Міністерство освіти і науки України
Запорізький національний університет

Є. В. Панасенко

МАТЕМАТИЧНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності Е7 Математика
освітньо-професійної програми «Математика»

Затверджено
вченою радою ЗНУ
Протокол №
від __.__.2025

Запоріжжя
2025

УДК 004.43:51(076.5)
П641

Панасенко Є. В. Математичне програмне забезпечення : методичні рекомендації до лабораторних занять для здобувачів ступеня вищої освіти бакалавра спеціальності Е7 Математика освітньо-професійної програми «Математика». Запоріжжя : Запорізький національний університет, 2025. 80 с.

У методичних рекомендаціях в систематизованому вигляді подано матеріал для виконання лабораторних робіт з дисципліни «Математичне програмне забезпечення». У виданні представлено теоретичний матеріал з кожної теми дисципліни, хід виконання та рекомендації щодо виконання лабораторних робіт, які пропонуються при вивченні даного курсу. Кожна лабораторна робота супроводжується необхідними прикладами та поясненнями до них, певними практичними завданнями, виконання яких поглиблює сприйняття матеріалу та контрольними запитаннями.

Методичні рекомендації призначено для здобувачів ступеня вищої освіти бакалавра спеціальності Е7 Математика освітньо-професійної програми «Математика».

Рецензент
А.О. Лісняк, кандидат фізико-математичних наук, доцент, завідувач кафедри програмної інженерії

Відповідальний за випуск
С.М. Гребенюк, доктор технічних наук, професор, завідувач кафедри фундаментальної та прикладної математики

ЗМІСТ

ВСТУП

Математичне програмне забезпечення – це спеціалізовані комп'ютерні програми або системи, призначені для розв'язання математичних задач. Вони дозволяють проводити складні обчислення, моделювання та симуляції, аналіз та візуалізація даних, побудову графіків, символьні перетворення, статистичний аналіз та інші математичні операції.

У сучасному світі інформаційних технологій математичне програмне забезпечення відіграє ключову роль у науці, техніці, економіці, освіті та багатьох інших сферах діяльності. З розвитком комп'ютерних систем з'явилися потужні інструменти, які значно розширили можливості розв'язання складних математичних задач, аналізу великих обсягів даних, побудови математичних моделей реальних процесів і їхньої візуалізації.

Дисципліна «Математичне програмне забезпечення» спрямована на формування у здобувачів освіти компетентностей щодо використання сучасних програмних засобів для обчислень, символьної математики, статистичного аналізу та моделювання. Вивчення цієї дисципліни дозволяє не лише опанувати практичні навички роботи з такими системами, як Maxima, SageMath та Scilab, але й розвинути вміння застосовувати їх у професійній та науково-дослідній діяльності.

Засвоєння курсу сприятиме підвищенню ефективності математичних розрахунків, оптимізації процесів прийняття рішень, автоматизації інженерних задач, а також підготовці якісної наукової та технічної документації. Таким чином, вивчення дисципліни є важливою складовою професійної підготовки фахівців, які працюватимуть у галузях, де необхідно поєднувати математичні методи та інформаційні технології.

Метою вивчення навчальної дисципліни «Математичне програмне забезпечення» є засвоєння систематичних знань, умінь і навичок щодо використання спеціалізованого програмного забезпечення для розв'язання прикладних і математичних задач, моделювання процесів, аналізу даних, візуалізації результатів та автоматизації обчислювальних процедур у професійній діяльності.

Основні функції математичного програмного забезпечення:

- числові обчислення (наприклад, розв'язання рівнянь, інтегрування, диференціювання);
- символьна математика (робота з формулами, наприклад у системах типу Maxima, SageMath або Scilab);
- статистичний аналіз (обробка даних, статистичні тести);
- побудова графіків і візуалізація;
- моделювання процесів і симуляції.

Отже, для студентів системи комп'ютерної математики – це зручний засіб розв'язування всіляких задач, пов'язаних із символьними перетвореннями (математичний аналіз, вища математика, лінійна алгебра і аналітична геометрія тощо), а також засіб розв’язування задач моделювання статичних (описуваних алгебраїчними рівняннями) і динамічних (описуваних диференціальними рівняннями) систем.

Основними **завданнями** вивчення дисципліни «Математичне програмне забезпечення» є:

- ознайомлення з видами математичного програмного забезпечення, його класифікацією та можливостями застосування в різних галузях науки, техніки й освіти;
- формування навичок роботи з популярними математичними програмними системами;
- опанування інструментів візуалізації математичних даних та результатів обчислень, включаючи побудову графіків, діаграм, анімацій;
- набути навички із застосування математичного програмного забезпечення до розв’язання конкретних прикладних науково-технічних задач математики;
- навчитися обирати потрібне програмне забезпечення для розв’язання конкретних математичних задач.

У результаті вивчення навчальної дисципліни студент повинен набути таких програмних компетентностей:

- здатність застосовувати знання у практичних ситуаціях;
- здатність учитися і оволодівати сучасними знаннями;
- здатність до пошуку, обробки та аналізу інформації з різних джерел;
- здатність застосовувати спеціалізовані мови програмування та пакети прикладних програм;
- здатність використовувати обчислювальні інструменти для чисельних і символьних розрахунків.

У результаті вивчення навчальної дисципліни студент повинен набути таких програмних результатів навчання:

- розуміти фундаментальну математику на рівні, необхідному для досягнення інших вимог освітньої програми;
- мати навички використання спеціалізованих програмних засобів комп’ютерної та прикладної математики і використовувати інтернет-ресурси;
- уміти працювати зі спеціальною літературою іноземною мовою;
- розв’язувати задачі придатними математичними методами, перевіряти умови виконання математичних тверджень, коректно переносити умови та твердження на нові класи об’єктів, знаходити й аналізувати відповідності між поставленою задачею й відомими моделями;
- відшуковувати потрібну науково-технічну інформацію у науковій літературі, базах даних та інших джерелах інформації;
- розв’язувати основні математичні задачі аналізу даних; застосовувати базові загальні математичні моделі для специфічних ситуацій, мати

навички управління інформацією, і застосування комп'ютерних засобів статистичного аналізу даних.

Курс «Математичне програмне забезпечення» є логічним продовженням курсів: «Дискретна математика» і «Лінійна алгебра. Набуті знання при вивченні курсу «Математичне програмне забезпечення» необхідні для подальшого проходження «Виробничої практики».

1. Лабораторна робота. Символьні перетворення у системі комп'ютерної алгебри Maxima

Мета: засвоїти можливості роботи системи комп'ютерної алгебри Maxima до виконання символьних і числових обчислень.

Теоретичні відомості та методичні рекомендації

Maxima – класичний приклад математичного програмного забезпечення, яке використовується для символьної математики, аналітичних розрахунків і навчальних цілей.

Maxima – це система комп'ютерної алгебри, яка призначена для виконання символьних і числових обчислень. Вона дозволяє:

– виконувати аналітичні обчислення (спрощення виразів, диференціювання, інтегрування, розклад у ряд);
– розв'язувати рівняння і системи рівнянь;
– виконувати дії з матрицями та векторами;
– будувати графіки функцій;
– здійснювати числові обчислення з високою точністю.

