)$;

$F(t_1, t_2) = \cos(t_1, t_2)$.

```
FourierTransform[t Exp[a t],t,w]
```

$$-i\sqrt{2\pi}\,\text{DiracDelta}'[-i\,a+w]$$

```
InverseFourierTransform[%,w,t]
```

$$e^{at}\,t$$

```
FourierSinTransform[t^2  Cos[b t],t,w]
```

$$\frac{1}{\sqrt{2\pi}\ (b-w)^3}-\frac{1}{\sqrt{2\pi}\ (-b+w)^3}-\frac{\sqrt{\frac{2}{\pi}}}{(b+w)^3}$$

```
FourierCosTransform[t^2  Sin[a t],t,w]
```

$$-\frac{1}{\sqrt{2\pi}\ (a-w)^3}+\frac{1}{\sqrt{2\pi}\ (-a+w)^3}-\frac{\sqrt{\frac{2}{\pi}}}{(a+w)^3}$$

```
FourierTransform[ Cos[t1*t2],{t1,t2},{w1,w2}]
```

$$\frac{1}{2}\left(e^{-i\,w1\,w2}+e^{i\,w1\,w2}\right)$$

```
InverseFourierTransform[%,{w1,w2},{t1,t2}]
```

$$\frac{1}{2}\left(e^{-i\,t1\,t2}+e^{i\,t1\,t2}\right)$$

```
FullSimplify[%]
Cos[t1 t2]
```

Лістинг 1

З лістингу 1 видно, що перетоврення виконані вірно, оскільки зворотні перетворення Фур'є співпадають з вихідними функціями.

Спектральний аналіз на основі прямого перетворення Фур'є. Пряме перетворення Фур'є дозволяє отримати частотний спектр сигналу, представленого отсчетами його тимчасової залежності. Нерідко це є кінцевою метою спектрального аналізу.

На рис. 10.1 представлений приклад спектрального аналізу простого сигналу – трикутного імпульсу, заданого за допомогою функції **If**. Потім, за допомогою функції **Fourier** прямого перетворення Фур'є отримані в явному вигляді вектори амплітуд Mg і фаз Ag гармонік цього сигналу.

Спектральный анализ для функции f2[t], задающей пилообразный импульс

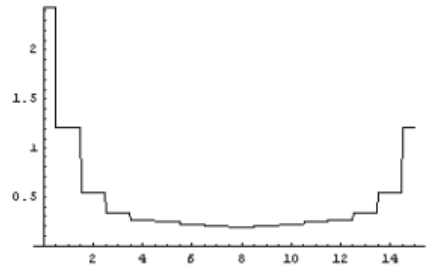
```
f2[t_] := If[(0 > t) || (t > 1.5), 0, t];

data := Table[f2[t / 8], {t, 16}]; ft := Fourier[data]; ft

{2.4375 + 0. i, -1.21082 + 0.0764104 i, 0.112056 + 0.528109 i, 0.294312 - 0.154429 i,
 -0.1875 - 0.1875 i, -0.131399 + 0.202265 i, 0.200444 + 0.0906092 i, 0.0479075 - 0.191895 i,
 -0.1875 + 0. i, 0.0479075 + 0.191895 i, 0.200444 - 0.0906092 i, -0.131399 - 0.202265 i,
 -0.1875 + 0.1875 i, 0.294312 + 0.154429 i, 0.112056 - 0.528109 i, -1.21082 - 0.0764104 i}
```

Создадим векторы амплитуд и фаз всех 16 гармоник

```
Mg := Abs[ft]; Mg

{2.4375, 1.21323, 0.539867, 0.332367, 0.265165, 0.241199, 0.219972, 0.197785,
 0.1875, 0.197785, 0.219972, 0.241199, 0.265165, 0.332367, 0.539867, 1.21323}

Ag := Arg[ft]; Ag

{0., 3.07857, 1.36171, -0.483222, -2.35619, 2.14692, 0.424551, -1.32614,
 3.14159, 1.32614, -0.424551, -2.14692, 2.35619, 0.483222, -1.36171, -3.07857}
```

Построим спектрограмму модулей гармоник, обеспечив верное отражение нулевой гармоники.

```
Plot[Mg[[Round[i] + 1]] , {i, 0, 15}, PlotPoints → 16, PlotRange → {0, Mg[[1]]}]
```



Построим спектрограмму фаз гармоник.

```
Plot[Ag[[Round[i] + 1]], {i, 0, 15}, PlotPoints → 16, PlotRange → {-π, π}]
```

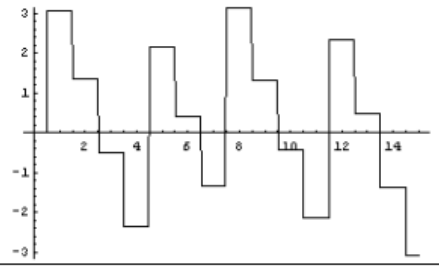


Рис. 10.1. Приклад спектрального аналізу трикутного імпульсу

Тут за допомогою графіків сходового типу, підкреслюють дискретність гармонік, побудовані спектрограми амплітуд і фаз гармонік пилообразного імпульсу. Добре видно симетричне відображення ліній спектра щодо восьмий гармоніки - в нашому випадку було 16 відліків сигналу. Це означає, що амплітуда і фаза дев'ятої гармоніки ті ж, що у сьомий гармоніки, у десятій – ті ж, що у шостий і т.д.

Лекція № 11. Основи програмування в середовищі Mathematica

Mathematica як система програмування

Поняття про вхідну мову системи і мов уреалізації

В системі Mathematicaздатна можливо вирішити багато задач без використання програмування. Однак всі засоби системи (алфавіт, літери, цифри, оператори та спеціальні знаки) є частиною проблемно-орієнтованої мови програмування надвисокого рівня. За своїми можливостями у виконанні математичних та науково-технічних обчислень ця мова має значні переваги у порівнянні зі звичайними мовами програмування – Паскаль, С, С++.

Можливості мови програмування системи Mathematica

Система Mathematica містить велику кількість функцій , серед яких чимало таких , які реалізують математичні перетворення і сучасні обчислювальні методи, як чисельні , так і аналітичні . Мова програмування системи Mathematica є типовим інтерпретатором і не призначений для створення виконуваних файлів. Проте, окремі вирази можна компілювати за допомогою функції Compile, що корисно при необхідності збільшення швидкості рахунків. Мова системи Mathematica дозволяє реалізувати більшість типів програмування: функціональне, структурне, об'єктно-орієнтоване, математичне, логічне, рекурсивне і т.д.

Ядро системи Mathematica має функціональну структуру. Мова системи дозволяє розбивати програми на окремі модулі (блоки), процедури і функції з локальними змінними. Об'єктно-орієнтоване програмування базується на узагальненому понятті об'єкта. В системі Mathematica об'єктами можуть бути математичні вирази, вхідні і вихідні дані, графіки та малюнки, звуки і т.д. З поняттям об'єкта тісноп ов'язані три основні властивості: інкапсуляція, спадкування і поліформізм. Всі вони притаманні об'єктам системи Mathematica і не вимагають для своєї реалізації спеціальних засобів. Інкапсуляція означаєо б'єднання в одному об'єкті як даних, так і методів їх обробки. Спадкування означає, що кожен об'єкт, похідний від інших об'єктів, успадковує їх властивості. Поліформізм - властивість, що дозволяє передати ряду об'єктів повідомлення, яке буде оброблятися кожним об'єктом відповідно до його індивідуальних особливостей.

Мова програмування системи Mathematica спеціально створена для реалізації будь-якого з перерахованих підходів до програмування, а також ряду інших, наприклад рекуррентного програмування, за умови використання якого черговий крок обчислень базується на даних, отриманих на попередніх кроках. Можливо і рекурсивне програмування - це коли функція в загальному випадку неодноразово звертається до себе самої. Засоби мови Mathematica дозволяють здійснити і елементи візуально-орієнтованого програмування. Mathematica дозволяє створювати палітри і панелі з різними кнопками, що дозволяють керувати програмою або вводити нові програмні об'єкти.

Структура програмного середовища системи Mathematica

Стркутуру програмного середовища системи Mathematica можна представитив наступному вигляді, зображеному на рис. 1.

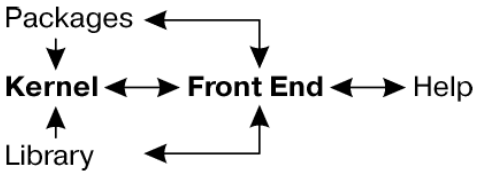


Рис. 1. Стркутура програмного середовища системи Mathematica

Для орієнтації системи на конкретну машинну платформу служить інтерфейсний процесор **FrontEnd** . Саме він визначає , який вигляд має інтерфейс користувача системи . Центральне місце в системах класу Mathematica займає машинно незалежне ядро математичних операцій - **Kernel** . Воно містить набір операторів і функцій, правил обчислень і перетворень математичних виразів. Ядро зроблено досить компактним для того, щоб будь-яка функція з нього викликалася досить швидко. Для розширення набору функцій служить бібліотека **Library** і набір пакетів розширення **Packages**. Пакети розширень створюють на власній мові програмування систем Mathematica і є головним засобом розширення можливостей системи та їх адаптації до вирішення конкретних класів задач користувача. Крім того, системи мають вбудовану довідкову систему **- Help**.

Основи функціонального програмування

Принцип функціонального програмування передбачає використання під час написання лише функцій. При цьому можливе неодноразове вкладення функцій одна в одну. У ряді випадків, особливо під час символьних перетворень, відбувається взаємна рекурсія функцій, супроводжувана майже необмеженим поглибленням рекурсії і наростанням складності оброблюваних системою виразів. Поняття функції асоціюють з обов'язковим поверненням деякого значення у відповідь на звернення до функції. Повернення функціями деяких значень дозволяє застосовувати їх поряд з операторами для складання математичних виразів. Функції поділяються на внутрішні і функції, задані користувачем.

Функції користувача

Процес створення функції у системі Mathematica подібний до інших мов програмування. Наприклад, функцію для зведення x в ступінь n можна було б визначити так :

powerxn [x , n]: = x ^ n

Однак така функція є непрацездатною. Причиною цього є те, що в системі Mathematica символи x і n є звичайними символами і не можуть сприймати формальні параметри. Для реалізації необхідно використовувати змінні-зразки, що мають після своїх імен символи підкреслення. Зразки можуть бути формальними параметрами функцій і сприймати значення фактичних параметрів. Таким чином, правильним буде запис функції користувача у вигляді :

powerxn [x_ , n_]: = x ^ n

Розглянемо ще один простий приклад , в якому задана функція scn [x , n], яка обчислює суму синуса в ступені n і косинуса в ступені n, приклади використання якої наведені у лістингу 1:

scn [x_ , n_]: = Sin [x] ^ n + Cos [x] ^ n
scn [1,2]
$Cos[1]^2 + Sin[1]^2$
scn [x , 2]
$Cos[x]^2 + Sin[x]^2$
scn [x , n]
$Cos[x]^n + Sin[x]^n$

Лістинг 1

Функція може складатись з декількох виразів, які об'єднано круглими дужками:

f [x_]: = (t = (1 + x) ^ 2 ; t = Expand [t])

Змінні списку параметрів, після імені яких стоїть знак « _ » , є локальними в тілі функції або процедури з параметрами. На їх місце підставляється фактичне значення відповідного параметра. Приклад, який ілюструє використання локальних змінних показано у лістингу 2:

f [a + b]
$1 + 2a + 2a^2 + 2b + 2ab + b^2$
t
$1 + 2a + 2a^2 + 2b + 2ab + b^2$

Лістинг 2

Зверніть увагу на те , що змінна t у функції f є глобальною, що пояснює результат останньої операції. Застосування глобальних змінних в тілі функції цілком можливо, але створює так званий побічний ефект, в даному випадку змінює значення глобальної змінної t. Для усунення побічних ефектів необхідно використовувати зразки та інші спеціальні способи створення функцій.

Параметрами функцій можуть бути списки, за умови допустимості їх комбінації. Наприклад у функції powerxn, лістинг 1, припустимо як параметр x використовувати список, а як n – змінну або число, що показано у лістингу 3:

powerxn [{1,2,3,4,5 }, z]
$\{1^z, 2^z, 3^z, 4^z, 5^z\}$
powerxn [{ 1,2,3,4,5 }, 2]
{1,4,9,16,25}

Лістинг 3

Після свого створення функції користувача можуть використовуватися за тими ж правилам , що і вбудовані функції.

Чисті функції

Іноді може знадобитися використання деякої функції тільки в момент її створення. Ця функція представлена тільки виразом без імені, що обумовило її назву. Для створення такого об'єкта служить вбудована функція Function, яку використовують у вигляді :

- Function [body] – створює чисту функцію з тілом body;
- Function [{ x}, body] – створює чисту функцію параметра x з тілом body;

- Function [{ x1, x2, ...}, body] – створює чисту функцію ряду параметрів x1, x2 , ... з тілом body.

Для обчислення створеної таким чином функції після неї записують список параметрів у квадратних дужках. Як приклад у лістингу 4 записано код чистої функції підесення до степеня:

```
Function [{ x , n}, x ^ n]
Function [{ x , n}, x^n]
% [ 2,3 ]
8
```

Лістинг 4

Чисту функцію можна легко перетворити на звичайну функцію користувача, що показує приклад лістинга 5:

```
fun = Function [{ x, n }, x ^ n]
Function [{ x , n}, x^n]
fun [2, 3]
8
```

Лістинг 5

Анонімні функції

Гранично компактну форму задання функцій мають так звані анонімні функції. Вони не мають ні назви, ні звичайного визначення, їх записують виразами спеціального виду. У цьому виразі замість змінних використовують позначення # (для однієї змінної) , # 1 , # 2 , ... для ряду змінних. Завершують тіло функції символом & . Якщо необхідно розрахувати значення функції, то після її запису в квадратних дужках вказують список фактичних параметрів. Приклад застосування анонімної функції показано у лістингу 6:

```
# 1 ^ # 2 & [2,3]
8
# 1 ^ # 2 & [ y , z ]
y^z
```

Лістинг 6

За допомогою анонімних функцій неважко створити звичайні функції користувача :

```
f [x_ , y_ ] = # 1 ^ # 2 & [x, y]
x^y
f [2,3]
8
```

Суперпозиція функцій

Функціональне програмування передбачає використання суперпозиції функцій. Для її реалізації використовують функції:

- Nest[expr, x, n] – застосовує вираз (функцію) до заданого аргументу x n разів.

- NestList[f , x, n] – повертає список використань функції f до заданого аргументу x n +1 раз .

- Fold [f, x, list] – повертає наступний елемент у FoldList[f , x, list].

- FoldList [f, x, {a, b, ... }] – повертає {x, f [x, a], f [f [x, a], b], j}.

- ComposeList[{f1, f2, ...}, x] – генерує список у формі {x , a[x], a[a[x]], ...}.

Функції FixedPoint і Cath

У функціональному програмуванні замість циклів , описаних далі , можна використовувати наступну функція:

- FixedPoint [f, expr] – обчислює expr і застосовує до нього вираз f , поки результат не повториться.

- FixedPoint [f , expr , SameTest_ > comp] - обчислює expr і застосовує до нього вираз f, поки два наступних результати не будуть істинними.

Приклад застосування функції FixedPoint наведено у лістингу 7.

```
FixedPoint[Function[t, Print[t]; Floor[t/2]], 27]
27
13
6
3
1
0
0
```

Лістинг 7

Останній результат 0 виводиться в окремій (нумерованій) комірці виведення і означає завершення процесу ітерацій ділення t на 2. Ланцюговий дріб за допомогою функції Nest можна створити за допомогою таких аргументів:

```
Nest[Function[t, 1/(1 + t)], y, 3]
```

$$\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+y}}}$$

Ще одна функція такого роду - Catch []:

- Catch [expr] – обчислює expr, до першого виконання функції Throw[value], після чого повертає value .

- Catch [expr, form] – обчислює expr до першого виконання функції Throw[value, tag], після чого повертає value .

- Catch[expr, form, f] – повертає f[value, tag] замість value .

Реалізація рекурсивних та рекурентних алгоритмів

Важливе місце у вирішенні багатьох математичних задач займає реалізація рекурсивних та рекурентних алгоритмів. Розглянемо типовий приклад реалізації ітераційного рекуррентного алгоритму обчислення квадратного кореня з виразу f (x) при початковому значенні x0 = a , за такими формулами метода Ньютона:

$x_0 = a$ і $x_n = x_{n-1} - f(x_{n-1}) / f'(x_{n-1})$.

Цю функцію можна записати таким чином:

```
newtoniter[f_, x0_, n_ ]: = Nest[ ( # - f [#] / f '[#] ) &, N[x0], n]
```

Тоді обчислення кореня з виразу $e^x - 2$ з початковим наближенням $x_0 = 0.5$ і кількістю ітерацій n можна організувати за допомогою функцій Nest[] і NestList[], що показано у лістингу 8:

```
newtoniter [ Function [{x}, Exp[x] -2.0], 0.5,5]
```

0.693147
newtoniter[Function[{x}, Exp[x]-2.0], 0.5, #] & / @ Range [5]

{ 0.713061, 0.693344, 0.693147, 0.693147, 0.693147}
newtoniter1 [f_ , x0_ , n_]: = NestList [(# - f [#] / f ' [#]) & , N [x0] , n]
newtoniter1 [Function [{ x } , Exp [x] -2.0] , 0.5,5]
{0.5, 0.713061, 0.693344, 0.693147, 0.693147, 0.693147}

Лістинг 8

У першому випадку повернено останній результат , а в інших – всі проміжні. Функція FixedPoint дозволяє здійснювати ітерації до тих пір, поки результат не перестане змінюватися (з машинною точністю). Це ілюструє приклад, наведений у лістингу 9.

newtonfp[f_,init_,options_]:=FixedPoint[#1-f[#1]/f'[#1]&,init,options]
newtonfp [Function [{ x } , Exp [x] -2.0] , 0.5,5]
0.693147

Лістинг 9

Рекурсивні алгоритми використовують обчислення, у яких у тілі функції відбувається звернення до неї ж самої. Mathematica допускаєтаку можливість. Типовий приклад цього - обчислення факторіала по формулі N! = N * (N-1)! .

Основи процедурного програмування

В основі процедурного програмування лежить поняття процедури як закінченого програмного модуля і типових засобів керування: циклів, умовних та безумовних п і т.д. Хоча, програмування систем Mathematica і в цьому випадку залишається функціональним, оскільки елементи процедурного програмування задані в вигляді функцій. Однак з'являється можливість застосування традиційних засобів програмування – умовних виразів, циклів, процедур і т.д.

Процедури є повністю самостійними програмними модулями , мають свій ідентифікатор і виконують деяку послідовність операцій. Вони можуть бути описані в одному рядку з використанням в якості роздільника символа «;» (крапка з комою), наприклад:

r=(1+x)^2;r=Expand[r];r-1

Ця процедура повертає такий символьний вираз:

Expand[(1+x)^2] - 1

Для створення повноцінних процедур і функцій використовують структуру Block у двох модифікаціях:

Block[{x, y, ...}, procedure] – процедура з декларацією списку локальних змінних x,y,…

Block[{x = x0, y=y0, ...}, procedure] – процедура з декларацією списку локальних змінних x,y,… з початковими значеннями.

Приклад використання структури Block наведено у лістингу 10:

g[x_] := Block[{u}, u = (1+x)^2; u=Expand[u]]
g[a+b]

$1 + 2a + a^2 + 2b + 2ab + b^2$
u
u
u=123456;g[2]
9
u
123456

Лістинг 10

Зверніть увагу, що змінна u, яку використовують у тілі базової структури, є локальною, і присвоєння їй символьного виразу (1+ x) ^ 2 в тілі блоку ігнорується поза блоком. Якщо змінна u до застосування у функції була не визначена,то вона так і залишається невизначеною. А якщо вона мала до цього деяке значення (наприклад, 123456 у нашому випадку), то і після виходу з процедури вона матиме це значення.

Організація циклів

Цикли типу Do

Цикли цього типу мають декілька модифікацій:

- Do[expr, {imax}] – розраховує вираз expr imax разів.

- Do[expr, {i, imax}] – розраховує вираз expr зі змінною i, яка почергово приймає значення від 1 до imax з кроком 1.

- Do[expr, {i, imin, imax}] – розраховує вираз expr зі змінною i, яка почергово приймає значення від imin до imax з кроком 1.

- Do[expr, {i, imin, imax, di}] – розраховує вираз expr зі змінною i, яка почергово приймає значення від imin до imax з кроком di.

- Do[expr, {i, imin, imax}, {j, jmin, jmax}, ...] – розраховує вираз expr з використаннями ряду вкладених циклів зі змінними j, i і т.д.

Приклади організації циклу Do і його виконання наведені у лістингу 11:

Do[Print[«hello»],{5}]
hello
hello
hello
hello
hello
Do[Print[i],{i,3}]
1
2
3
Do[Print[i],{i,5,8}]
5
6
7
8
Do[Print[i],{i,0,1,0.25}]
0
0.25
0.5
0.75
1

Лістинг 11

Змінна i в тілі циклу – ітератор – є локальною. Вся програма з циклом зберігається в одній комірці. В лістингу 12 показано процедуру з циклом Do, що обчислює n-не число Фібоначчі:

```
fibonacci[(n_Integer)?Positive] :=
Module[{fn1 = 1, fn2 = 0},
Do[{fn1, fn2} = {fn1 + fn2, fn1}, {n - 1}]; fn1]
```

Лістинг 12

Зверніть увагу на застосування в цьому прикладі функції Module. Вона створює програмний модуль з локальними змінними (у нашому випадку fn1 і fn2), в якому організовано рекуррентное обчислення чисел Фібоначчі. У лістингу 13 показано застосування циклу Do для створення ланцюгового дробу:

```
x = y; Do[x = 1/(1 + k x), k, 2, 8, 2]; x
```

$$\cfrac{1}{1+\cfrac{8}{1+\cfrac{6}{1+\cfrac{4}{1+\cfrac{2}{y}}}}}$$

Лістинг 13

Цикли типу For

Інший вид циклу For реалізується функцією:

For [start, test, incr, body]

У ній змінній циклу спочатку присвоюється значення start, потім циклічно змінюється від цього значення до значення body з кроком incr; і так до тих пір, поки умова test буде іситинною. Коли умова стане хибною, цикл закінчується. У прикладі лістингу 14 показано просту програму з циклом For і результат її виконання:

```
Print[«i x»]
For [x=0;i=0,i<4,i++
[x+=5*i,Print[i,» «,x]]]
i x
1 5
2 15
3 30
4 50
Return[x]
Return[50]
```

Лістинг14

Програма, наведена вище, дозволяє спостерігати за зміною значень змінної циклу *i* і змінної x, що одержує за кожен цикл прирощення рівне 5*i. У кінці документа показаний приклад на використання функції повернення значень Return [x]. У циклі For використовують глобальні змінні, тому необхідний контроль за їх використанням.

Цикли типу While

Форма запису циклу:

While [test, expr] .

У цьому типі циклу розраховується значення expr до тих пір, поки умова test істинна. Нижче розглянуто приклад з організації та використання циклу While.

```
While[i<5,i+=1;x+=2*i;Print[i,» «,N[x]]]
i x
2       5.
3       11.
4       19.
5       29.
```

Лістинг 15

Апарат локальних змінних в цьому типі циклів не використовується.

Директиви переривання і продовження циклів

У зазначених типах циклів і в інших структурах можна використовувати такі директиви-функції:

- Abort [ ] – припиняє обчислення з повідомленням $ Aborted.
- Break [ ] – виконує вихід з тіла циклу або рівня вкладеності програми, що містить даний оператор (цикли типу Do , For і While або тіло оператора - перемикача Swith) . Оператор повертає Null - значення.
- Continue [ ] – задає перехід на наступний крок поточного циклу Do , For абоWhile .
- Interrupt [ ] – перериває обчислення з можливістю їх поновлення за допомогою діалогового вікна.
- Return [ ] – перериває виконання з поверненням значення Null .
- Return [expr] – перериває виконання з поверненням значення виразу expr.

Умовні вирази і безумовні переходи

Функція If

Як і у більшості мов програмування, умовні вирази задаються за допомогою оператора або функції If. Система Mathematica має такі модифікації функцію If:

- If [condition, t, f] – повертає результат t, якщо обчислення умови condition істинне, та f, якщо результат хибний.
- If [condition, t, f, u] - повертає результат u, якщо результат обчислення умоови condition не буде ні істинним, ні хибним.

У лістингу 16 показаний опис процедури з циклом Do, вихід з якої організовано за допомогою функції If і директиви переривання Aborted []:

```
x:=1;Print["i x"];
Do[{If [i==5,Abort[],None],
i+=1;x+=2*i;Print[i," ",N[x]]},
{i,1,100}]
i x
2 5.
3 11.
```

4 19.
5 29.
$Aborted
Return[x]
Return[29]

Лістинг 16

Аналогічний приклад з використанням функції Break у функції If показано у лістингу 17.

```
x:=1;Print["i x"];
Do[{If [i==5,Break[],None],
i+=1;x+=2*i;Print[i," ",N[x]]},
{i,1,100}]
```

i x
2 5.
3 11.
4 19.
5 29.
Return[x]
Return[29]

Лістинг 17

У даному випадку ніяких спеціальних повідомлень про вихід з циклу не дається.

Функції - перемикачі

Для організації декількох розгалужень в системі Mathematica використовують оператори – перемикачі Which і Switch :

- Which [test1 , value1 , test2 , value2 , ...] - обчислює в порядку проходження кожну умову testi і повертає значення valuei першої умови testi, яка була істинна.

- Switch [expr , form1 , value1 , form2 , value2 , ...] - обчислює значення умови expr , потім порівнює його послідовно з кожним виразом formi і повертає значення valuei, першого збігу виразу formi з умовою expr.

Приклади використання функції Which наведено у лістингу 18.

```
Which[1==2,1,2==2,2,3==3,3]
```

2

```
Which[1==2,x,2==2,y,3==3,z]
```

y

Лістинг 18

Приклади, що демонструють використання функції Swhitch наведено у лістингу 19.

```
Switch[2,1,a,2,b,3,c]
```

b

```
Switch[3,1,a,2,b,3,c]
```

c

```
Switch[8,1,a,2,b,3,c]
```

Switch[8,1,a,2,b,3,c]

Лістинг 19

Зверніть увагу на останній приклад: за умови неправильного запису першого параметра, запис функції повторюється.

Безумовні переходи

Оператор безумовного переходу Goto [tag] створює перехід до місця програми, позначене міткою Label [tag]. Можливі також форми Goto [expr] і Label [expr], де expr – певний вираз. Використання оператора Goto показано у лістингу 20.

```
(q = 2; Label[start]; Print[q]; q += 2;
If[q < 7, Goto[start]])
2
4
6
```

Лістинг 20

Тут за допомогою оператора Goto [start] організовано цикл з переходом на мітку Label [start], який виконується , поки значення q менше 7. При цьому q змінюється від початкового значення 2 з кроком 2. Цікавою особливістю мови програмування Mathematica є можливість створення переходів за значенням обчислюваного виразу. Наприклад, Goto [2 + 3] дає перехід до мітці Label [5] або навіть Label [1 + 4], що видно з наступного прикладу:

```
Goto[2+3];
Print["aaaaa"];
Label[1+4];
Print["bbbbb"]
bbbbb
```

Лістинг 21

Лекція №12. Анонімні та чисті функції

Функції користувача

Хоча в системах Mathematica близько тисячі вбудованих функцій, будь-якому користувачеві рано чи пізно може знадобитися створення будь-якої своєї функції. Здається природним задати її за правилами, прийнятим у багатьох мовах програмування. Наприклад, функцію для зведення x в ступінь n можна було б визначити так:

powerxn [x, n]: = x ^ n

Однак виявляється, що така функція працювати відмовляється:

powerxn [a, b]

powerxn [a, b]

Причина цього в тому, що в системі Mathematica символи x і n є звичайними символами, які не наділені особливими властивостями. Будучи використаними в якості параметрів функції, вони не здатні сприймати формальні параметри (в нашому випадку a і b).

Для того щоб функція користувача була повноцінною, в списку параметрів необхідно використовувати зразки у вигляді змінних, але мають після своїх імен символи підкреслення. Зразки здатні бути формальними параметрами функцій і сприймати значення фактичних параметрів. Таким чином, правильною буде запис функції користувача у вигляді:

powerxn [x_, n _]: = x ^ n

Тепер обчислення за заданою функції користувача пройде гладко, причому як в чисельному, так і в символьному вигляді:

powerxn [2,3]

8

powerxn [a, b]

Розглянемо ще один простий приклад, в якому задана функція **scn[x, n]**, обчислює суму синуса в ступеня n і косинуса в ступеня n:

scn [x_, n _]: = Sin [x] ^ n + Cos [x] ^ n

scn [1,2]

$Cos[1]^2 + Sin[1]^2$

scn [x, 2]

$Cos[x]^2 + Sin[x]^2$

scn [x, n]

$Cos[x]^n + Sin[x]^n$

У цьому простому прикладі результат обчислень – є повернене функцією **scn** символьне значення. Можна також задати функцію користувача, що містить кілька виразів, уклавши їх в круглі дужки:

f [x _]: = (t = (1 + x) ^ 2; t = Expand [t])

Змінні списку параметрів, після імені яких стоїть знак «_», є локальними в тілі функції або процедури з параметрами. На їх місце підставляється фактичне значення відповідного параметра, наприклад:

f [a + b]

$1+2a+a^2+2b+2ab+b^2$

t

$1+2a+a^2+2b+2ab+b^2$

Зверніть увагу на те, що змінна t у функції f є глобальною.

Це пояснює результат останньої операції. Застосування глобальних змінних в тілі функції цілком можливо, але створює так званий побічний ефект, в даному випадку змінює значення глобальної змінної t. Для усунення побічних ефектів необхідно використовувати зразки та інші спеціальні способи завдання функцій, описані нижче.

Отже, можна сформулювати ряд правил для завдання функцій користувача:

• така функція має ідентифікатор-ім'я, який повинен бути унікальнавим і досить зрозумілим;

• в списку параметрів функції, розміщеному в квадратних дужках після ідентифікатора, повинні використовуватися зразки змінних, а не просто імена змінних;

• змінні в тілі функції, зазначені в списку змінних зразків є локальними;

• може використовуватися відкладене: = або неотложения = привласнення;

• тіло функції може містити кілька виразів, укладених в круглі дужки, при цьому повертається значення останнього виразу;

• змінні зразків в списку параметрів є локальними і діють в межах тільки тіла функції;

• в тілі функції можуть використовуватися глобальні змінні, але при цьому можливі побічні ефекти.

Параметрами функцій можуть бути списки, за умови допустимості їх комбінації. Наприклад, припустимо поставити x списком, а n – змінної або числом:

powerxn [{1,2,3,4,5}, z]

$\{1, 2^z, 3^z, 4^z, 5^z\}$

powerxn [{1,2,3,4,5}, 2]

{1,4,9,16,25}

Після свого завдання функції користувача можуть використовуватися за тими ж правилами, що і вбудовані функції.

Завдання чистих функцій

Іноді може знадобитися завдання деякої функції тільки в момент її створення. Ця функція є лише вираженням без імені, звідси і їїназва чиста функція. Для створення такого об'єкта служить вбудована функція Function, яка використовується у вигляді:

- **Function [body]** - створює чисту функцію з тілом body;
- **Function [{x}, body]** - створює чисту функцію параметра x з тілом body;
- **Function [{x1, x2, ...}, body]** - створює чисту функцію ряду параметрів x1, x2, ... з тілом body.

Для обчислення створеної таким чином функції після неї задається список параметрів в квадратних дужках. Наприклад, для взятої в якості прикладу функції її можна задати і використовувати таким чином:

Function [{x, n}, x ^ n]

Function [{x, n}, x ^ n]

% [2,3]

8

Чисту функцію можна легко перетворити в звичайну функцію користувача, що показує наступний приклад:

fun = Function [{x, n}, x ^ n]

Function [{x, n}, x ^ n]

fun [2,3]

8

Анонімні функції

Гранично компактну форму завдання функцій мають так звані анонімні функції. Вони не мають ні назви, ні звичайного визначення і задаються тільки виразами спеціального виду. У цьому виразі замість змінних використовують позначення # (для однієї змінної), # 1, # 2, ... для ряду змінних. Завершується тіло функції символом &. Якщо необхідно обчислити функцію, то після її запису в квадратних дужках вказується список фактичних параметрів.

Для нашого прикладу така функція виглядає так:

 # 1 ^ # 2 & [2,3]

8

 # 1 ^ # 2 & [y, z]

y^z

За допомогою анонімних функцій неважко створити звичайні функції користувача:

f [x_, y _] = # 1 ^ # 2 & [x, y]

x^y

f [2,3]

8

Незважаючи на те, що застосування анонімних функцій відкриває можливості до компактного завданням багатьох функцій, ця форма завдання функцій чи цікава для більшості читачів; вони напевно віддадуть перевагу нехай трохи довший, але абсолютно очевидне завдання функцій в інших формах.

Лекція № 13. Графічні об’єкти інтерфейсу користувача та засоби введення-виведення

Введення-виведення в системі Mathematica організовано за допомогою інтерфейсного процесору FrontEnd. Додатково у системі реалізовано ряд додактових функцій введення-виведення:

- Input[] – зупиняє роботу системи і повертає значення виразу яке буде введено в діалоговому вікні (використовують для організації діалогового введення);
- Input[«коментар»] – працює аналогічно попередній функції, додатково виводить у діалогове вікно «коментар»;
- InputString[ ] – зчитує данні і зберігає їх у формі символьного рядка;
- InputString[«коментар»] – працює аналогічно попередній функції, додатково виводить у діалогове вікно «коментар»
- StylePrint[expr] – створює в поточному документі нову комірку зі стилем за замовчуванням і заносить в нього вираз expr;
- StylePrint[expr, «style»] – створює у поточному документі нову комірку зі стилем style і заносить в неї вираз expr;
- Print[expr] – виводить на екран дисплея значення виразу expr, разом з функцією Input[] може використовуватись для организації діалогу;
- Print[«prompt», expr] – виводить на екран дисплея текстовий коментар у лапках, а згодом – значення виразу expr.