Після запуску Maxima з'являється вікно програми, де у верхній частині вікна інтерфейсу вказано, яка завантажена версія (див. рис. 1.1).

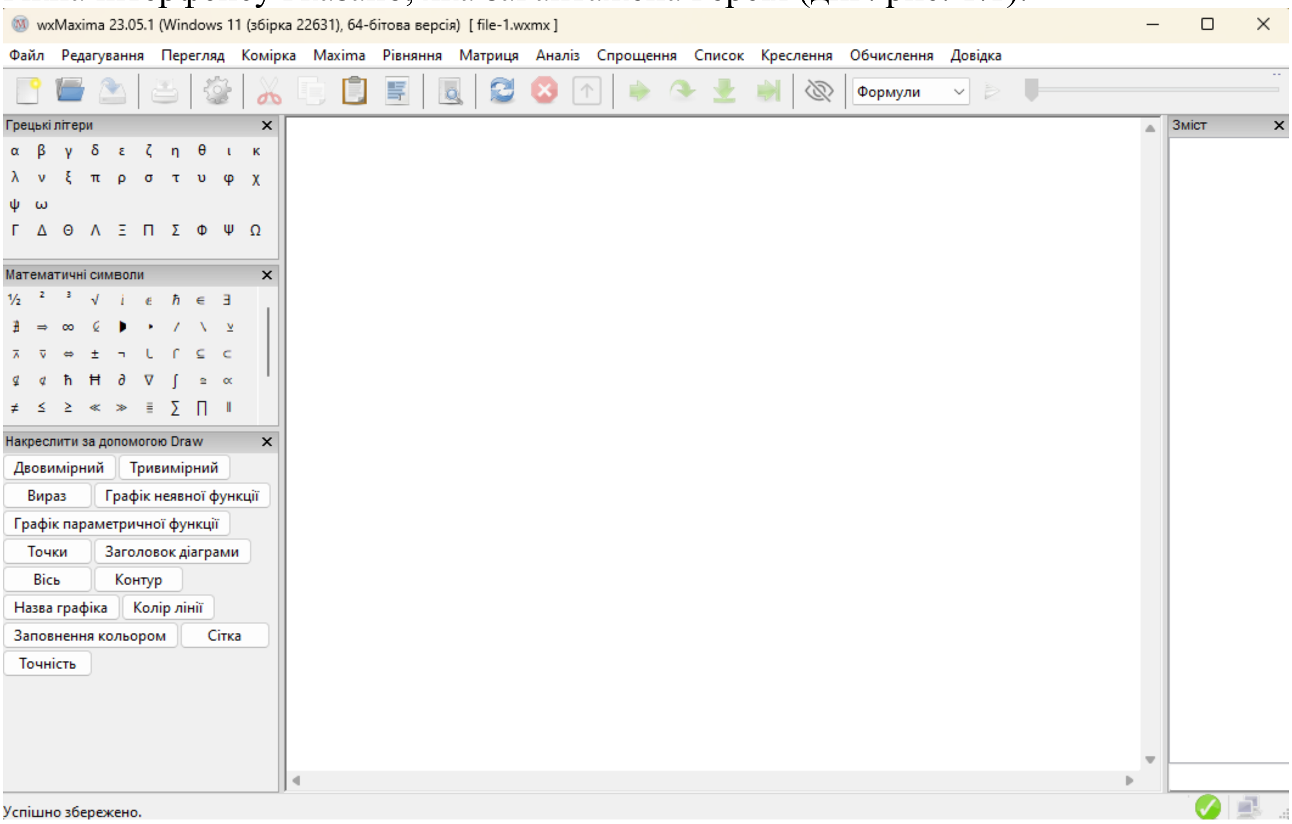

Рисунок 1.1 – Графічний інтерфейс wxMaxima

Особливості:

1) Maxima є безкоштовним програмним забезпеченням з відкритим кодом;

2) Maxima базується на системі Macsyma, одній із перших систем символьної математики розробленої Massachusetts Institute of Technology (MIT);
3) Maxima має зручну графічну оболонку wxMaxima, яка спрощує роботу завдяки зручному інтерфейсу, а також є можливість працювати в режимі командного рядка;
4) Maxima дає можливість вирішувати широкий клас задач.

Після введення кожній команді надається порядковий номер (%i1), (%i2) і так далі. Тут літера «і» – це скорочення від англійського input. Результати обчислень мають відповідно порядковий номер (%o1), (%o2) і так далі (див. рис. 1.2). Тут літера «о» – це скорочення від англійського output.

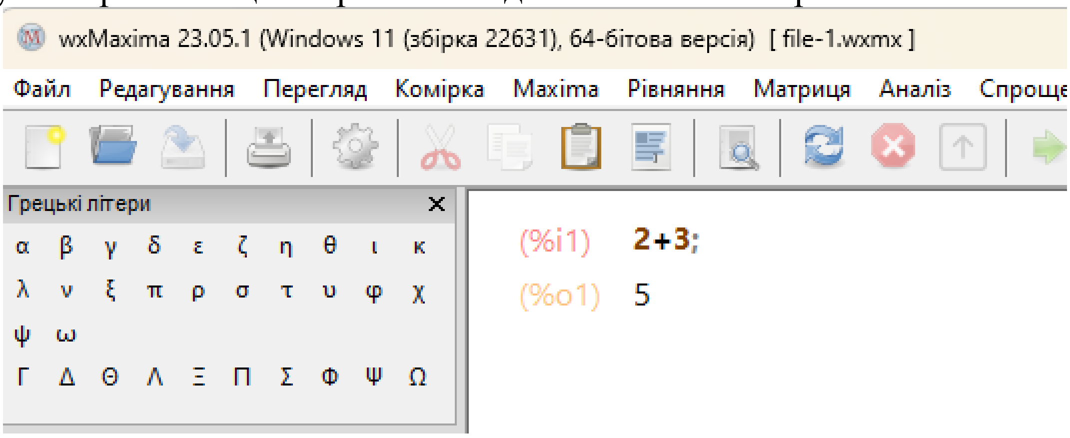

Рисунок 1.2 – Результати виводу команди

Основні принципи роботи.

– введення команд здійснюється через інтерфейс wxMaxima;
– кожна команда завершується **крапкою з комою** «;»;
– у Maxima знак `$` у кінці рядка означає, що треба виконати команду, але не виводити результат на екран;
– можна писати коментарі в оточенні `/*    */`;
– чисельник і знаменник звичайних дробів розділяється за допомогою символу / (прямий слеш);
– для отримання конкретного числового значення використовується оператор `numer` (див. рис. 1.3);
– виконання команди підтверджується натисканням клавіш Shift і Enter для її обробки та виведення результату.