Цих функцій достатньо для організації найпростішого діалогу з программою. У лістнингу 1 показано найпростіший приклад організації діалогу. У данному випадку обчислюється довжина кола за значенням радіуса R.

```
R=Input[“Input radius R”]
L:=2*.Pi*R
Print[“Circuit length =”, L]
```

Лістинг 1

Після виконання функції Input[] з’являється діалогове вікно у центрі екрану. У вікні виводиться запит, який вказаний у лапках як параметр функції Input[]. Після введення потрібного значення (у загальному прикладі це радіус кола) функція Input[] повертає введене значення і воно присвоюється змінній R. Після цього функція Print[] виводить на екран обчислене значення довжини кола з коротким коментарем.

Формат виведення даних

В програмі Mathematica реалізовано ряд функцій для налаштування формату представлення даних. Найчастіше використовують такі функції:

- AccountingForm[expr] – виодить всі числа, що містяться в виразі expr, у бухгалтерскій формі представлення;
- CForm[expr] – виводить визраз expr у форматі мови C;
- EngineeringForm[expr] – виодводить дійсні числа виразу expr в інженерній формі;
- FortranForm[expr] – виводить вираз expr в формі, принятої для мови Фортран;

- FullForm[expr] – виводить повну форму виразу expr без використання спеціального синтаксису;

- InputForm[expr] – виводить вираз expr у вхідній формі;

- NumberForm[expr, n] – виводить вираз expr у формі дійсного числа з точністю до n цифр;

- OutputForm[expr] – виводить враз expr в стандартній вихідній формі системи Mathematica;

- ScientificForm[expr] – виводить вираз expr у науковому форматі;

- TeXForm[expr] – виводить вираз expr у формі мови TeX, яка орієнтована на верстання текстів з математичними формулами;

- TextForm[expr] – виводить вираз expr в звичайному текстовому форматі;

- TreeForm[expr] – виводить вираз expr з демонстрацією різних рівнів виразу.

У лістингу 2 наведено приклади використання різних форм виведення.

```
AccountingForm[30*10^15]          30000000000000000
BaseForm[55434,16]                d88a16
CForm[x^2+3*x+x]                  4*x + Power(x,2)
ColumnForm[{a,b,c}]               a
                                  b
                                  c
EngineeringForm[N[12*10^29]]      1.2 10^30
Format[Exp[x^2]/a]                 x^2
                                   e
                                  ---
                                   a
FortranForm[Exp[x]^2/a]           E**(2*x)/a
HoldForm[Exp[x]^2/a]               (x)^2
                                  (e  )
                                  ------
                                    a
NumberForm[N[Exp[2]],15]          7.38905609893065
OutputForm[Exp[x]^2/a]             2 x
                                   E
                                  ----
                                   a
TeXForm[Exp[x]^2/a]               \frac{e^{2\,x}}{a}
ScientificForm[12*10^5]           1200000
```

Лістинг 2

У лістингу 3 наведено ще кілька прикладів використання різних форм виведення.

```
FullForm[Exp[x]^2/a]
Times[Power[a, -1], Power[E, Times[2, x]]]
TreeForm[Exp[x]^2/a]
Times[|            , |                    ]
      Power[a, -1]  Power[E, |           ]
                             Times[2, x]
PaddedForm[(x^3+2*x^2+3*x-1)/(x-1),3]
                          2       3
  -1 +    3 x +    2 x     + x
-------------------------------
               -1 + x

PrecedenceForm[12*b/c,5]
     12 b
a + (--)
      c
SequenceForm[Exp[x]^2/a]
 2 x
E
- -
 a
TableForm[{{"x","y"},{1,2},{3,4},{5,6}}]
x   y
1   2
3   4
5   6
Prefix[f[x^2]]
     2
f@ (x )
Unevaluated[Exp[x^(a/b)]/x/a]
                 a/b
            Exp[x   ]
Unevaluated[----]
              x a
```

Лістинг 3

Елементи графічного інтерфейсу

Засоби Mathematica дозволяють створювати об'єкти графічного інтерфейсу користувача GUI (Graphic User Interface) ноутбуків, що робить останні набагато наочнішими і зручними в роботі.

Слайдери однокоординатні

Слайдери дозволяють плавно або дискретно змінювати значення певної змінної. На рис. 1 показано створення трьох слайдерів функцією Slider з різними варіантами значень параметрів, лістинг 4.

```
Slider[x]                Slider[x,{x_min,x_max}]    Slider[x,{x_min,x_max,dx}]
Slider[x,{{e1,e2,...}}]                             Slider[x,{{{e1,w1},{e2,w2},...}}]
```

Лістинг 4

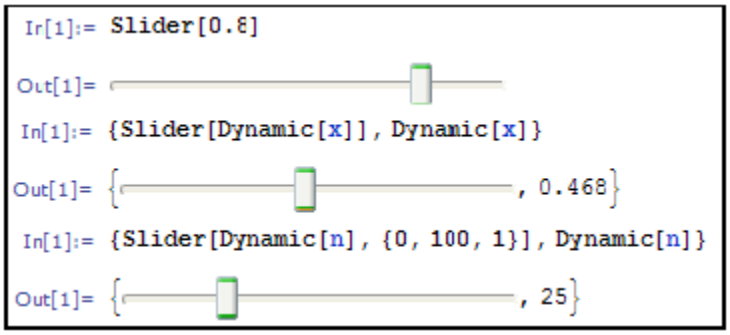


Рис. 1. Типи слайдерів

У першому випадку (верхній слайдер) параметр (число 0,8) визначає положення його движка. При переміщенні движка буде повернено значення в діапазоні від 0 до 1. Другий слайдер (у центрі) дозволяє змінювати значення змінної x. Щоб зробити значення змінної динамічно доступним в усьому ноутбуці, змінна є динамічною, для її створення використовують функцію Dynamic [x]. Типово діапазон значень змінної становить від 0 до 1. Третій слайдер дозволяє розширити діапазон значень слайдера від 0 до 100 з кроком 1.

Двокоординатні слайдери

Для побудови поверхонь, які описують функціями двох змінних використовують двокоординатні слайдери. Їх створюють функцією Slaider2D, лістинг 5.

Slider2D[{*x*,*y*}]
Slider2D[*pt*,{*min*,*max*}]
Slider2D[*pt*,{{x_{min},y_{min}},{x_{max},y_{max}}}]
Slider2D[*pt*,{{x_{min},y_{min}},{x_{max},y_{max}},{*dx*,*dy*}}]
Slider2D[Dynamic[*pt*]]
Slider2D[*pt*,{*min*,*max*,*d*}]

Лістинг 5

Приклад створення двокоординатного слайдера представлений на рис. 2.

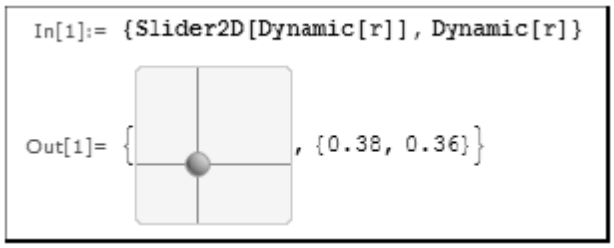


Рис. 2. Двокоординатний слайдер

Движок такого слайдера може переміщатися мишею в будь-якому напрямку. Слайдер повертає два числових значення залежно від положення движка. Ці значення можна використовувати для обчислення функцій двох змінних і побудови графіків поверхонь, тривимірних фігур, параметрично заданих графіків і т.д.

Елемент графічного інтерфейсу користувача CheckBox

Нерідко обчислення необхідно проводити за умови встановлення деяких опцій, для чого зручно використовувати елемент CheckBox, який створюють функцією CheckBox[], лістинг 6.

Checkbox[*x*] Checkbox[Dynamic[*x*]]
Checkbox[*x*,{*val*1,*val*2}] Checkbox[*x*,{*val*1,*val*2,*val*3,…}]

Лістинг 6

Приклади застосування функції CheckBox[] показано на рис. 3.

```
In[1]:= {Checkbox[False], Checkbox[True]}

Out[1]= {☐, ☑}

In[1]:= {Checkbox[1, {1, 2, 3}], Checkbox[2, {1, 2, 3}], Checkbox[3, {1, 2, 3}]}

Out[1]= {☐, ☑, ■}
```

Рис. 3. Елемент графічного інтерфейсу користувача CheckBox

Локатор

Локатор – об'єкт, який має форму точки на графічному вікні та повертає її координати. Цей об'єкт представлений зафарбованим кружком в середині перехрестя, що має світлу крапку в середині. Локатор переміщують мишею. Для його створення використовують функцію, лістинг 7.

Locator[{*x*,*y*}] Locator[Dynamic[*pos*]]
Locator[{*x*,*y*},*obj*] Locator[{*x*,*y*},None]

Лістинг 7

Приклад створення локатора показано на рис. 4.

```
In[1]:= DynamicModule[{p = {0.5, 0.5}},
         {Graphics[Locator[Dynamic[p]], PlotRange → 2], Dynamic[p]}]

Out[1]= {                    , {0.126667, 1.03333}}
```

Рис. 4. Локатор

Локатори часто застосовують для побудови графіків по точкам.

Функції керування мишею

Іноді виникає необхідність у визначенні координат курсору миші, для чого використовують функцію MousePosition. Приклад її застосування показано у лістингу 8.

```
In[1]:= Dynamic[MousePosition[]]

Out[1]= {229, 521}
```

Лістинг 8

Координати миші виводятьу форматі цілих чисел. Додавання функції MousePosition в список параметрів функції Dynamic дозволяє вивести координати курсору миші у вигляді списку поточних координат.
Функція Opener [x] або Opener [Dynamic [x]] реагує на кожен клік миші по чорному трикутнику. Наприклад, команда {Opener [Dynamic [x]], Dynamic [x]} виводить в рядок виводу список:
{► ,False}
Однак якщо клікнути по ньому мишею з'явиться список
{▼,True}
Функція
Toggler[x] Toggler[Dynamic[x]] Toggler[x,{val1,val2,…}]
Toggler[x,{val1_>pict1,val2_>pict2,…}] Toggler[x,vlist,dpict]
забезпечує можливість реакції на неодноразові натискання клавіши миші. Наприклад,
Toggler[1,{a,b,c,d}]
1
спочатку виводиться 1. Однак, після натиснення кнопки миші 4 рази, виводяться символи a, b, c і d.

Кнопка з написом

Нерідко деякі дії повинні виконуватися при натисканні кнопки з написом. Для створення такої кнопки використовують функцію Button [label, action]. У списку її параметрів вказують рядок з ім'ям кнопки і вираз, який виконують за умови активізації кнопки мишею. Приклад застосування функції Button, з виведення факторіалу числа 10, зображено у лістингу 9.

```
In[1]:= Button["Click Here", Print[10!]]

Out[1]= [Click Here]

3 628 800
```

Лістинг 9

Маніпулятор

Маніпулятор – об'єкт, схожий на слайдер, але має більші функціональні та візуальні можливості. Маніпулятор створюють за допомогою функції: Manipulator[*x*], Manipulator[Dynamic[*x*]], Manipulator[*x*,{*xmin*,*xmax*}], Manipulator[*x*,{*xmin*,*xmax*,*dx*}]. Відмінною особливістю маніпулятора є характерна кнопка у вигляді сірого прямокутника зі знаком «+ » в ній. При активізації мишею цієї кнопки під слайдером маніпулятора з'являється панель з органами керування слайдером (див. рис. 5 , приклад зверху). За допомогою органів (кнопок) керування маніпулятора можна запустити його і забезпечити автоматичне переміщення движка слайдера, можна змінити напрямок і швидкість переміщення слайдера, здійснити його зупинку. За умови наближення слайдера до початкової та кінцевої точок у движка з'являється характерна тінь.

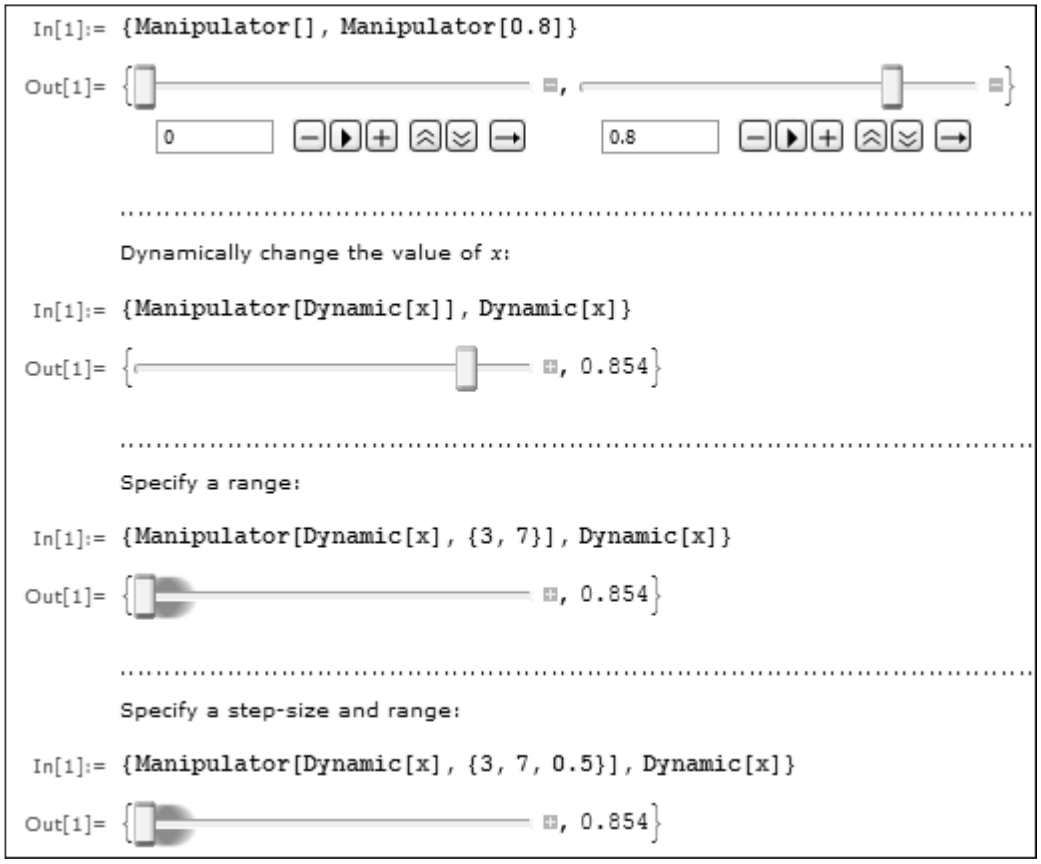


Рис. 5. Маніпулятор

Задавач кута повороту радіус-вектора

У деяких випадках, наприклад, при побудові графіків функцій в полярній системі координат, потрібен об'єкт, що задає кут повороту радіус-вектора. Такий об'єкт – задавач кута повороту – задається функцією angularSlider[]. На рис. 6 представлений приклад застосування цієї функції.

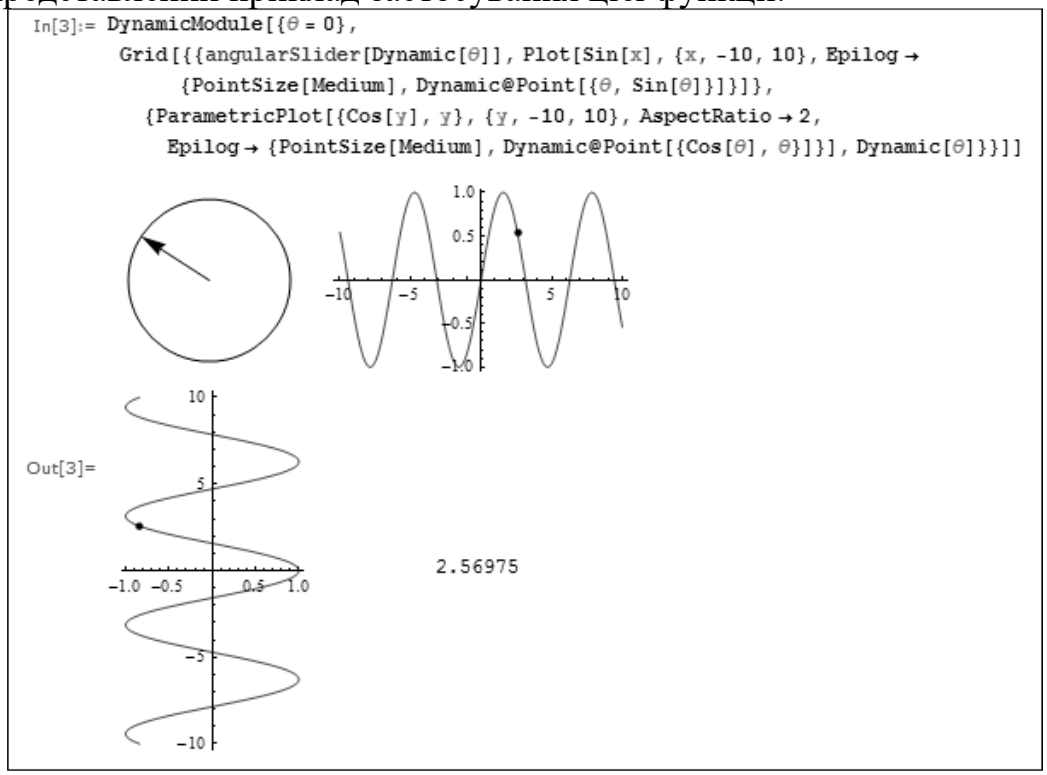


Рис. 6. Задавач кута повороту радіус-вектора

Задавач є графічним об'єктом у вигляді кола, всередині якого розміщено радіус-вектор. Він може обертатися в ту чи іншу сторону за допомогою миші, що веде до зміни задаваного задатчиком кута повороту. У прикладі, показаному на рис. 6, побудовані проекції кінця радіус-вектора на координатніій осі. Ці проекції, як відомо, є синусоїдальні функції. Кут, що відповідає поточному положенню радіус-вектора, позначений крапками на графіках синусоїдальних функцій.

Випадаюче меню

Випадаюче меню – ще один широко застосовуваний об'єкт для побудови графічного інтерфейсу. Його створюють функцією ActionMenu[*name*,{*lbl*1:>*act*1,*lbl*2:>*act*2,…}]. Параметрами функції є рядок з написом на кнопці і список назв позицій меню і дій, виконуваних при активізації відповідних позицій меню. У прикладі, показаному на рис. 7, розраховують значення факторіалів 4!, 7! і 10!.

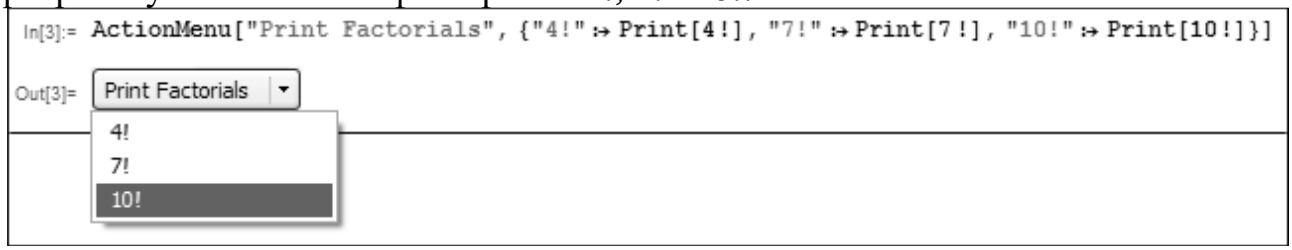


Рис. 7. Приклад створення меню

Панель введення даних

Панель введення виразів використовують для інтерактивного введення довільного виразу, наприклад, для побудови його графіка. Вона вводиться функцією Panel.

Panel[*expr*], Panel[*expr*,*title*], Panel[*expr*,*title*,*pos*],
Panel[*expr*,{*title*1,*title*2,…},{*pos*1,…}], Panel[].

Приклад задання панелі введення виразу і побудови його графіка показаний на рис. 8.

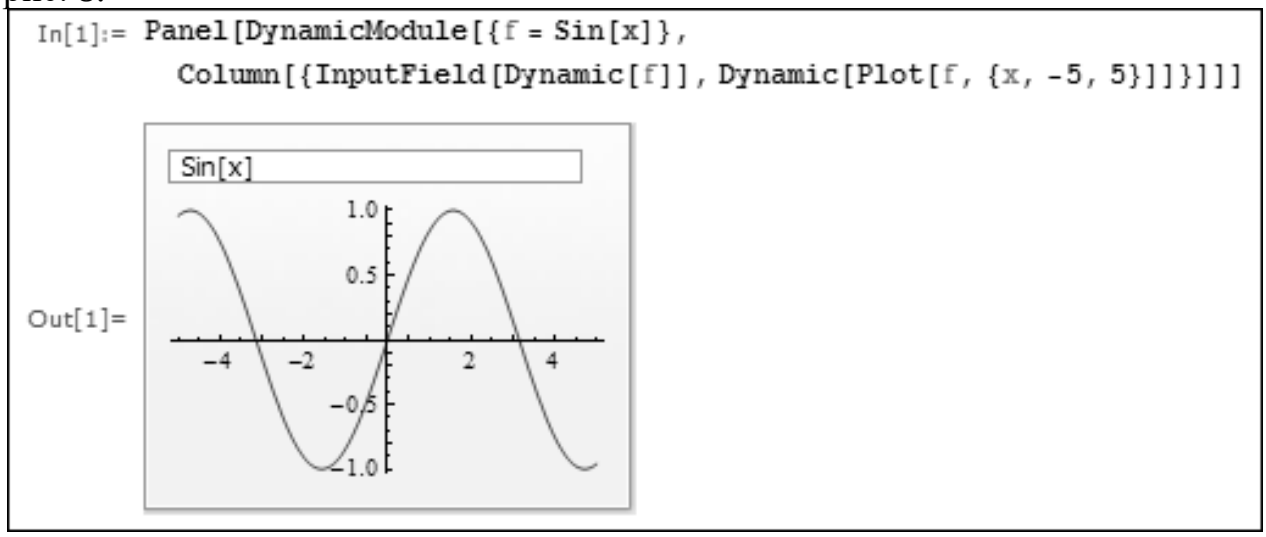


Рис. 8. Приклад створення панелі введення даних

Зверніть увагу на те, що в панелі введення за замовчуванням з'являється вираз (у нашому випадку Sin [x]), який як параметр функції DynamicModule всередині списку параметрів функції Panel. Цей вираз використовують для побудови графіка в області виведення.

Радіокнопки та меню налаштування

Функція RadioButton [x, val] створює так звану радіокнопку у вигляді кола. Результат виконання функції можна використовувати для програмного введення тієї або іншої дії. Приклади застосування радіокнопки дано на рис. 9.

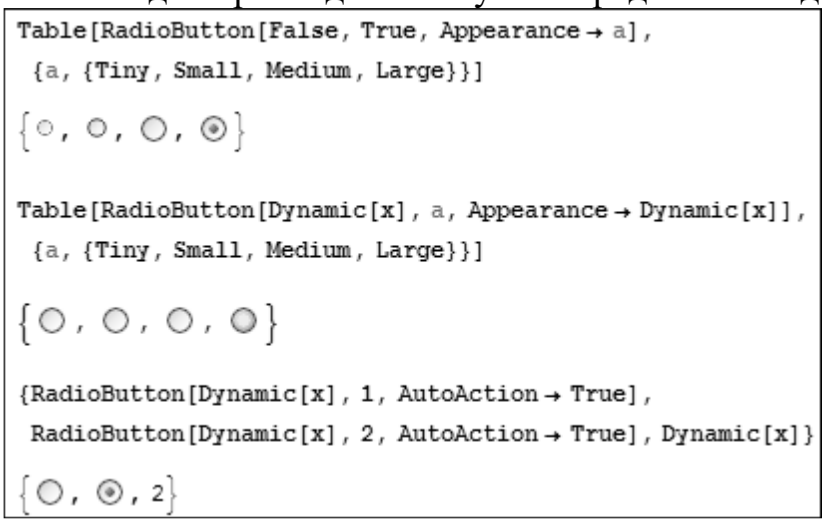


Рис. 9. Елемент RadioButton

Ще одна функція в ряді форматів запису задає побудову меню налаштувань має такі формати запису: SetterBar[*x*,{*val*1,*val*2,…}], SetterBar[Dynamic[*x*],{*val*1,*val*2,…}], SetterBar[*x*,{*val*1_>*lbl*1,*val*2_>*lbl*2,…}]. Приклад її представлено на рис. 10.

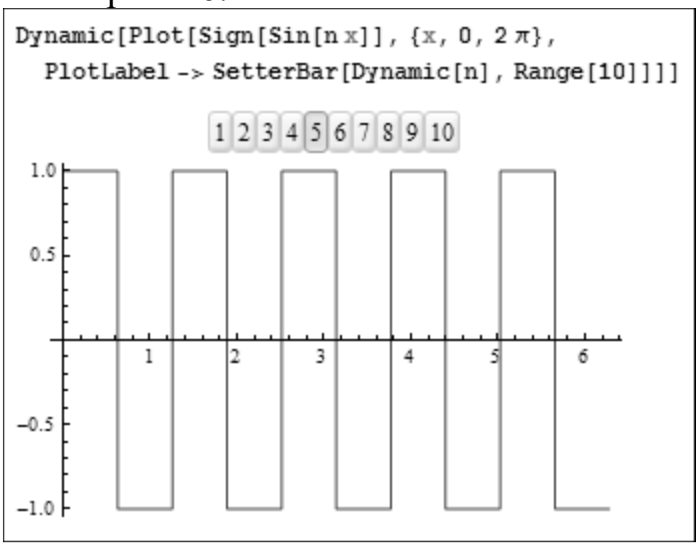


Рис. 10. Створення меню налаштувань функцією SetterBar[]

У цьому прикладі можливо, активізувавши ту чи іншу позицію меню з цифрою, задати частоту меандра.

Ще один варіант побудови меню налаштувань задає функція:Setter[*x*, *val*], Setter[Dynamic[*x*], *val*], Setter[*x*, *val*, *label*], Setter[*x*, {*val*1, *val*2,…}, *label*]. На

рис. 11 показано приклад, в якому кнопки меню налаштувань забезпечують установку відповідного розміру графіка функції $\sin(x)^3$.

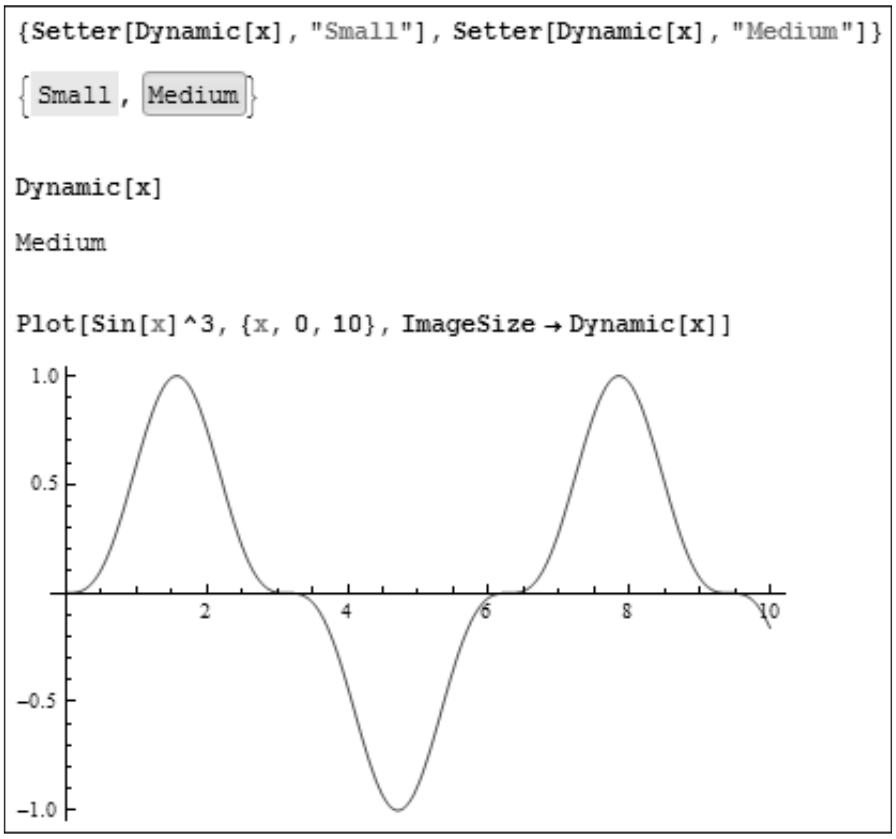


Рис. 11. Приклад використання функції Setter[]

Слайдер зміни кольору

Для зміни кольору об'єктів графічного інтерфейсу зручно використовувати слайдер, котрий задають наступною функцією:ColorSlider[*color*], ColorSlider[Dynamic[*color*]], ColorSlider[].

Приклади застосування цієї функції показано на рис. 12.

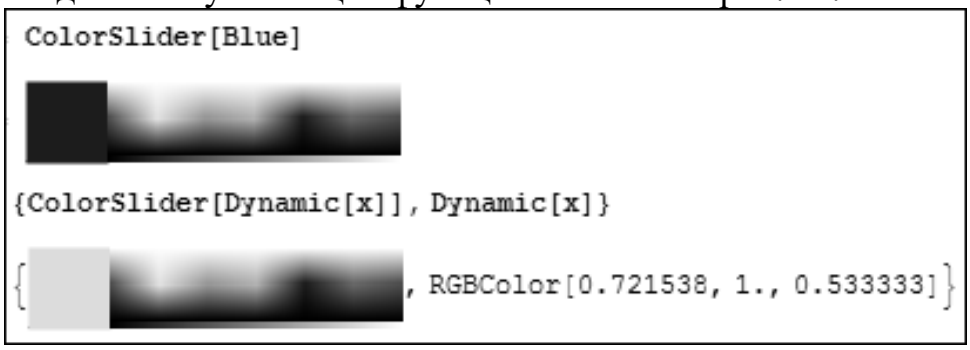


Рис.12. Слайдер зміни кольору

Колір першого квадрату – поточний колір. Ліворуч від нього – панель вибору кольору. Для вибору кольору досить встановити курсор миші на потрібний колір панелі і натиснути ліву клавішу миші. Колір контрольного квадрату стане аналогічним обраному.

Пусковий механізм

Функція Trigger імітує роботу пускового механізму. Вона має кілька форм запису: Trigger[Dynamic[u]], Trigger[Dynamic[u],{umin,umax}], Trigger[Dynamic[u],{umin,umax,du}], Trigger[Dynamic[u],{umin,umax},ups]. Результатом виконання функції, показаним на рис. 13, є панель керування пусковим «механізмом».

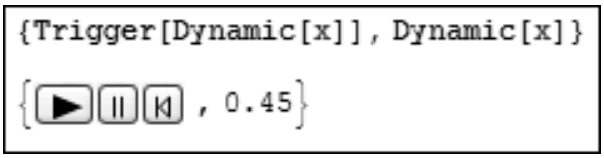


Рис. 13. Пусковий механізм

При натисканні на кнопку пуску (великий трикутник) значення динамічної змінної x починає змінюватися від 0 до 1 протягом декількох секунд. Потім ця зміна припиняється. На панелі є також кнопки зупинки змінних і повернення до 0 (Reset).

Функції позначення додаткової інформації на графіках

Функцію ClickPane[image,func], ClickPane[image,{{xmin,ymin},{xmax,ymax}},func] використовують для позначення точки на малюнку image із застосоуванням виразу func. Найчастіше функцію ClickPane використовують для вибору курсором миші деякого місця малюнка, в яке необхідно клацанням миші помістити якийсь графічний об'єкт, наприклад точку, вістря стрілки, коло і т.д. На рис. 14 показано приклад застосування функції для позначення екстремуму синусоїдальної функції за допомогою стрілки, що виходить від напису «Екстремум».

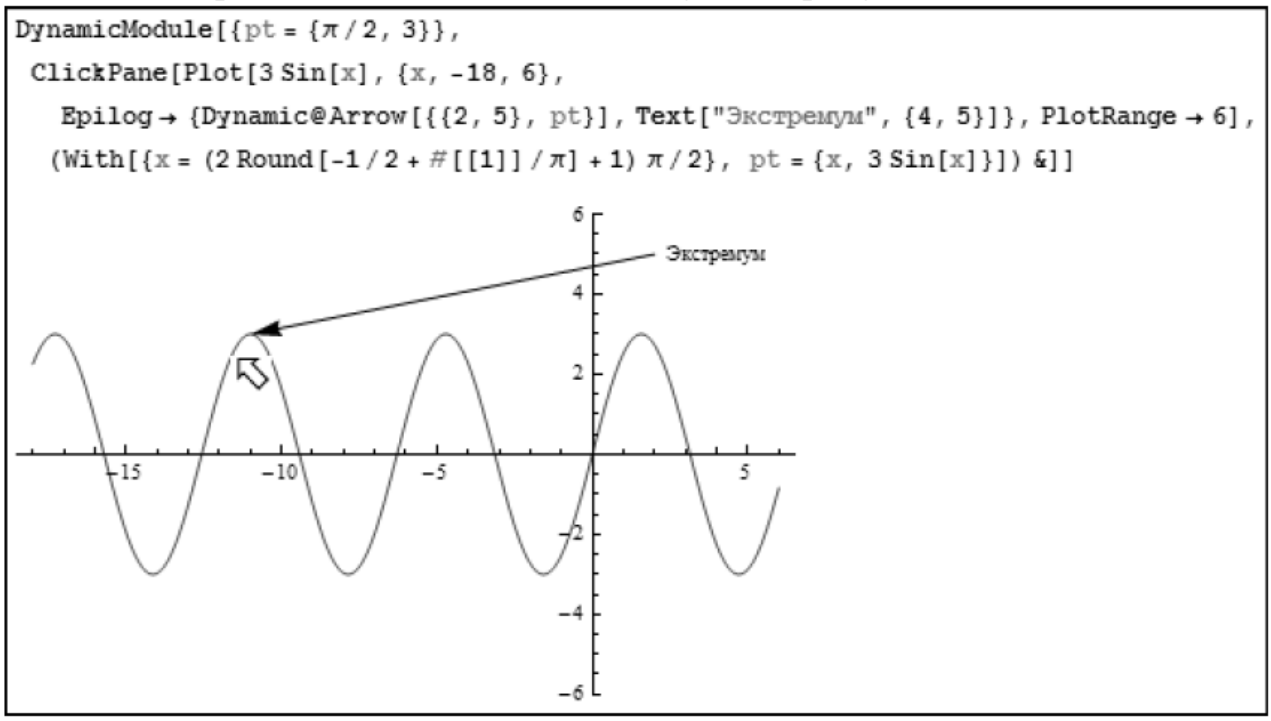


Рис. 14. Приклад використання функції ClickPane[]

Фунція Tooltip[*expr, label*] виводить вираз expr і заміняє його написом label, якщо курсор миші встановлено на вираз expr. На рис. 15 показано найпростішу реалізацию цієї функції.

```
In[1]:= Tooltip[2 + 3, "Це сума 2+3!"]

Out[1]= 5
```

Це сума 2+3!