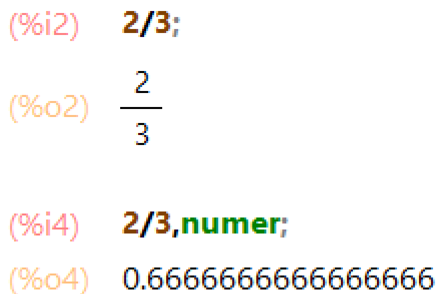

Рисунок 1.3 – Використання оператора `numer`

Однією із дуже небагатьох речей, які є нестандартними у wxMaxima є те, що дані для Maxima у програмі впорядковано за комірками, які обчислюються (тобто надсилаються до Maxima) лише на запит користувача. При обчисленні комірки усі команди у ній, і лише у ній, обчислюються пакетно. Підхід wxMaxima до надсилання команд на виконання може спочатку видатися незвичним. Втім, він значно спрощує роботу із великими документами (у яких користувач, звичайно ж, не є хоче, щоб програма автоматично вмикала повне переобчислення усього документа у відповідь на будь-яку незначну зміну). Крім того, такий підхід є дуже зручним для усування усіх недоліків.

У Maxima є кілька вбудованих математичних констант, які можна використовувати у виразах і обчисленнях. Усі спеціальні символи починаються з `%`. У таблиці 1.1 наведено основні сталі (константи), що підтримуються системою:

Таблиця 1.1 – Основні числові та математичні константи в Maxima

Назва	Позначення	Приклад використання
Число π	`%pi`	`float(%pi);`
Число e	`%e`	`exp(1);, float(%e);`
Уявна одиниця	`%i`	`%i^2;`
Плюс нескінченність	`inf`	`1/inf;`
Мінус нескінченність	`minf`	`1/minf;`
Золоте число (≈ 1.618)	`%phi`	`float(%phi);`
Стала Ейлера–Маскероні	`%gamma`	`float(%gamma);`

У Maxima підтримуються всі основні арифметичні операції, як і в більшості мов програмування та математичних систем. У таблиці 1.2 наведено основні базові операції.

Таблиця 1.2 – Базові арифметичні операції в Maxima

Операція	Позначення	Приклад	Результат
Додавання	`+`	`2 + 3;`	5
Віднімання	`-`	`7 - 4;`	3
Множення	`*`	`6 * 5;`	30
Ділення	`/`	`8 / 2;`	4
Піднесення до степеня	`^` або `**`	`2^3;` або `2**3;`	8
Остача від ділення	`mod`	`mod(7, 3);`	1
Цілочисельне ділення	`quotient`	`quotient(7, 3);`	2
Унарний мінус	`-`	`-5 + 2;`	-3
Факторіал	`!`	`5!;`	120

У Maxima є велика кількість вбудованих математичних функцій, які поділяються на кілька основних груп (див. таблиці 1.3–1.7).

Таблиця 1.3 – Арифметичні функції

Назва функції	Приклад	Опис
`abs(x)`	`abs(-5);`	Модуль (абсолютне значення)
`floor(x)`	`floor(3.7);`	Округлення вниз
`ceiling(x)`	`ceiling(3.1);`	Округлення вгору
`round(x)`	`round(2.6);`	Округлення до найближчого цілого
`mod(x,y)`	`mod(7,3);`	Остача від ділення
`min(a,b,...)`	`min(3,7,2);`	Мінімальне значення
`max(a,b,...)`	`max(3,7,2);`	Максимальне значення

Таблиця 1.4 – Тригонометричні функції

Назва функції	Приклад	Опис
`sin(x)`	`sin(%pi/2);`	Синус
`cos(x)`	`cos(0);`	Косинус
`tan(x)`	`tan(%pi/4);`	Тангенс
`sec(x)`	`sec(0);`	Секанс
`csc(x)`	`csc(%pi/2);`	Косеканс
`cot(x)`	`cot(%pi/4);`	Котангенс
`asin(x)`	`asin(1);`	Арксинус
`acos(x)`	`acos(0);`	Арккосинус
`atan(x)`	`atan(1);`	Арктангенс

Таблиця 1.5 – Експонента та логарифмічні функції

Назва функції	Приклад	Опис
`exp(x)`	`exp(1);`	Експонента e^x
`log(x)`	`log(10);`	Натуральний логарифм
`log10(x)`	`log10(100);`	Десятковий логарифм
`sqrt(x)`	`sqrt(9);`	Квадратний корінь
`nthroot(x,n)`	`nthroot(8,3);`	Корінь n-го степеня

Таблиця 1.6 – Статистичні функції (для списків)

Назва функції	Приклад	Опис
`mean(...)`	`mean([2, 4, 6]);`	Середнє значення
`median(...)`	`median([1, 5, 7, 9]);`	Медіана
`variance(...)`	`variance([2,3,4]);`	Дисперсія
`std(...)`	`std([2,3,4]);`	Стандартне відхилення

Таблиця 1.7 – Інші корисні функції

Назва функції	Приклад	Опис
`factorial(n)`	`factorial(5);`	Факторіал $n!$
`gamma(x)`	`gamma(5);`	Гамма-функція
`sign(x)`	`sign(-7);`	Знак числа $(-1,0,1)$
`unit_step(x)`	`unit_step(-2);`	Одинична ступенева функція

У Maxima є багато команд для перетворення алгебраїчних виразів, тобто для спрощення, розкладання, перетворення виразу в канонічну або зручну форму.

Команда `expand(expr)` призначена для розкриття дужок, розкладання добутків (див. рис. 1.4).

```
(%i18) expand((x + 1)^3);
```

$$(\%o18)\quad x^3+3x^2+3x+1$$

Рисунок 1.4 – Використання команди `expand(expr)`

Команда `factor(expr)` призначена для розкладання на множники (див. рис. 1.5).

```
(%i19) factor(x^2 - 1);
```

$$(\%o19)\quad (x-1)(x+1)$$

Рисунок 1.5 – Використання команди `factor(expr)`

Команда `ratsimp(expr)` призначена для спрощення дробів, зведення дробів, скорочення дробів (див. рис. 1.6).

```
(%i20) ratsimp((x^2 - 1)/(x - 1));
```

$$(\%o20)\quad x+1$$

Рисунок 1.6 – Використання команди `ratsimp(expr)`

Команда `fullratsimp(expr)` призначена для спрощення виразів із коренями та степенямив (див. рис. 1.7).

```
(%i21) fullratsimp((x^2 - y^2)/(x - y));
```

$$(\%o21)\quad y+x$$

Рисунок 1.7 – Використання команди `fullratsimp(expr)`

Команда `radcan(expr)` призначена для загального скорочення складних виразів (див. рис. 1.8).

```
(%i22) radcan(sqrt(x^2));
```

$$(\%o22)\quad |x|$$

Рисунок 1.8 – Використання команди `radcan(expr)`

За потреби можна використовувати і інші команди:

- команда `rat(expr)` призначена для перетворення раціональних виразів до так званої канонічної форми;
- команда `logexpand(expr)` призначена для розкладання логарифмів, застосування логарифмічних тотожностей;

– команда `logcontract(expr)` призначена для згортання логарифмів до однієї функції;
– команда `radcan(expr)` призначена для перетворення виразів, які містять логарифмічні, показникові та степеневі функції;
– команда `trig(expr)` призначена для загального спрощення тригонометричних виразів;
– команда `trigsimp(expr)` призначена для спрощення тригонометричних виразів;
– команда `trigexpand(expr)` призначена для розкладання тригонометричних виразів за формулами;
– команда `trigreduce(expr)` призначена для перетворення добутків тригонометричних функцій на суми;
– команда `ratsubst(expr)` призначена для підстановки значення у вираз, з подальшим спрощенням;
– команда `subst(expr)` призначена для заміни підвиразів у формулі;
– команда `collectterms(expr)` призначена для збирання подібних членів за змінною;
– команда `partfrac(expr, x)` призначена для перетворення дробово-раціональної функції на суму елементарних дробів.