Рис. 15. Приклад використання функції Tooltip[]

Якщо на результат, виведений у рядку навести курсор миші, буде показано текстовий напис «Це сума 2 +3!», яка була задана як параметр label.

В іншому прикладі, рис.16, показано застосування цієї функції для розпізнання однієї з двох кривих, побудованих одним кольором.

```
Plot[Tooltip[{Sin[x]^3, Sin[x]}], {x, 0, 10}]
```

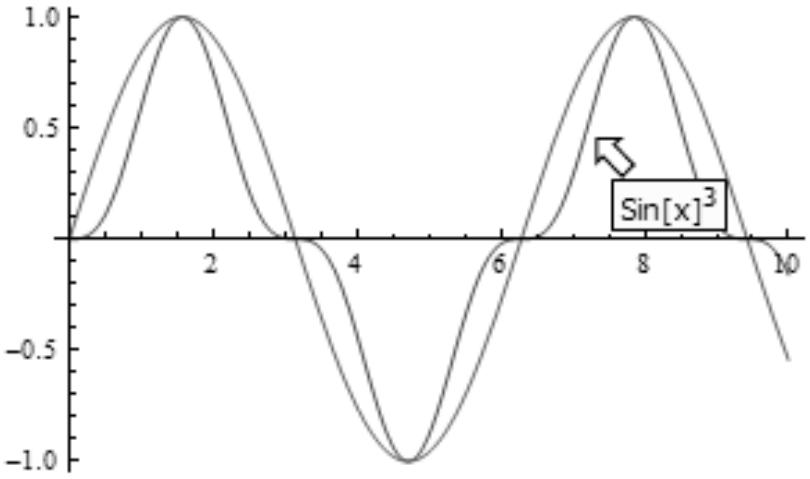


Рис. 16. Розпізнавання графіка функцією Tooltip[]

Для розпізнавання графіка функцією Tooltip[] необхідно навести курсор миші на одну з кривих.

На рис. 17 показано приклад застосування функції Tooltip[] для визначення точного значення ординати точкового графіка синусоїди під час наведення на потрібну точку курсору миші.

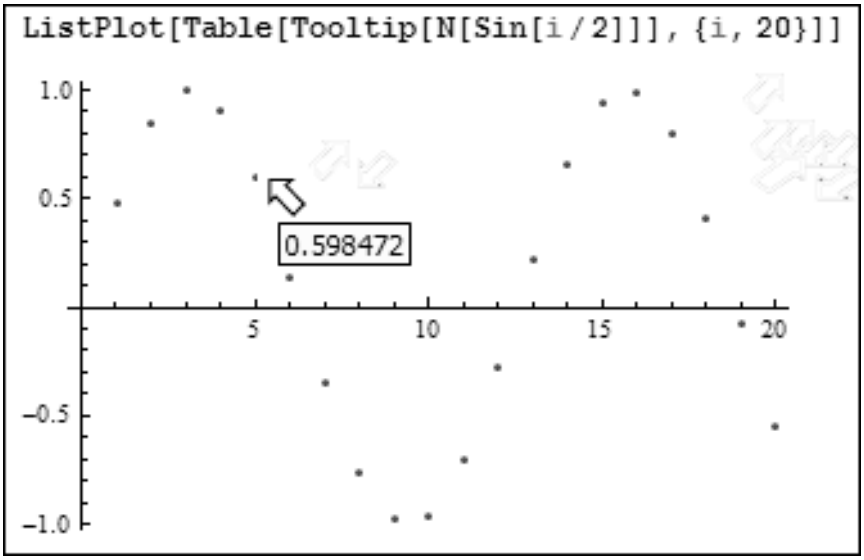


Рис. 17. Визначення значення ординати точкового графіка функцією Tooltip[]

Повідомлення, активізовані курсором

Функція PopupWindow[] забезпечує контроль над наведенням курсора мишіна поставлений в ній об'єкт. Якщо курсор наведений на об'єкт, то він модифікується і з'являється панель з заданим повідомленням. Наприклад, на рис. 18 об'єктом є коло, що в момент наведення на нього курсору миші змінює забарвлення. Потім після появи панелі з написом «Це диск» коло відновлює забарвлення.

```
PopupWindow[Graphics[Disk[],ImageSize->50],"Це диск"]
```

Рис. 18. Повідомлення, активізовані курсором

Виведення меню і вибір його позиції

Функція MenuView[] забезпечує створення меню з випадаючими позиціями під номерами, які відповідають значенню змінної n. У прикладі рис. 19 обчислюють функцію tan(nx) для різних n.

Виведення меню з вкладеннями та навігація між ними

Функція TabView[] виводить меню з вкладеннями, перехід між якими здійснюють натисканням лівої кнопки миші. Приклад застосування цієї функції показано на рис. 20. У даному випадку у вікні об'єкта будують малюнок, який створено клітинним автоматом за допомогою функції CellularAutomaticc[] з різним значенням параметра r (rule). Можливі значення r вибирають зі списку активізацією відповідної вкладинки мишею.

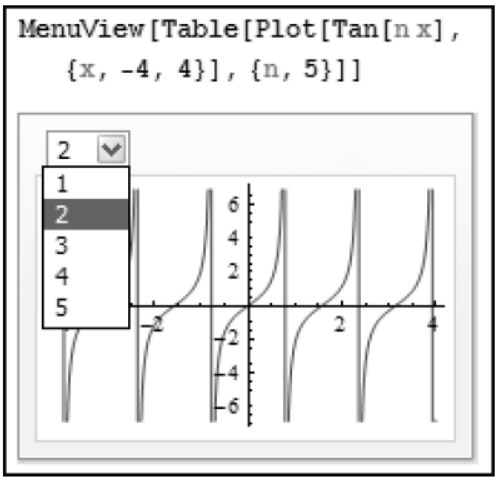


Рис. 19. Приклад використання функції MenuView[]

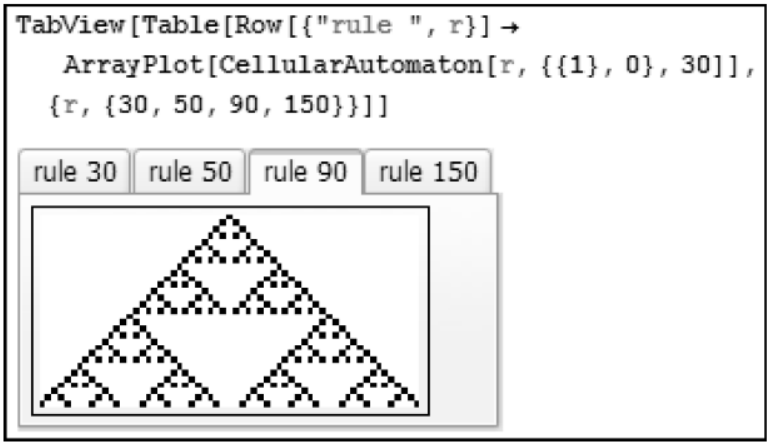


Рис. 20. Приклад використання функції TabView[]

Слайд- меню

Слайд-меню використовують для створення презентацій. Для створення слайд-меню використовують функцію SlaidMenu[]. Роботу з функцією показано на рис. 21. При активізації кнопок із зображеннями трикутника можна вибирати символ зі списку. Перемикання може йти в будь-яку сторону, а також відразу на початок або в кінець списку.

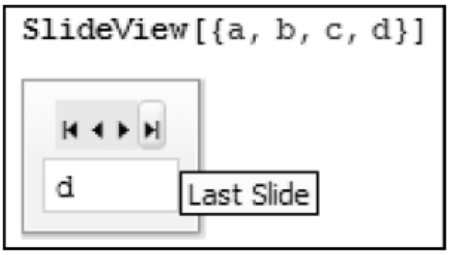


Рис. 21. Приклад використання функції SlaidMenu[]

Створення вікон інтерактивного діалогу з користувачем

У програмі Mathematica передбачено можливість створення діалогових вікон на основі пакету розширення GUIKit . На рис. 22 показано виклик цього пакету і вікна для демонстрації арифметичних операцій з двома числами що вводяться. Для створення навіть такого простого прикладу потрібна програма, що містить близько шестидесяти рядків . Вона представлена файлом Calculator.m.

Лекція № 14. Правила написання ефективного коду та відлагодження програм

Як і будь-яка програмна система, Mathematica призначена для автоматичного оброблення інформації – в нашому випадку математичної. Найчастіше ця обробка зводиться до обчислень і їх візуалізації, наприклад, графічної.

Інформація може бути представлена в самому різному вигляді: числами, математичними формулами, текстовими символами та іншими елементами інформації.

На рівні своєї вхідної мови програмування та спілкування з користувачем система Mathematica оперує з трьома основними класами даних:

• чисельні, що представляють числа різного виду;

• символьні, що представляють символи, тексти і математичні вирази (формули);

• списки - дані у вигляді безлічі однотипних або різнотипних даних.

Кожен з цих класів даних, в свою чергу, має ряд спеціальних, більш приватних типів даних.

Робота з цілими числами

У Mathematica використовуються цілі числа з різним підставою і десятковічисла з плаваючою точкою (їх часто називають числами з плаваючою комою), представлені в різній нотації. З цілих чисел широко використовуютьсядвійкові числа з основою 2, восьмеричні з основою 8, десяткові з основою 10 і шістнадцяткові числа з основою 16. Найбільш поширеними є десяткові числа (DECIMAL). Кожен розряд таких чисел має представлення, задане однією з арабських цифр: 0, 1, 2, ... 9. Ваговий коефіцієнт старшого розряду відносно попереднього дорівнює 10.

Для обчислення чисел з довільним підставою використовується конструкція **Основа ^^ Число**.

Число повинно бути записано за правилами запису чисел з відповідною основою. Для основ більше 10 для позначень значень чисел використовуються літери від a до z. Найбільш відомими з чисел з розрядністю більше 10 є шістнадцятиричні числа (HEX – від слова hexagonal). Розряд таких чисел може мати значення:

HEX 0 1 2 3 4 5 6 7 8 9 a b c d e f

DECIMAL 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Старший розряд має ваговий коефіцієнт відносно заданого розряду, рівний16.

Приклади завдання шістнадцятирічного і довічного чисел:

16 ^^ 123abcde

305839326

2 ^^ 1010111

87

Для представлення чисел з довільною основою n (до 32) використовується функція **BaseForm [expr, n]** – повертає вираз expr в формі

BaseForm [87,2]

1010111_2

BaseForm [305839326,16]

$123abcde_{16}$

Для отримання списків чисел різних цілих чисел служить функція **IntegerDigits [n, b, len]**, де n – число, b – основа, і len – довжина числової послідовності, що доповнюється зліва нулями. Параметри b і len можуть бути відсутні. Приклади застосування цієї функції представлені нижче:

IntegerDigits [1234]

{1,2,3,4}

IntegerDigits [1234,2]

{1,0,0,1,1,0,1,0,0,1,0}

IntegerDigits [10!, 8]

{1,5,6,5,7,4,0,0}

IntegerDigits [10!, 10,12]

{0,0,0,0,0,3,6,2,8,8,0,0}

Насправді, застосовувані в системах символьної математики способи подання чисел більш компактні, ніж наведений найпростіший, але це не змінює суті головного: чим більше кількість цифр числа, тим більше пам’яті відводиться на його зберігання. Особливо багато уваги приділено компактному зберігання чисел в системі Mathematica, що дозволило в кілька разів знизити витрати пам'яті на зберігання великих числових масивів і зменшити час роботи з ними.

Характерним прикладом роботи з цілими числами великої розрядності є обчислення факторіала n! = 1 * 2 * 3 * ... * n (при 0! = 1):

1000! / 950!

```
28773134312001300070246880730666449532231680786014125 8 \
46038434211648017743491487747660401226637845326344373 4 \
57983357703768094811080359936000000000000
```

(20! +5!) / 22!

$$\frac{20274183401472001}{9366672731480064000}$$

Зверніть увагу, що знак «\» в кінці рядка виводу першого прикладу означає переклад наступних символів на новий рядок.

Цілочисельні дані (Integer) – це цілі числа, наприклад 1, 2 або 123, які надаються системою без похибки і обмеження розрядності.

Більш того, арифметичні операції над цілими числами система виконує також без похибок і без обмеження числа цифр:

+12345678912345678912345678 ^ 2

152415787806736785461057782810547205156226207501905 21

% / +12345678912345678912345678

12345678912345678912345678

10000000000000000000000000000000000000000000000000 + 123

10000000000000000000000000000000000000000000000123

10000000000000000000000000000000000000000000000000-1

99999999999999999999999999999999999999999999999999

Раціональні дані задаються відношенням цілих чисел, наприклад 123/567, і також представляють результат точно:

1000000/3000000

$\frac{1}{3}$

(124-1) / (455 + 1)

$\frac{41}{152}$

Робота з числами дійсного типу

Дійсні числа можуть мати різну форму, наприклад 123.456, 1.23456 10 ^ 2, 12345.6 10 ^ 2 і т.д. У загальному випадку вони містять мантиссу з цілою і дробовою частиною і порядок, який вводиться як ступінь числа 10. Як правило, дійсні числа в системах символьної математики можуть мати мантиссу з будь-яким, але кінцевим числом знаків.

Пропуск між мантиссой і порядком еквівалентний знаку множення *:

23.456 * 10 ^ 100

2.3456×10^{101}

10 ^ -100

1/10000000000000000000000000000000000000000000000000 \
0000000000000000000000000000000000000000

10. ^ - 100

1×10^{-100}

Як прийнято в більшості мов програмування, ціла частина мантиси відділяється від дробової частини точкою, а не коми. Знак «\» означає переклад рядки (частини числа). Його, до речі, можна використовувати і в осередках введення.

Mathematica виробляє операції з числами, спочатку як з цілими. Однак установка значка розділової точки означає, що число розглядається як дійсне. Наприклад, число 1 - ціле число, але 1. - вже дійсне число. Для подання виразу expr в формі дійсного числа використовується функція **N [expr]** або **N [expr, число_ціфр_результата]**

1/3

$\frac{1}{3}$

1./3

0.333333

N [1/3]

0.333333

N [2 * Pi, 50]

6.2831853071795864769252867665590057683943387987502

N [2 * Pi, 500]

6.2831853071795864769252867665590057683943387987502116419498891846
15\63281257241799725606965068423413596429617302656461329418768921
910116\44634507188162569622349005682054038770422111192892458979
098607639288\57621951331866892256951296467573566330542403818291
297133846920697220\90865329642678721452049828254744917401321263
117634976304184192565850\81834307287357851807200226610610976409330
427682939038830232188661145\40731519183906184372234763865223586
210237096148924759925499134703771\50544978245587636602389 83

Дійсні числа завжди мають деяку похибку представлення результатів через неминуче округлення їх і існування так званого «машинного нуля» - найменшого числа, яке сприймається як нуль.

Mathematica має дві системні змінні, що дозволяють вивести значення максимально і мінімально можливих значень чисел, з якими оперує система: **$MaxMachineNumber** та **$ MinMachineNumber**.

Символьні дані і рядки

Символьні дані в загальному випадку можуть бути окремими символами (наприклад, a, b, ..., z), рядками (strings) і математичними виразами expr (від expression - вираз), представленими в символьному вигляді. Символьні рядки задаються ланцюжком символів в лапках, наприклад «Sssss». У них можуть використовуватися такі управляючі символи для малих об'єктів:

\ n нова лінія (line feed),

\ t табуляція.

Це ілюструється такими прикладами:

«Hello my friend!»

Hello my friend!

«Hello \ nmy \ friend!»

Hello

my

friend!

«Hello \ tmy \ tfriend!»

Hello my friend!

Слід пам'ятати, що керуючі символи не виводяться принтером і не відображаються дисплеєм, а лише змушують їх виконувати призначені ними дії. Mathematica має безліч функцій для роботи з рядками, які будуть описані в подальшому.