Наприклад, (див. рис. 1.9):

```
(%i35) expr: (x^2 - 1)/(x - 1);
       ratsimp(expr);
```

$$\text{expr} \quad \frac{x^2-1}{x-1}$$

```
(%o35) x + 1
```

Рисунок 1.9 – Приклад використання функції перетворення

Ви можете комбінувати кілька команд, наприклад (див. рис. 1.10):

```
(%i36) factor(ratsimp((x^3 - x)/(x^2 - 1)));
(%o36) x
```

Рисунок 1.10 – Приклад комбінування кількох команд

Отже, Maxima – потужний і безкоштовний інструмент для навчання, наукових розрахунків і моделювання. Його зручно використовувати для аналітичних обчислень та символьної математики.

Завдання до лабораторної роботи

1. Обчислити у СКА Maxima:

1) $\frac{(7-6,35):6,5+9,9}{\left(1,2:36+1,2:0,25-1\frac{5}{16}\right):\frac{169}{24}}$.

2) $\left(\left(\frac{7}{9}-\frac{47}{72}\right):1{,}25+\frac{7}{40}\right):(0{,}358-0{,}108)\cdot 1{,}6-\frac{19}{25}$.

3) $\frac{\left(0{,}5:1{,}25+\frac{7}{5}:1\frac{4}{7}-\frac{3}{11}\right)\cdot 3}{\left(1{,}5+\frac{1}{4}\right):18\frac{1}{3}}$.

4) $\left(\frac{(2{,}7-0{,}8)\cdot 2\frac{1}{3}}{(5{,}2-1{,}4):\frac{3}{70}}+0{,}125\right):2\frac{1}{2}+0{,}43$.

5) $\frac{2\frac{3}{4}:1{,}1+3\frac{1}{3}}{2{,}5+0{,}4\cdot 3\frac{1}{3}}:\frac{5}{7}-\frac{\left(2\frac{1}{6}+4{,}5\right)\cdot 0{,}375}{2{,}75-1\frac{1}{2}}$.

6) $\frac{\left(13{,}75+9\frac{1}{6}\right)\cdot 1{,}2}{\left(10{,}3-8\frac{1}{2}\right)\cdot\frac{5}{9}}+\frac{\left(6{,}8-3\frac{3}{5}\right)\cdot 5\frac{5}{6}}{\left(3\frac{2}{3}-3\frac{1}{6}\right)\cdot 56}-27\frac{1}{6}$.

7) $\frac{\left(\frac{1}{6}+0{,}1+\frac{1}{15}\right):\left(\frac{1}{6}+0{,}1-\frac{1}{15}\right)\cdot 2{,}52}{\left(0{,}5-\frac{1}{3}+0{,}25-\frac{1}{5}\right):\left(0{,}25-\frac{1}{6}\right)\cdot\frac{7}{13}}$.

8) $\left(\frac{3\frac{1}{3}+2{,}5}{2{,}5-3\frac{1}{3}}\cdot\frac{4{,}6-2\frac{1}{3}}{4{,}6+2\frac{1}{3}}\cdot 5{,}2\right):\left(\frac{0{,}05}{\frac{1}{7}-0{,}125}+5{,}7\right)$.

9) $\frac{0{,}4+8\left(5-0{,}8\cdot\frac{5}{8}\right)-5:2\frac{1}{2}}{\left(1\frac{7}{8}\cdot 8-\left(8{,}9-2{,}6:\frac{2}{3}\right)\right)\cdot 34\frac{2}{5}}\cdot 90$.

10) $\frac{\left(5\frac{4}{45}-4\frac{1}{6}\right):5\frac{8}{15}}{\left(4\frac{2}{3}+0{,}75\right)\cdot 3\frac{9}{13}}\cdot 34\frac{2}{7}+\frac{0{,}3:0{,}01}{70}+\frac{2}{7}$.

2. Створити випадковий набір даних з 50 чисел за допомогою функції `makelist(random(100), i, 1, 50)`. Обчислити середнє значення вибірки, медіану, дисперсію, стандартне відхилення. Зробити висновки.

3. Розкласти дробово-раціональну функцію на суму елементарних дробів у СКА Maxima:

1) $\frac{x^2-5}{(x^2-4x+5)(x^2+9)}$.

2) $\frac{x+1}{x(x^4+6x^2+8)}$.

3) $\frac{1}{x^2(x^2+4)^2}$.

4) $\frac{3x^3+2x^2+1}{x(x+1)^2(x^2+4)}$.

5) $\frac{x^4-1}{(x^2+9)(x^3+x^2)}$.

6) $\frac{x^3+2x^2+3x+4}{x^4+x^3+2x^2}$.

7) $\frac{3x^2+4x-1}{(x^2+4x+29)^2}$.

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

9) $\frac{x^4-1}{(x^2+16)(x^3+x^2)}$.

10) $\frac{5}{(x+4)(x^2+4x+20)}$.

4. Спростити вираз у СКА Maxima:

1) $\frac{\sqrt{x}+1}{x\sqrt{x}+x+\sqrt{x}}:\frac{1}{x^2-\sqrt{x}}$.

2) $\left(\left(\sqrt[4]{p}-\sqrt[4]{q}\right)^{-2}+\left(\sqrt[4]{p}+\sqrt[4]{q}\right)^{-2}\right):\frac{\sqrt{p}+\sqrt{q}}{p-q}$.

3) $\frac{2x^{-\frac{1}{3}}}{x^{\frac{2}{3}}-3x^{-\frac{1}{3}}}-\frac{x^{\frac{2}{3}}}{x^{\frac{5}{3}}-x^{\frac{2}{3}}}-\frac{x+1}{x^2-4x+3}$.

4) $\frac{\left(\sqrt{a}+\sqrt{b}\right)^2-4b}{(a-b):\left(\sqrt{\frac{1}{b}}+3\sqrt{\frac{1}{a}}\right)}:\frac{a+9b+6\sqrt{ab}}{\frac{1}{\sqrt{b}}+\frac{1}{\sqrt{a}}}$.

5) $x\cdot\frac{1+\frac{2}{\sqrt{x+4}}}{2-\sqrt{x+4}}+\sqrt{x+4}+\frac{4}{\sqrt{x+4}}$.

6) $\left(\frac{1+\sqrt{x}}{\sqrt{1+x}}-\frac{\sqrt{1+x}}{1+\sqrt{x}}\right)^2-\left(\frac{1-\sqrt{x}}{\sqrt{1+x}}-\frac{\sqrt{1+x}}{1-\sqrt{x}}\right)^2$.

7) $\left(\frac{1}{\sqrt{a}+\sqrt{a+1}}+\frac{1}{\sqrt{a}-\sqrt{a-1}}\right):\left(1+\sqrt{\frac{a+1}{a-1}}\right)$.

8) $\frac{x-1}{x^{\frac{3}{4}}+x^{\frac{1}{2}}}\cdot\frac{x^{\frac{1}{2}}+x^{\frac{1}{4}}}{x^{\frac{1}{2}}+1}\cdot x^{\frac{1}{4}}+1$.

9) $\left(\frac{1+x+x^2}{2x+x^2}+2-\frac{1-x+x^2}{2x-x^2}\right)^{-1}\cdot(5-2x^2)$.

10) $\sqrt{\frac{2a}{(1+a)\sqrt[3]{1+a}}}\cdot\sqrt[3]{\frac{4+\frac{8}{a}+\frac{4}{a^2}}{\sqrt{2}}}$.

Питання для самоконтролю

1. Наведіть порядок виконання операцій у Maxima.
2. Перерахуйте арифметичні операції у Maxima.
3. Наведіть як задаються основні математичні константи у Maxima.
4. Наведіть приклади статистичних функцій у Maxima.
5. Охарактеризуйте способи завдання тригонометричних функцій.
6. Чи можна задати свою функцію у Maxima?
7. Наведіть функції для завдання експоненти та логарифму.
8. Охарактеризуйте функції перетворення алгебраїчних виразів у Maxima.
9. Чи можна комбінувати кілька команд при перетворенні виразів?
10. Наведіть способи створення випадкових списків.

2. Лабораторна робота. Застосування системи комп'ютерної алгебри Maxima до розв'язання задач аналізу

Мета: засвоїти можливості системи комп'ютерної алгебри Maxima до розв'язання рівнянь, нерівностей, систем та задач математичного аналізу.

Теоретичні відомості та методичні рекомендації

У Maxima розв'язування рівнянь і нерівностей здійснюється за допомогою спеціальних команд.

Команда

`solve(рівняння, змінна);`

розв'язує задане рівняння в аналітичному вигляді. Її можна застосовувати і для розв'язання параметричних і тригонометричних рівнянь. Наприклад (див. рис. 2.1):

```
(%i1) solve(2·x − 4 = 0, x);
(%o1) [x=2]

(%i2) solve(x^2 − 5·x + 6 = 0, x);
(%o2) [x=3,x=2]

(%i3) solve(a·x + b = 0, x);
```

$$(\%o3)\quad \left[x=-\left(\frac{b}{a}\right)\right]$$

Рисунок 2.1 – Приклади використання команди `solve`

У Maxima є багато додаткових пакетів, які розширюють її можливості – для роботи з графіками, рівняннями, нерівностями, статистикою, матрицями, теорією чисел та логікою. Наприклад, команда `load("solve_rat_ineq");` у Maxima завантажує додатковий пакет, який дозволяє розв'язувати раціональні нерівності – тобто нерівності, у яких є дроби, чисельник і знаменник, вирази з змінними в знаменнику тощо (див. рис. 2.2).

```
(%i11) load("solve_rat_ineq");
       solve_rat_ineq(7·x+2<−x−3);
(%o10)
       C:/maxima−5.47.0/share/maxima/5.47.0/share/solve_rat_ineq/solve_rat_ineq.mac
```

$$(\%o11)\quad \left[\left[x<-\left(\frac{5}{8}\right)\right]\right]$$

Рисунок 2.2 – Приклади використання модуля `solve_rat_ineq`

Коли це потрібно? Звичайна команда `solve(...)` не завжди коректно працює з раціональними нерівностями. Наприклад, вона може:

– не враховувати точки, де вираз не визначений;
– давати неповний розв'язок.

Після завантаження пакету `solve_rat_ineq`, Maxima правильно враховує знаки чисельника і знаменника, та знаходить усі інтервали, на яких виконується нерівність. Команда `load("solve_rat_ineq")` активує можливість точного розв'язування раціональних нерівностей у Maxima. Це корисно для навчальних і аналітичних задач з ОДЗ.

Аналогічно можна покращити вихідний результат при розв'язання диференціальних рівнянь. Модуль `contrib_ode` призначений для розв'язування звичайних диференціальних рівнянь (див. рис. 2.3).

```
(%i16) load("contrib_ode");
       contrib_ode('diff(y,x) = y^2, y, x);
(%o15) C:/maxima-5.47.0/share/maxima/5.47.0/share/contrib/diffequations/contrib_ode.mac
```

$$\text{(\%o16)}\quad \left[-\left(\frac{1}{y}\right)=x+\%c\right]$$

Рисунок 2.3 – Приклади використання модуля `contrib_ode`

Системи рівнянь теж можна розв'язувати за допомогою команди (див. рис. 2.4):

```
solve([рівняння_1, рівняння_2], [x, y]);
```

```
(%i17) solve([x + y = 5, x - y = 1], [x, y]);
(%o17) [[x=3,y=2]]
```

Рисунок 2.4 – Приклади розв'язання системи

У таблиці 2.1 наведено основні альтернативні команди.

Таблиця 2.1 – Альтернативні команди в Maxima

Команда	Призначення
`algsys(...)`	Розв'язування систем алгебраїчних рівнянь (потужніше)
`find_root(...)`	Чисельне розв'язання на інтервалі
`to_poly_solve(...)`	Покращене символьне розв'язання

Як висновок, якщо треба знайти точні розв'язки – то тоді треба використовувати команду `solve`. Якщо потрібні наближені значення – треба обернути результат у `float(...)`.

У Maxima для знаходження границі функції використовується команда (див. рис. 2.5):

```
limit(expr, x, a)
```

де

– `expr` – функція, границю якої шукаємо;
– `x` – змінна;
– `a` – точка, до якої прямує змінна.

```
(%i1) limit(sin(5·x)/x, x, 0);
(%o1) 5
```

Рисунок 2.5 – Знаходження границі функції

Інтерфейс wxMaxima підтримує другий варіант обчислення границі функції, використовуючи меню Аналіз – Знайти границю (див. рис. 2.6).

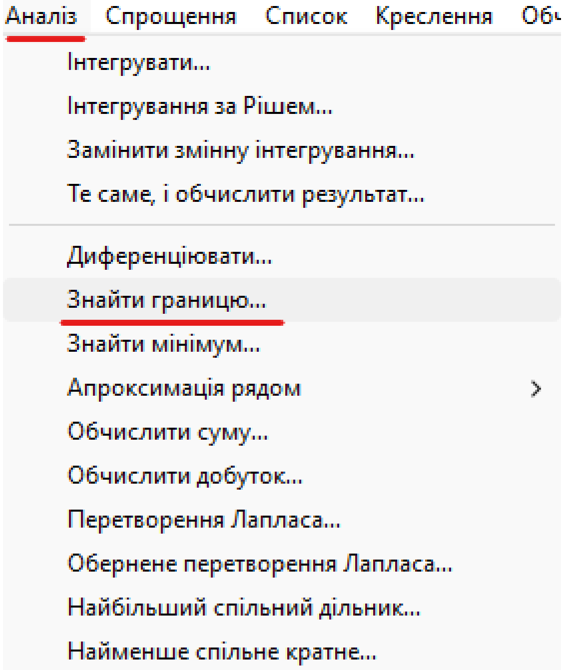

Рисунок 2.6 – Меню Аналіз – Знайти границю

З’явиться вікно, яке треба заповнити (див. рис. 2.7) і по натисканню кнопки «Гаразд» wxMaxima виведе результат аналогічний до рисунку 2.5.