Для запису математичних виразів використовуються як оператори, так і функції. Їх особливості будуть розглянуті трохи пізніше. А поки відразу відзначимо деякі тонкощі синтаксису системи, що використовується під час запису арифметичних операцій:

• знак множення може бути замінений пропуском;

• вбудовані функції починаються з великої літери і зазвичай повторюють своє загальноприйняте математичне позначення (за винятком тих, в назві яких є грецькі букви - вони відтворюються латиною?

ми буквами по звучанню відповідних грецьких букв);

• круглі дужки (...) використовуються для виділення частин виразів і завдання пріоритету їх виконання;

• параметри функцій задаються в квадратних дужках [...];

• фігурні дужки використовуються при завданні списків {...}.

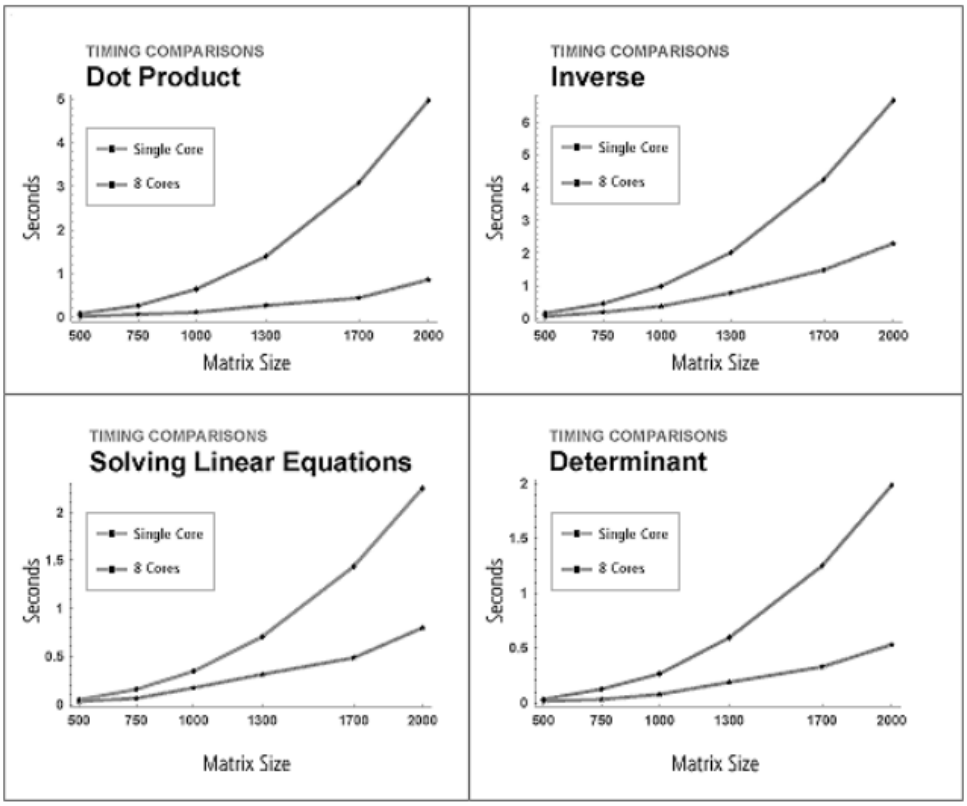
Підтримка багатоядерних мікропроцесорів

В даний час провідні виробники мікропроцесорів (корпорації Intel, AMD, IBM і ін.) Переходять на випуск нових багатоядерних мікропроцесорів (multicore processors), що реалізують методи паралельних обчислень, до цього застосовуються лише в супер? ЕОМ (зокрема кластерних). Mathematica стала першою СКМ, в деяких варіантах якої вперше забезпечена як технологія ущільнення потоків Hyper Threading, так і підтримка можливостей новітніх багатоядерних процесорів. Ці переваги реалізовані в першучергу при вирішенні задач з лінійної алгебри. У цій області найпростіше досягається розбиття процесів обчислень на окремі частини, що виконуються окремими ядрами процесора.

Інтерфейсна частина Mathematica реалізована як окремий процес, відокремлений від процесів її обчислювального ядра. Це створює діалоговий інтерфейс навіть тоді, коли багатоядерний процесор знаходиться під граничним навантаженням. Навіть при використанні двоядерного процесора Mathematica

забезпечує роботу на різних ядрах интерфейсного модуля і обчислювального ядра (kernel). В результаті час «обмірковування», традиційно значна для колишніх версій Mathematica, скорочено приблизно в 1000 разів.

На рисунку показано порівняння часів обчислень при вирішенні чотирьох найбільш характерних завдань лінійної алгебри в СКМ Mathematica на компьютері. Порівняльні результати матричних обчислень в середовищі Mathematica для ПК з одноядерним і 8-миядерним мікропроцесорами.



Лекція № 15. Команди збереження даних системи Mathematica

Хоча Mathematica орієнтована на математичні додатки, в ній досить повно представлені функції для роботи з рядками (strings). Вони можуть знадобитися як для організації виведення текстових повідомлень (наприклад написів на графіках), так і для організації текстового діалогу при розробці пакетів розширень і додатків системи. До того ж треба постійно пам'ятати, що Mathematica – система символьної математики, так що символьним перетворенням, як суто математичним, так і загальноприйнятим, в ній, природно, приділено багато уваги.

Багато функцій для роботи з рядками виконують загальноприйняті перетворення, наявні в більшості мов програмування високого"рівня. рядком є довільна ланцюжок символів, укладена в лапки, наприклад «String». Нижче представлені деякі функції для роботи з рядками:

• StringByteCount ["string"] - повертає повне число байтів, використовуваних для зберігання символів в рядку "string";

• StringDrop ["string", {m, n}] - повертає рядок "string", видаливши в ній символи від m до n;

• StringJoin ["sl", "s2", ...] або StringJoin [{ "s1", "s2", ...}] - формує рядок, що містить конкатенацію (об'єднання) зазначених рядків "s1";

• Stringlnsert ["string1", "string2", M] - вставляє рядок "string2" в рядок "string1", починаючи з позиції M від початку цього рядка (при негативному M позиція відраховується від кінця зазначеного рядка);

• StringLength ["string"] - повертає число символів в рядку;

• StringReplace ["string", "s1 ->" spl "] або StringReplace [" string ", {" s1 "->" spl "," s2 "->" sp2 ", ...}] - заміщає s1" на "spi" всякий раз, коли вони з'являються як підрядка "string";

• StringReverse ["string"] - змінює порядок символів в рядку "string" на протилежний;

• StringPosition ["string", "sub"] - повертає список з позиціями рядки "sub" в рядку "String";

• StringTake ["string", n] - повертає рядок, що складається з перших n символів рядка "String";

• StringTake ["string", -n] - повертає останні n символів з рядка "string";

• StringTake ["string", {n}] - повертає n-й символ в рядку "string";

• StringTake ["string", {m, n}] - повертає рядок з символів, розташованих в позиціях від m до n рядка "string".

Ці функції добре відомі програмістам, що працюють з сучасними мовами програмування. Велике число додаткових функцій для роботи з рядками можна знайти в додатку. Велика кількість таких функцій в мові програмування системи Mathematica вказує на його універсальний характер і великі можливості в рішенні навіть на перший погляд далеких від математики завдань. Нижче наведені приклади дії ряду функцій роботи з рядками.

Введення (In)	**Виведення (Out)**
StringByteCount ["Hello!"]	6
StringDrop ["Hello my friend!", 6]	my friend!
StringDrop ["Hello my friend!", -10]	Hello
StringDrop ["Hello my friend!", {7}]	Hello y friend!
StringDrop ["Hello my friend!", {6, 8}]	Hello friend!
Stringlnsert ["Hello friend!", "My", 6]	Hello my friend!
StringJoin ["Hello", "my"] <> "friend!"	Hello my friend!
StringLength ["Hello"]	5
StringPosition ["Hello my friend!", "E"]	{{2, 2}, {13, 13}}
StringReplace ["Hilo", "i" -> "el"]	Hello
StringReverse ["Hello!"]!	OlleH
StringTakef "Hello my friend!", 6]	Hello
StringTake ["Hello my friend!", -8]	friend!
StringTake ["Hello my friend!", {7, 9}]	my

Відзначимо ще кілька функцій, що відносяться до роботи з символами і рядками:

- FromCharacterCode [n] - повертає рядок, що складається з одного символу з кодом n;
- FromCharacterCode [{n1, n2, ...}] - повертає рядок, що складається з послідовності символів з кодами ni;
- Characters ["string"] - повертає список цілочисельних кодів, відповідних символів рядка "string";
- ToLowerCase ["string"] - виробляє рядок, в якій всі букви перетворені в нижній регістр;
- ToString [expr] - повертає рядок, відповідну формі виведення виразу expr. Опції встановлюють ширину лінії, тип формату і т. д .;
- ToUpperCase ["string"] - виробляє рядок, в якій всі букви перетворені в верхній регістр;
- Unique [] - створює новий символ з ім'ям в формі $ nnn (nnn - унікальний порядковий номер);
- Unique [x] - створює новий символ з ім'ям в формі x $ nnn (nnn – унікальний порядковий номер);
- Unique [{x, y, ...}] - створює список нових символів з унікальними іменами;
- Unique ["xxx"] - створює новий символ з ім'ям в формі xxxnnn (nnn – унікальний порядковий номер);
- Unique [name, {attrl, attr2, ...}] - створює символ із зазначеними атрибутами attri;
- UpperCaseQ [string] - повертає True, якщо всі символи рядка string є прописними буквами (з великої літери), інакше повертає False.

Приклади, наведені нижче, показують роботу з цими функціями.

Введення (In)	**Виведення (Out)**
ToCharacterCode ["Hello!"]	
{72,101,108,108,111,33}	
FromCharacterCode [{72,101, 108, 108, 111, 33}]	Hello!
ToExpression ["2 + 3 * 4"]	14
ToLowerCase ["HeLLo!"]	Hello!
ToUpperCase ["Hello"]	HELLO

Введення (In)	**Виведення (Out)**
x: = ToString [2 + 3 * 4]	
X	14
Unique []	$ 1
Unique [xyz]	xyz $ 2
Unique [xyz]	xyz $ 3
UpperCaseQ ["Hello"]	False
UpperCaseQ ["HELLO"]	True

Потоки і файли

Система Mathematica має розвинені засоби для роботи з потоками (streams) і файлами (Files). Під *потоком* розуміється безперервна послідовність даних, що циркулюють всередині комп'ютера. Обмін потоками відбувається практично безперервно, наприклад, при введенні потік вводу поступає від клавіатури в комп'ютер, при друку потік даних надходить від комп'ютера в принтер через порт принтера і т. д.

Файлом є упорядкована структура даних, що має ім'я і зберігається на будь-якому носії, найчастіше на магнітному диску. Файли можуть мати різні формати і різний тип доступу до інформації, що зберігається на них інформації. Найбільш поширені в системі Mathematica файли документів є файлами з послідовним доступом і мають текстовий формат.

Послідовний доступ означає, що інформація з відкритого файлу може бути зчитана строго послідовно від його початку до кінця, зазначеного спеціальною міткою. це нагадує зчитування з магнітофонного касети. *Текстовий формат* означає, що всі дані записані в вигляді ASCII-кодів. Отже, прочитати такий файл можна за допомогою будь-якого текстового редактора, що працює з текстами у вигляді ASCII-кодів.

Потоки і файли мають багато спільного: імена, певну структуру, необхідність відкриття перед використанням і закриття після використання. Однак якщо з файлами користувач стикається вже на початку роботи з системою (потрібно викликати файл з демонстраційним документом або зберегти його, а потім викликати інший файл), то з поняттям потоку при роботі

з системою стикатися практично не доводиться, хоча поза нашою волею потоки даних постійно течуть між комп'ютером і його периферійним обладнанням.

Спрощена робота з файлами

Перш ніж розглядати досить великі можливості системи по роботі з файлами в цілому, відзначимо спрощений прийом виклику файлу за допомогою подвійного символу «<<»:

<< filename

Ця команда зчитує файл з вказаним ім'ям filename і заносить в пам'ять комп'ютера визначення що в ньому містяться. Файл треба вказувати повністю, тобто разом з розширенням. Винятком є випадок, коли файл знаходиться в основному каталозі системи. Ця команда еквівалентна функції **Get ["filename", key]**

Для запису об'єкта (змінної, масиву, списку і т. Д.) В файл служать спрощені команди:

- expr >> filename - передає значення expr в файл з заданим ім'ям;
- expr >>> filename - додає expr в кінець файлу з заданим ім'ям.

Зазначені команди по суті є укорочені (і тому більш зручні) форми наступних функцій:

- Get ["filename", "key"] - читає файл, який закодований функцією Encode з використанням ключа "key";
- GetContext ["context '"] - завантажує файл з заданим контекстом;
- Put [exprl, expr2, ..., "filename"] - записує послідовність виразів exprі в файл з ім'ям filename;
- PutAppend [expr1, expr2, ..., "filename"] - приєднує послідовність виразів exprі до файлу з ім'ям filename.

Ще одна спрощена функція -! ! filename - виводить вміст файлу з заданим ім'ям.

Наступні приклади показують запис списку в файл C: \ ma.vat, його зчитування, потім додавання в файл ще одного списку і контроль контексту файлу:

{{1, 2,3}, {4,5,6}, {a, b, c}} >> C: \ ma.val

<< C: \ ma. val

{{1, 2, 3}, {4, 5, б), {a, b, c}} {d, e, f} >>> C: \ ma.val

<< C: \ ma. val

{D, e, f}

!! 3: \ ma.val
1, 2, 3, 4, 5, б, a, b, з d, e, f

Така форма виклику особливо зручна для виклику файлів пакетів розширень і застосувань системи. Файл вказується за правилами, прийнятим в MS-DOS. Файли пакетів застосувань мають розширення .т. Ми вже наводили приклади використання визначень, що містяться в файлах пакетів розширення системи.

Є ще ряд функцій для роботи з файлами:

• ReadList ["filename"] - читає все, що залишилися в файлі "filename" вираження і повертає їх у вигляді списку;

• ReadList ["filename", type] - читає з файлу "filename" об'єкти зазначеного типу type доткінця файлу. Повертає список лічених об'єктів;

• ReadList ["filename", {typel, type2, ...}] - читає об'єкти зазначених типів typei до кінця файлу filename;

• ReadList ["filename", types, n] - читає тільки перші n об'єктів зазначених типів types з файлу filename;

• Save ["filename", x1, x2, ...] - створює файл із заданим ім'ям filename, що містить значення змінних x1, x2, ...;

•! command - виконує задану команду операційної системи.

Припустимо, що в якомусь текстовому редакторі створений файл з повним ім'ям C: \ datas.txt в ASCII-форматі, що містить просто шість чисел з розділовими пропусками, розміщені в двох рядках і представляють масив 2x3 елемента:
1 11.2 34.5
2. 3.4 56

Тоді про структурі файлу можна судити, використовуючи команду
!! 3: \ datas.txt

1 1.2 34.5 2. 3.4 56.

Неважко помітити, що структура файлу відповідає структурі масиву. Однак зчитування файлу командою << name дає наступний результат:
<< C: \ datas. txt

380.8

Результат представляет обчисленя виразів іншого рядка файлу. Зчитування функцією ReadList без додаткового аргументу також дає помилкові результати:

ReadList ["3: \ datas.txt"]

{41.4, 380.8}

Неважко помітити, що функція сприйняла КОЖЕН рядок вмісту файлу як результат перемноження трьох чисел (пробіл на мові Mathematica означає множення). З додатковим параметром Number все числа зчітуються вірно:

ReadList ["3: \ datas.txt", Number]

{1, 1.2, 34.5, 2., 3.4, 56.}

Однак ми отримали одновимірний список - дані просто зчитуються построково.

Застосування додаткового параметра у виді {Number, Number} дає наступний результат:

ReadList ["3: .txt", {Number, Number}]

{{1, 1.2), {34.5, 2.}, {3.4, 56.}}

Правильний результат можна отримати, використовуючи опцію RecordList-> True :.

ReadList ["C: .txt", Number, RecordLists-> True]

{{1, 1.2, 34.5), {2, 3.4, 56.}}

Доля завантаження файлів пакетів Розширення (Add-On) використовують функції, що дозволяють задати контекст файлів:

- Needs ["context '", "filename"] - завантажує файл, якщо зазначений контекст відсутній в списку завантаження;
- Needs ["context s"] - завантажує файл, ім'я котрого визначається за допомогою функції ContextToFilename ["context ч"], якщо зазначений контекст відсутній в списку завантаження.

Завантаження файлів із зазначеним їх контекстів дозволяє запобігти конфліктів між різними пакетами розширення, використаними одночасно.