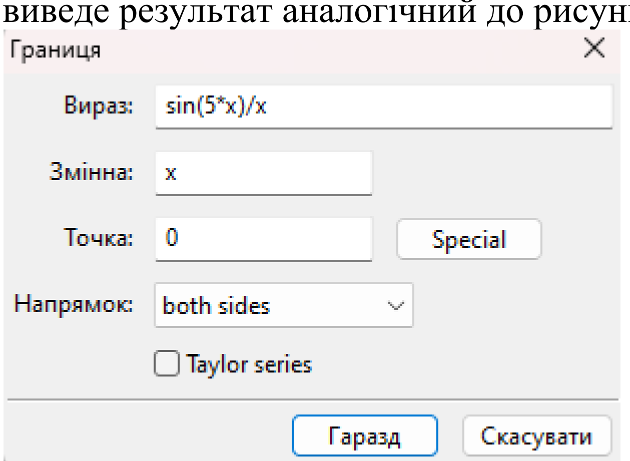

Рисунок 2.7 – Вікно «Границя»

Maxima підтримує обчислення односторонніх границь (див. рис. 2.8):

$\lim\limits_{x \to a-0} f(x)$ еквівалентно команді: `limit(`$f(x)$`, `x`, `a`, minus);`

$\lim\limits_{x \to a+0} f(x)$ еквівалентно команді: `limit(`$f(x)$`, `x`, `a`, plus);`

```
(%i4) limit(1/x, x, 0, minus);
(%o4) - ∞

(%i5) limit(1/x, x, 0, plus);
(%o5) ∞
```

Рисунок 2.8 – Знаходження односторонніх границь функції

Для нескінченності аналогічно (див. рис. 2.9):

$\lim\limits_{x \to -\infty} f(x)$ еквівалентно команді: `limit(`$f(x)$`, `x`, minf);`

$\lim\limits_{x \to +\infty} f(x)$ еквівалентно команді: `limit(`$f(x)$`, `x`, inf);`

```
(%i6) limit((1 + 1/x)^x, x, inf);
(%o6) %e
```

Рисунок 2.9 – Знаходження границі функції на нескінченності

Якщо Maxima не може обчислити границю, то вона залишає `limit(...)` без обчислення. У такому випадку треба спробувати спростити вираз, наприклад, `ratsimp(...)`, або використати команду для обчислення односторонньої границі, або задати параметри явно, наприклад, `assume(x>0)`.

У Maxima похідну функції можна знайти за допомогою команди (див. рис. 2.10):

```
diff(expr, x, n)
```

де

– `expr` – функція, похідну яку треба обчислити;
– x – змінна, за якою береться похідна;
– `n` – порядок похідної.

```
(%i1) diff(exp(x)·x, x, 2);
              x        x
(%o1) x %e  + 2 %e
```

Рисунок 2.10 – Знаходження другої похідної функції

Частинну похідну також можна знайти, використавши команду `diff`, вказавши через кому кожну змінну і порядок похідної по цій змінній (див. рис. 2.11).

```
(%i2) diff(x^5·y^3,x,2,y,3);
              3
(%o2) 120 x
```

Рисунок 2.11 – Знаходження частинної похідної функції

Інтерфейс wxMaxima підтримує другий варіант обчислення похідної функції, використовуючи меню Аналіз – Диференціювати (див. рис. 2.12).

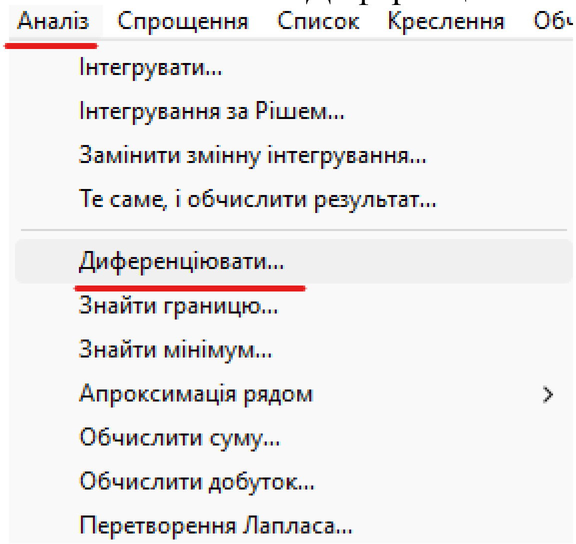

Рисунок 2.12 – Меню Аналіз – Диференціювати

З’явиться вікно, яке треба заповнити і по натисканню кнопки «Гаразд» wxMaxima виведе результат (див. рис. 2.13).

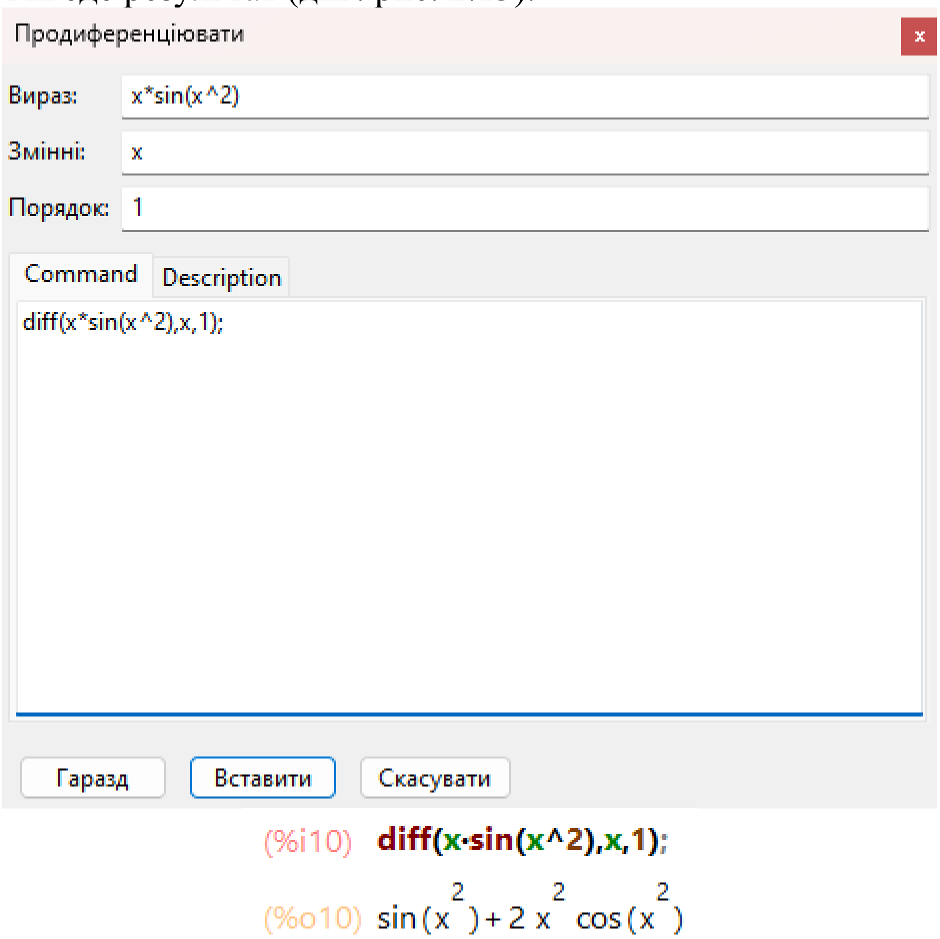

Рисунок 2.13 – Вікно «Продиференціювати»