Уніфікованій підхід

У загальному випадку для читання даних з довільного файлу використовують наступні функції:

- OpenRead ["file"] - відкриває вказаний файл для читання й повертає відповідний б'єкт типу InputStream.
- Read [stream] - зчитує вирази із зазначеним потоку введення и повертає його.
- Read [stream, type] - зчитує один об'єкт зазначеного типу та повертає його.
- Read [stream, {type1, type2, ...}] - зчитує і повертає послідовність об'єктів

зазначених типів. В якості типу зчитуваного об'єкта можуть бути вказані:
Byte - один байт даних, представленний як цілий код,
Character - одиночний символ,
Expression – закінчений вираз Математики,
Number - ціле або дійсне число в експоненційному форматі,
Real - дійсне число в експоненційному форматі,
Record - послідовність символів, розділеніх рядком "\ n",
String - рядок, что завершується символом нового рядка,
Word - послідовність символів, розділена або рядком "\ t", або "".
Фунція Read Повертає значення EndOfFile, якщо досягнут кінець файлу.

• Close [stream] - закриває вказаний потік.

Приклад нижче зчитує перший рядок з текстового файлу і виводить її у вікно комірку ноутбука, якщо файл не пустий:

```
fileStream = OpenRead [ "З: \ sampleFile.txt"];

textLine = Read [fileStream, String];

If [textLine ≠ "EndOfFile",

Print [textLine];

];

Close [fileStream];
```

Читання і запис текстів російською мовою

Підтримка російської мови реалізована в Mathematica в повному обсязі: ви можете набирати російський текст в ноутбуці, використовувати російські символи в строкових константах, але при записі тексту в файл або читанні файлу будь-якого формату (ASCII, UNICODE) російські літери замінюються іншими символами. У зв'язку з цим, при необхідності читання тексту російською мовою з зовнішнього файлу, його необхідно перетворити в послідовність цілих кодів, відповідних російським буквах в Mathematica. Новий файл (з кодами) вже можна прочитати з допомогою стандартних процедур Mathematica, а потім декодувати в рядок. Для цього можна використовувати такі функції роботи з рядками:

• ToCharacterCode ["string"] - повертає список цілих кодів відповідних символам рядка string;
• FromCharacterCode [n] - повертає рядок, що складається з символу з кодом n;
• FromCharacterCode [{n1, n2, ...}] - созвращает рядок, що складається з послідовності символів з кодами ni;

• ToExpression [input] - повертає вираз, отримане в результаті інтерпретації рядки input.

Наприклад, ось такий код

```
s = "Правила дорожнього руху РФ";
clist = ToCharacterCode [s];
Print [clist];
```

видасть список

```
{1055,1088,1072,1074,1080,1083,1072,32,1044,1086,1088,1086,1078,1085,1086,1075,1086,32,1044,1074,1080,1078,1077,1085,1080,1103,32,1056,1060}
```

Якщо цей список присвоїти якійсь змінній як рядок (в разі, якщо він лічений з файлу), то отримати вихідний рядок можна так:

```
slist =
"{1055,1088,1072,1074,1080,1083,1072,32,1044,1086,1088,1086,1078,1085,1086,1075,1086,32,1044,1074,1080,1078,1077,1085,1080,1103,32,1056,1060} ";
clist = ToExpression [slist];
s = FromCharacterCode [clist];
Print [s];
```

При перетворенні російського тексту в список кодів необхідно пам'ятати, що новий рядок в Mathematica представлений всього одним кодом (10). Лапки "і новий рядок є службовими сиволов, тому при пошуку символу нового рядка в тексті потрібно шукати подстроку "\ n", а при пошуку лапок - підрядок "\" ".

Використання файлів інших мов програмування

З функцій для роботи з файлами особливо треба відзначити таку функцію-директиву:

• Splice ["file .mx"] - вставляє в файли на інших мовах програмування обчислені вираження системи Mathematica, які повинні бути записані в дужках виду <* і *>;

• Splice ["infile", "outfile"] - читає файл infile, інтерпретує фрагменти, що містяться між дужками <* і *>, і записує результат в файл outfile.

Ця можливість особливо істотна при використанні програм на мовах програмування C (розширення .me), Fortran (розширення .mf) і TeX (розширення .mtex), для форматів яких Mathematica має кошти конвертації виразів (CForm, FortranForm і TexForm відповідно). Таким чином, є можливість експорту виразів системи Mathematica в програми, складені на цих мовах.

Пояснимо застосування функції-директиви Splice. Нехай є експортована програма на мові C, яка повинна розраховувати чисельне значення деякого інтеграла, і ми хочемо отримати формулу для цього інтеграла засобами системи Mathematica. Припустимо, вона представлена файлом demo.me. Його можна переглянути в такий спосіб:

!! demo.me

```
#include "mdefs.h"

double f (x)
```

```
double x;

{

double y;

y = <* Integrate [Sin [x] ^ 5, x] *>;

return (2 * y- 1);

}
```

Після виконання функції Splice ["demo.me"] програма буде записана в файл demo.c, в якому вираз в дужках <* ... *> замінено обчисленим значенням інтеграла (в формі CForm). Файл при цьому буде виглядати так:

!! demo.c

```
#include "mdefs.h" double f (x) double x;

{

double y;

y = 5 * Cos (x) / 8 + 5 * Cos (3 * x) / 48- Cos (5 * x) / 80;

return (2 * y- 1);

}
```

Запис визначень

З простих функцій, що забезпечують створення файлів з заданими визначеннями, треба відзначити також функцію Save:

```
Save [ "filename", symb1, symb2, ...]
```

Вона додає визначення символів symbi до файлу filename (можливі спрощені форми Save).

Наведемо приклад її використання:

```
f [x_] = Sin [x] + y

y + Sin [x]
```

y = a

a
Save ["demol", f]

!! demol

f [x_] = y + Sin [x]

y = a

Інші функції для роботи з файлами

В цілому кошти системи Mathematica забезпечують можливості роботи з різними файлами, властиві MS-DOS, без виходу з середовища системи. Які стосуються цій групі функції наведені у додатку. Для цих функцій характерно, що в момент виконання вони не дають видимого ефекту. До таких функцій відносяться функції копіювання директорій і файлів,
зміни їх імен, видалення і т. д. Вони добре відомі користувачам MS-DOS і можуть виконуватися з середовища Mathematica.

Розглядаючи великий список файлових і поточних операцій, можна мимоволі зробити висновок про їх надмірності. Але тут діє просте правило: не хочеш застосовувати ці функції - не застосовувати! Вони розраховані на користувача, всерйоз займається стикуванням систем Mathematica з іншими програмними системами.

Важливе місце займають функції, що дають інформацію про директорії, файлах і потоках. До них належать такі функції:

- Directory [] - повертає поточний робочий каталог;
- DirectoryStack [] - повертає вміст стека каталогів, яке представляє послідовність використовуваних в поточному сеансі каталогів;
- $ Display- повертає список файлів і каналів (pipes- канал або абстрактний файл), використовуваний функцією виведення $DisplayFunction за замовчуванням;
- FileByteCount ["filename"] - повертає кількість байтів у файлі;
- FileDate ["filename"] - повертає дату і час останньої модифікації файлу у вигляді списку;
- Filelnformation ["filename"] - повертає інформацію про фото;
- FileNames [] - наводить список всіх файлів в поточному робочому каталозі;
- FileNames ["form"] - перераховує всі файли в поточному робочому каталозі, чиї імена збігаються з шаблоном form;
- FileNames [{ "forml", "form2", ...}] - перераховує всі файли, чиї імена відповідають будь-якому з шаблонів formi;

• FileNames [forms, { "dirl", "dir2", ...}] - перераховує файли з іменами, відповідними шаблонами forms, в будь-якому із зазначених каталогів diri;
• FileType ["filename"] - повертає тип файлу: File, Directory або None (якщо зазначеного файлу не існує);
• $ HomeDirectory - дає ім'я «домашньої» директорії користувача;
• $ Output - дає список файлів і каналів, в яких спрямовується стандартний висновок системи Mathematica;
• ParentDirectory [] - повертає ім'я батьківського каталогу для поточного робочого каталогу;
• ParentDirectory ["dir"] - повертає ім'я батьківського каталогу для каталогу dir;
• $ Path - дає список каталогів для перегляду при спробі пошуку зовнішнього файлу;
• StreamPosition [stream] - повертає ціле число, яке вказує позицію поточної точки у відкритому потоці stream;
• Streams [] - повертає список всіх потоків, відкритих в даний момент;
• Streams ["name"] - перераховує тільки потоки з вказаним ім'ям name.

Наведені нижче приклади ілюструють використання більшості з цих досить простих функцій:

Directory []

C: \ PROGRAM FILES \ WOLFRAM RESEARCH \ MATHEMATICA \ 4.0

DirectoryStack []

{} **/ $ Display**

stdout

FileByteCount ["C: .val"]

46

FileDatef'C: .val "]

{1999 року, 8, 3, 16, 4, 44}

FileInformation ["C: .val"]

{File-> C: \ ma.val, FileType-> File, Date -> 3142685084, ByteCount -> 46}

Filenames []

{Examples, FILES, MATHEMATICA.EXE, MATH.EXE, MATHINSTALLER.EXE, MATHKERNEL.EXE}

FileType ["C: .val"]

File HomeDirectory []

c: \ $ 0utput

{OutputStream [stdout, 1]}

ParentDirectory []

3: \ m3 Streams []

{OutputStream [stdout, 1],

OutputStream [stderr, 2]}

Висловлене вище міркування про надмірності набору операцій цілком можна застосувати і для цих функцій.

Лекція № 16. Взаємодія системи Mathematica з програмою MatLab за допомогою пакету MatLink

Для виклику функції MATLAB безпосередньо з Mathematica і організації обміну даними і змінними між двома системами існує багатоплатформовий пакет під назвою MATLink. За допомогою нього легко організувати виклик функцій MATLAB прямо з Mathematica і передавати різні дані від однієї системи інший.

Установка. Спершу варто перейти на головну сторінку MATLink і виконати зазначені там інструкції. Найпростіший шлях – завантажити архів і розпакувати його в цю папку:

In [1]: =

```
SystemOpen @ FileNameJoin [{$ UserBaseDirectory,"Applications"}]
```

Далі слід виконати інструкції для конкретної операційної системи в розділі «Link with MATLAB» на головній сторінці.

Використання MATLink. MATLink можна завантажити, обчисливши осередок з кодом:

In [2]: =

```
Needs ["MATLink`"]
```

А потім запустити MATLAB командою:

In [3]: =

```
OpenMATLAB []
```

Таким чином буде запущений новий процес в MATLAB, з яким Mathematica зможе взаємодіяти.

Для використання довільних команд MATLAB'а слід використовувати MEvaluate. Дані будуть передані у вигляді рядка.

In [4]: =

```
MEvaluate ["Magic (4)"]
```

Out [4] =

```
(* ==>
```

```
ans =
```

```
 16 2 3 13

 5 11 10 8

 9 7 6 12

 4 14 15 1

*)
```

Щоб передати дані в MATLAB, слід використовувати MSet:
In [5]: =

```
MSet [ "x", Range [10]]; MEvaluate [ "x"]
```

Out [5] =

```
(* ==>

x =

 1 2 3 4 5 6 7 8 9 10

*)
```

Щоб повернути дані назад, слід використовувати MGet:
In [6]: =

```
MGet ["X"]
```

Out [6] =

```
(* ==> {1., 2., 3., 4., 5., 6., 7., 8., 9., 10.} *)
```

Підтримується велика кількість типів даних, включаючи зріджені масиви, структури та осередки.

Функції MATLAB'a можуть викликатися безпосередньо з Mathematica за допомогою функції MFunction:

In [7]: =

```
eig = MFunction ["Eig"];
```

```
eig [{{1, 2}, {3, 1}}]
```

Out [7] =

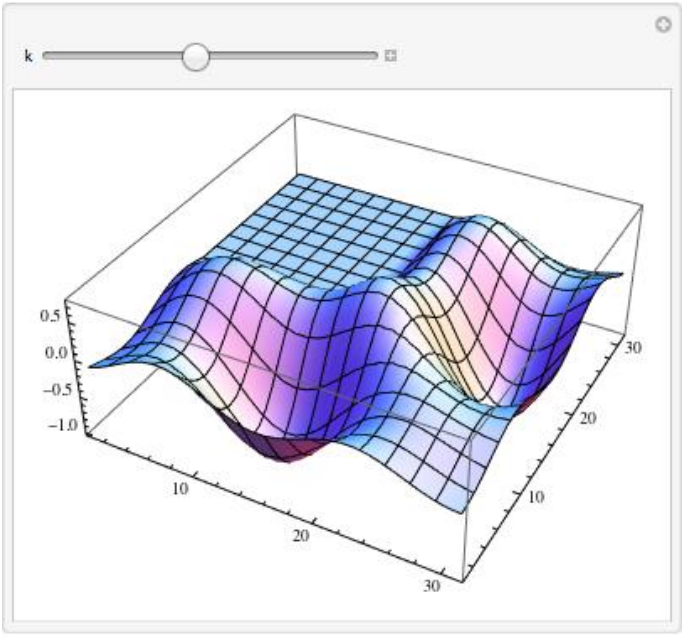
```
(* ==> {{3.44949}, {-1.44949}} *)
```

Побудуємо в Mathematica поверхню логотипу MATLAB і додамо маніпулятор, який буде регулювати період коливань:

In [8]: =

```
Manipulate [
ListPlot3D @ MFunction ["Membrane"] [k],
{K, 1, 12, 1}
]
```

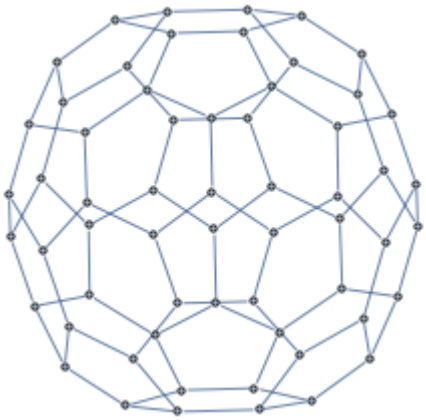
Out [8] =



Фуллеренова структура (bucky ball) прямо з MATLAB:
In [9]: =

```
AdjacencyGraph @ Round @ MFunction ["Bucky"] []
```

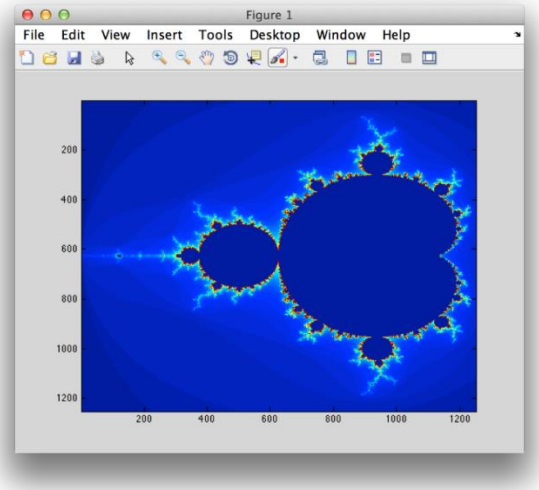
Out [9] =



Відобразити дані з Mathematica в окремому масштабованому вікні для малюнків, які застосовуються в MATLAB також легко:
In [9]: =

```
mlf = LibraryFunctionLoad [ "demo_numerical", "mandelbrot", {Complex}, Int
eger];

mandel = Table [mlf [x + I y], {y, -1.25, 1.25, .002}, {x, -2., 0.5, .002}];

MFunction ["Image", "Output" -> False] [mandel]
```

Out [9] =



Зазначені нижче приклади ілюструють вирішення реальних завдань з використанням MATLink, дозволяючи використовувати кращі якості систем MATLAB і Mathematica.