Невизначений інтеграл можна знайти командою:

```
integrate(expr, x);
```

а визначений інтеграл командою (див. рис. 2.14):

```
integrate(expr, x, a, b);
```

де

– expr – функція, інтеграл від якої треба обчислити;
– x – змінна, за якою обчислюється інтеграл;
– `a, b` – межі інтегрування.

```
(%i11) integrate(sqrt(1 - x^2), x, 0, 1);
```

$$(\%o11)\ \frac{\pi}{4}$$

Рисунок 2.14 – Знаходження визначеного інтеграла

Якщо Maxima не може знайти інтеграл, то вона залишає `integrate(...)` без обчислення. Для числового результату після визначеного інтегралу можна використати команду `float(integrate(...))`.

Завдання до лабораторної роботи

1. Знайти розв'язок рівняння в аналітичному вигляді у СКА Maxima:

1) $\frac{x^2+1}{x-4}-\frac{x^2-1}{x+3}=23;\quad \left(1+\frac{1}{2x}\right)\lg 3+\lg 2=\lg\left(27-3^{\frac{1}{x}}\right).$

2) $\frac{b}{x-a}+\frac{a}{x-b}=5;\quad 3\log_5 2+2-x=\log_5(3^x-5^{2-x}).$

3) $\frac{x^2+x-5}{x}+\frac{3x}{x^2+x-5}+4=0;\quad \sqrt{\log_3 x^9}-4\log_9\sqrt{3x}=1.$

4) $x^4-\frac{50}{2x^4-7}=14;\quad \log_{1-x}3-\log_{1-x}2-0{,}5=0.$

5) $\frac{1}{x(x+2)}-\frac{1}{(x+1)^2}=\frac{1}{12};\ \lg 5+\lg(x+10)=1-\lg(2x-1)+\lg(21x-20).$

6) $\frac{x-3}{x-1}+\frac{x+3}{x+1}=\frac{x+6}{x+2}+\frac{x-6}{x-2};\quad \log_2 182-2\log_2\sqrt{5-x}=\log_2(11-x)+1.$

7) $\frac{5a}{y+a}+\frac{4a}{y+2a}+\frac{3a}{y+3a}=8;\quad \log_5\sqrt{x-9}-\log_5 10+\log_5\sqrt{2x-1}=0.$

8) $\frac{1}{x^3+2}+\frac{1}{x^3+3}=\frac{1}{12};\quad \lg(x+1{,}5)=-\lg x.$

9) $\frac{x-2}{x-1}+\frac{x+2}{x+1}=\frac{x-4}{x-3}+\frac{x+4}{x+3}-\frac{28}{15};\quad 5^{2(\log_5 2+x)}-2=5^{x+\log_5 2}.$

10) $(x-1)(x^2-3)+(2x-1)(x^2+2)=3;$
$0{,}25^{\log_2\sqrt{x+3}-0{,}5\log_2(x^2-9)}=\sqrt{2(7-x)}.$

2. Знайти розв'язок системи рівнянь в аналітичному вигляді у СКА Maxima:

1) $\begin{cases} 2x+y+z=6,\\ 3x+2y+z=7,\\ (x-1)^3+(y+2)^3+(z-3)^3=7. \end{cases}$

2) $\begin{cases} x^4+6x^2y^2+y^4=136,\\ x^3y+xy^3=30. \end{cases}$

3) $\begin{cases} x^3+y^3=19,\\ (xy+8)(x+y)=2. \end{cases}$

4) $\begin{cases} x^3 + x^3y^3 + y^3 = 17, \\ x + xy + y = 5. \end{cases}$

5) $\begin{cases} \left(\frac{x}{y}\right)^2 + \left(\frac{x}{y}\right)^3 = 12, \\ (xy)^2 + xy = 6. \end{cases}$

6) $\begin{cases} \frac{x}{y} + \frac{x^2}{y^2} + \frac{x^3}{y^3} = 14, \\ x + y = 3. \end{cases}$

7) $\begin{cases} 8x + \frac{8}{y} = 3y^2, \\ y + \frac{1}{x} = 3x^2. \end{cases}$

8) $\begin{cases} x^2 + y^2 - x - y = 102, \\ xy + x + y = 69. \end{cases}$

9) $\begin{cases} xy + \frac{y}{x} = 2(x^2 + y^2), \\ xy - \frac{x}{y} = x^2 + y^2. \end{cases}$

10) $\begin{cases} \sqrt{x^2 + 5} + \sqrt{y^2 - 5} = 5, \\ x^2 + y^2 = 13. \end{cases}$

3. Обчислити границі у СКА Maxima:

1) $\lim\limits_{x\to 1}(1 - x)\,\mathrm{tg}\frac{\pi x}{2}$; $\lim\limits_{n\to+\infty}\frac{2n+1}{3n^2+4n+5}$.

2) $\lim\limits_{x\to 1-0}\mathrm{arctg}\frac{1}{1-x}$; $\lim\limits_{x\to+\infty}\frac{x\sqrt[3]{5x^2}+\sqrt[4]{9x^8+1}}{x^5+\sqrt{x}}$.

3) $\lim\limits_{x\to 1+0}\mathrm{arctg}\frac{1}{1-x}$; $\lim\limits_{x\to 0}\frac{\sin 3x}{-\sin 5x}$.

4) $\lim\limits_{x\to 0}\frac{6x}{2-e^x-\cos 4x}$; $\lim\limits_{x\to+\infty}\frac{5n^2+3n+1}{n^2+6n}$.

5) $\lim\limits_{x\to 0}\frac{\ln(x^2+1)}{\sin 2x}$; $\lim\limits_{x\to+\infty}\frac{5n^2+3n+1}{n^2+6n}$.

6) $\lim\limits_{x\to 0}\frac{e^{4x}-\cos 2x}{\sin 2x}$; $\lim\limits_{x\to+\infty}\frac{\sqrt[4]{x^2+2}-5x^2}{x-\sqrt{x^4-x+1}}$.

7) $\lim\limits_{x\to 0}\frac{\sin 7x}{3x^2}$; $\lim\limits_{x\to+\infty}\left(\frac{x+6}{x-2}\right)^{x+1}$.

8) $\lim\limits_{x\to 0}\frac{\ln(x^2+1)}{\sin 2x}$; $\lim\limits_{n\to+\infty}\frac{n(n+36)}{4n^2+1}$.