Швидкі тріангуляції Делоне. У складі Mathematica міститься функція DelaunayTriangulation всередині пакету ComputationalGeometry *(В 10-й версії цей пакет став вбудованим в ядро і тепер ця функції носить ім'яDelaunayMesh, Вона оптимізована і тепер її продуктивність не поступається MATLAB - прим. ред.)*, Однак працює вона дуже повільно (хоча у неї є і свої сильні сторони, такі як використання точної арифметики і робота з колінеарними точками). Це, в свою чергу, призводить до того, що ListDensityPlot працює дуже неефективно (що стає помітно при побудові декількох тисяч точок і більше). З використанням MATLink, ми можемо задіяти функцію Делоне з MATLAB для обчислення тріангуляції Делоне деякого набору точок наступним чином:

In [10]: =

```
delaunay = Composition [Round, MFunction ["Delaunay"]];
```

Оскільки функція системи Mathematica повертає список суміжних вершин, нам необхідно провести пост-обробку результату для того, щоб порівняти з результатом з MATLAB'a:

In [11]: =

```
Needs ["ComputationalGeometry`"];

delaunayMma [points_]: =

    Module [{tr, triples},

        tr = DelaunayTriangulation [points];

        triples = Flatten [

            Function [{v, list},

                Switch [Length [list],
```

```
                                    (* Account for nodes with connectivity 2 or less *
)
                                    1, {},
                                    2, {Flatten [{v, list}]}, _, {v, ##} & @@@ Partiti
on [list, 2, 1, {1, 1}]]
                                ] @@@ tr, 1];
                            Cases [GatherBy [triples, Sort], a_ /; Length [a] ==3 :> A[[1]]]
    ]
```

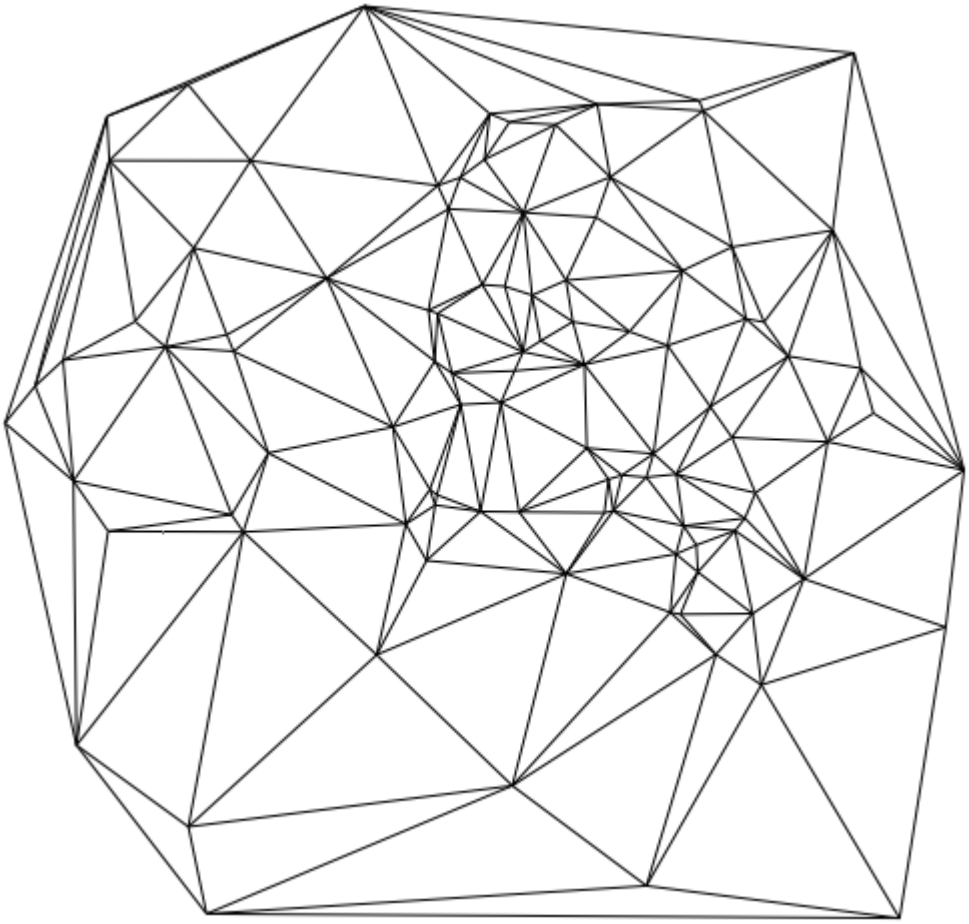
Випадковий набір точок зазвичай має унікальну тріангуляцію Делоне, так що нам необхідно буде перевірити, що системи видають один і той же результат.
In [12]: =

```
pts = RandomVariate [NormalDistribution [], {100, 2}];
Sort [Sort / @ Delaunay [pts]] === Sort [Sort /@ DelaunayMma [pts]]
```

І побудуємо тріангуляцію за допомогою:
In [13]: =

```
trianglesToLines [t_]: =
 Union @ Flatten [{{# 1, # 2}, {# 2, # 3}, {# 1, # 3}} & @@
        Transpose [Sort / @ t], {{1, 3}, {2}}];
Graphics @ GraphicsComplex [pts, Line @ trianglesToLines @ delaunay [pts]]
```

Out [13]: =



Однак крім того, що delaunay працює значно швидше DelaunayTriangulation (особливо для великих наборів даних), вона також швидше і triangulator, який використовується всередині ListDensityPlot. Отже, ми можемо використовувати delaunay з MATLAB'a для розробки своєї версії listDensityPlot, яка працює швидше, ніж вбудована функція, а також може обробляти великі набори даних в такий спосіб:

In [14]: =

```
Options[ListDensityPlot] = Options[Graphics] ~Join~ {ColorFunction -> Autom
atic, MeshStyle -> None, Frame -> True};

listDensityPlot [data_? MatrixQ, opt: OptionsPattern []]: =
    Module [{in, out, Tri, colfun},
        tri = delaunay [data [[All, 1;;2]]];
```

```
        colfun = OptionValue [ColorFunction];
        If[Not@MatchQ [colfun, _Symbol | _Function],Check[Colfun = C
olorData [colfun], colfun = Automatic]];
        If[Colfun === Automatic, colfun = ColorData [ "LakeColors"]];
        Graphics [
            GraphicsComplex [data [[All, 1;;2]],
                GraphicsGroup [{EdgeForm [OptionValue [Mesh
Style]], Polygon[Tri]}],
                VertexColors -> colfun / @ Rescale [data [[All, 3]
]]
            ],
            Sequence @@ FilterRules [{opt}, Options[Graphics]], Me
thod -> { "GridLinesInFront" -> True}
        ]
    ]
```

Порівняємо отриману функцію з вбудованою, використовуючи при цьому масив з 30 000 точок:

In [15]: =

```
pts = RandomReal [{-1, 1}, {30000, 2}];
```

```
values = Sin[3 Sqrt[# 1 ^ 2 + # 2 ^ 2]] & @@@ pts;
```

In [16]: =

```
listDensityPlot [ArrayFlatten [{{pts, List / @ Values}}], Frame -> True] // Absol
```

```
uteTiming
```

Out [16] =

```
{0.409001, --Graphics--}
```

In [17]: =

```
ListDensityPlot [ArrayFlatten [{{pts, List / @ Values}}]] // AbsoluteTiming
```

Out [17] =

```
{12.416587, --Graphics--}
```

Різниця в швидкості виконання вийшла досить значною (~ 30 раз). Для роботи з сотнями тисяч точок ListDensityPlot виявляється практично непридатною, в той час як listDensityPlottребует лише кілька секунд.

Також важливо враховувати, що для замірів швидкості роботи MATLink необхідно використовувати функцію AbsoluteTiming, За допомогою якої обчислюється все витрачений час, в той час як Timing вимірює тільки той час, коли CPU використовувався ядром Mathematica, що не вимірюючи час, який витратив MATLAB.

Фільтрація аудіо за допомогою інструментів для обробки сигналів (signal processing toolbox). Як відомо, функціонал по обробці сигналів був відсутній в Mathematica до дев'ятої версії, і як і раніше поступається інструментів MATLAB з точки зору функціональності та простоти використання. Припустимо, у нас 8 версія Mathematica, нові функції відсутні і

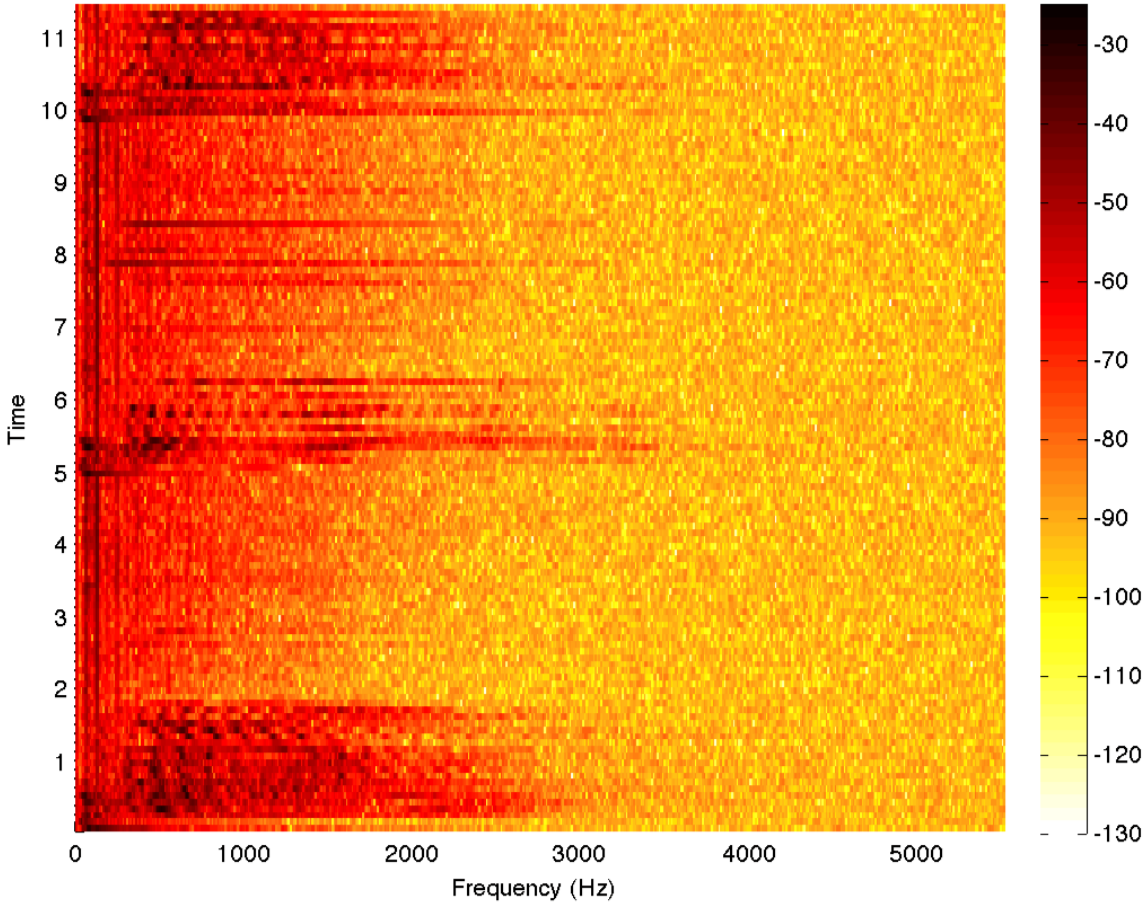
ми хочемо провести частотний аналіз деякого аудіофайлу і реалізувати фільтрацію. Ось як це можна буде зробити:
In [18]: =

```
{data, Fs} = {# [[1, 1, 1]], # [[1, 2]]} & @ExampleData[{ "Sound","Apollo13Pr
oblem"}];
spectrogram = MFunction["Spectrogram", "Output" -> False]; (*Use MATLAB'
s spectrogram *)
spectrogram[data, 1000, 0, 1024, fs]
```

Очевидно, що частоти в основному припадають на діапазон нижче 2,5 кГц, так що ми можемо розробити в MATLAB фільтр нижніх частот, а також зробити допоміжну функцію, яка буде повертати відфільтровані дані:
In [19]: =

```
MSet ["Fs", Fs];
MEvaluate ["
    [Z, p, k] = butter (6, 2.5e3 / fs, 'low');
    [Sos, g] = zp2sos (z, p, k);
    Hd = dfilt.df2tsos (sos, g);
"]
filter = MFunction ["Myfilt", "@ (X) filter (Hd, x)"];
```

Out [19] =



Тепер ми все підготували для того, щоб застосовувати фільтруючу функцію до даних безпосередньо з Mathematica. Цей приклад показує, як ми можемо заповнити прогалини у функціональності. Таким чином, ми можемо заощадити велику кількість часу на проектування фільтра в Mathematica (а це не найпростіше завдання) і багато годин на його налагодження. Код для фільтра Баттерворта можна взяти звідки завгодно - з файлообмінника або Stack Overflow, з фрагментів раніше написаного коду, або, як в даному випадку, з прикладу в документації. Невеликі зміни в параметрах відповідно до своїх потреб, і ми тепер можемо працювати з цим матеріалом в Mathematica.

Обробимо деякі дані за допомогою нашого фільтра і побудуємо спектрограму:

In [19]: =

```
filteredData = filter @data;
```

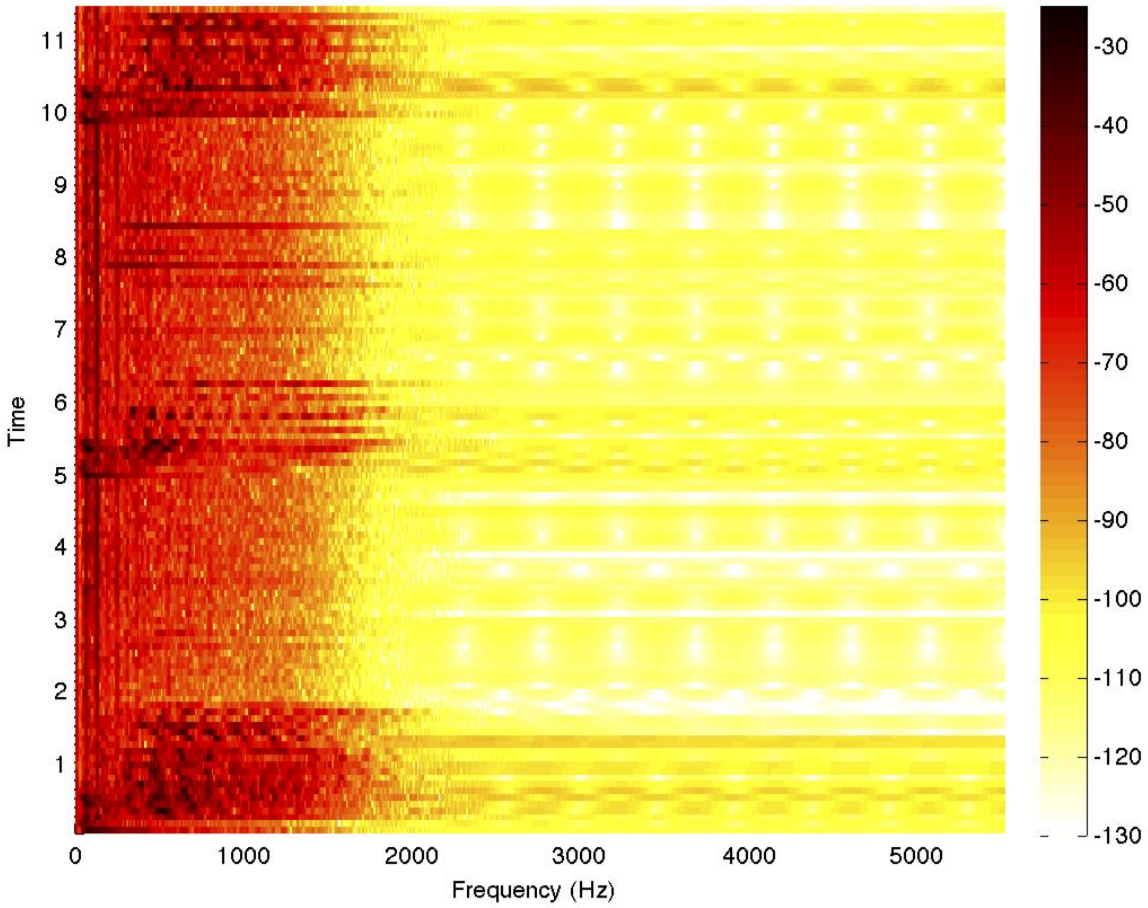
```
spectrogram [filteredData, 1000, 0, 1024, Fs]
```

Можемо програти обидва аудіофайлу - відфільтрований і вихідний -і порівняти різницю в їх звучанні:

In [20]: =

```
ListPlay[data, SampleRate -> fs]
```

```
ListPlay[FilteredData, SampleRate -> fs]
```



Література

1. В.З. Аладьев, Д.С. Гринь "Расширение функциональной среды системы Mathematica" / Монография / Херсон: Олди-Плюс, 2012, 552 с.
2. В.З. Аладьев, В.К. Бойко, Е.А. Ровба "Программирование в пакетах Maple и Mathematica: Сравнительный аспект" / Монография / Гродно: Гродненский Госуниверситет, 2011, 517 с.
3. Mathematica 5/6/7. Полное руководство / Дьяконов В. П. - ДМК Пресс – 2009, 624 стр.
4. Mathematica. Практический курс с примерами решения прикладных задач / Васильев А. Н. - КОРОНА-Век - 2008.
5. Mathematica / А. Половко - БХВ-Петербург – 2007 - 368 стр.
6. Введение в систему символьных, графических и численных вычислений "Математика-5" / Е. Воробьев - "Диалог-МИФИ" - 2005 - 368 стр.
7. Differential Equations with Mathematica, Third Edition / Brian R. Hunt, Ronald L. Lipsman, John E. Osborn, Donald A. Outing, Jonathan Rosenberg - 2009 John Wiley & Sons, 271 pp.
8. A Physicist's Guide to Mathematica, Second Edition / Patrick T. Tam – 2008 Academic Pres, 728 pp.
9. Computer Solutions in Physics: With Applications in Astrophysics, Biophysics, Differential Equations, and Engineering / Steve VanWyk - World Scientific 2008 - 282 pp.
10. Mathematica by Example, Fourth Edition / Martha L. Abell, James P. Braselton Publisher: Academic Press 2008 - 576 pp.
11. Mathematica DeMYSTiFied / Jim Hoste - McGraw-Hill Professional 2008 - 320 pp.
12. Mathematica Navigator: Mathematics, Statistics and Graphics, Third Edition / Heikki Ruskeepaa Academic Press 2009 - 1136 pp.
13.