9) $\lim\limits_{x\to+\infty}\frac{x+\sqrt[4]{81x^8-1}}{x+4\sqrt{x^5}}$; $\lim\limits_{x\to 0}\frac{\sin^2 2x}{x\sin 2x}$.

10) $\lim\limits_{x\to+\infty}\left(\frac{1-2x}{3-2x}\right)^{4x}$; $\lim\limits_{x\to 0}\frac{e^{4x}-\cos 2x}{\sin 8x}$.

4. Обчислити першу, другу і третю похідну функції у СКА Maxima. Обчислити наближено першу похідну у заданій точці:

1) Від функції $f(x) = \frac{45x^7+12x^3+x^2}{5x^4+x-10}$ у точці $x_0 = -20$.

2) Від функції $f(x) = \frac{\cos^5 x}{\sqrt{x^3}}$ у точці $x_0 = -5$.

3) Від функції $f(x) = (-3x^3 + 5x)\sin 5x^2$ у точці $x_0 = 2$.
4) Від функції $f(x) = \ln\left(\frac{1}{2x}\right)$ у точці $x_0 = -1$.
5) Від функції $f(x) = 4x \cdot \operatorname{arctg} x^3$ у точці $x_0 = -\pi$.
6) Від функції $f(x) = x^{-45x-9} + x^5 - e^{3x}$ у точці $x_0 = 13$.
7) Від функції $f(x) = \operatorname{ctg}\left(\frac{1}{3x^2}\right)$ у точці $x_0 = 16$.
8) Від функції $f(x) = e^{3\sin x - 7\cos x}$ у точці $x_0 = \frac{\pi}{12}$.
9) Від функції $f(x) = \frac{\operatorname{arctg} 4x}{16x^2+1}$ у точці $x_0 = -5$.
10) Від функції $f(x) = \log_5(18\arcsin x^2)$ у точці $x_0 = -4$.

5. Обчислити наближено визначений інтеграл у СКА Maxima:
1) Від функції $f(x) = e^{-3x^2}$ на проміжку $[0; 5]$.
2) Від функції $f(x) = \cos 9x^2$ на проміжку $[0; 3]$.
3) Від функції $f(x) = \sin 4x^2$ на проміжку $[0; 2]$.
4) Від функції $f(x) = \frac{1}{\ln 5x}$ на проміжку $[0; 1]$.
5) Від функції $f(x) = \frac{\cos 4x}{x}$ на проміжку $[0; 4]$.
6) Від функції $f(x) = \frac{\sin 9x}{x}$ на проміжку $[0; 7]$.
7) Від функції $f(x) = e^{-12x^2}$ на проміжку $[0; 9]$.
8) Від функції $f(x) = \frac{1}{\ln 11x}$ на проміжку $[0; 8]$.
9) Від функції $f(x) = \frac{\cos 2x}{x}$ на проміжку $[0; 1]$.
10) Від функції $f(x) = \frac{\sin 8x}{x}$ на проміжку $[0; 2]$.

6. Знайти значення інтеграла (невизначеного, визначеного, невласного) у СКА Maxima:

1) a) $\int \cos x \cos 2x \cos 3x \, dx$; b) $\int_0^{\frac{\pi}{2}} x \cos x \, dx$; c) $\int_1^{+\infty} \frac{dx}{\sqrt{x}}$.
2) a) $\int \frac{3x^4+4}{x^2(x^2+1)^3} dx$; b) $\int_{\frac{\pi}{6}}^{\frac{\pi}{4}} \frac{\cos x}{x} dx$; c) $\int_0^{+\infty} \frac{\sin x \cos x dx}{x}$.
3) a) $\int x^3 \sin x \, dx$; b) $\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{dx}{1+\cos x}$; c) $\int_1^{+\infty} \frac{dx}{1+x^2}$.
4) a) $\int \frac{x^3-6}{x^4+6x^2+8} dx$; b) $\int_0^{\frac{\pi}{2}} x^3 \cos x \, dx$; c) $\int_0^{+\infty} \frac{1-e^{-2x^2}}{xe^{x^2}} dx$.
5) a) $\int \operatorname{tg}^2 x \, dx$; b) $\int_0^{\frac{\pi}{2}} \frac{dx}{5-4\sin x+3\cos x}$; c) $\int_1^{+\infty} \frac{\operatorname{arctg} 4x}{x^2} dx$.
6) a) $\int \frac{dx}{\cos^2 x \sin^2 x}$; b) $\int_0^{\frac{\pi}{2}} \frac{\sin x \cos x}{(\cos^2 x+\sin^2 x)^2} dx$; c) $\int_1^{+\infty} \frac{\ln(x^2+1)}{x} dx$.
7) a) $\int \frac{dx}{\cos 2x+\sin^2 x}$; b) $\int_0^{\pi} x^3 \sin x \, dx$; c) $\int_0^{+\infty} e^{-2x} dx$.
8) a) $\int \frac{\cos 2x dx}{\cos^2 x \sin^2 x}$; b) $\int_0^{e} \ln^3 x \, dx$; c) $\int_0^{+\infty} \frac{dx}{x^2+4}$.
9) a) $\int \frac{3x-1}{x^2+9} dx$; b) $\int_4^{9} \frac{x-1}{\sqrt{x}+1} dx$; c) $\int_0^{+\infty} e^{-x}\sqrt{x} dx$.

10) a) $\int \frac{dx}{\sqrt{1+x}+\sqrt{x-1}}$; b) $\int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \frac{x}{\sin^2 x} dx$; c) $\int_1^{+\infty} \frac{x^3+1}{x^4} dx$.

Питання для самоконтролю

1. Наведіть команду у СКА Maxima для розв’язання рівнянь і їх систем. Охарактеризуйте їх можливості.
2. Чи можливо у СКА Maxima розв’язувати нерівності?
3. Розкажіть про особливості знаходження розв’язків диференціальних рівнянь у СКА Maxima.
4. Наведіть команду у СКА Maxima для обчислення границь.
5. Наведіть команду у СКА Maxima для обчислення похідних.
6. Наведіть команду у СКА Maxima для обчислення інтегралів.
7. Охарактеризуйте можливості інтерфейсу wxMaxima для обчислення границь, похідних і інтегралів.

3. Лабораторна робота. Застосування системи комп’ютерної алгебри Maxima до розв’язання задач лінійної алгебри

Мета: засвоїти можливості роботи СКА Maxima до розв’язання задач лінійної алгебри, функції для роботи з матрицями, матричні операції та роботу з масивами.

Теоретичні відомості та методичні рекомендації

У СКА Maxima є вбудовані засоби для роботи з матрицями та векторами, які дозволяють виконувати як базові операції, так і лінійну алгебру (визначники, обернені матриці, власні значення). Для створення матриці з елементами, які задані у вигляді списку `list`, у Maxima використовується функція `matrix(list)` (див. рис. 3.1):

```
(%i1) matrix([-5,2],[1,-4]);
```

(%o1) $\begin{pmatrix} -5 & 2 \\ 1 & -4 \end{pmatrix}$

Рисунок 3.1 – Приклад форми виклику функції `matrix()`

Вектор задається як список (див. рис. 3.2):

```
(%i2) v : [1, 2, 3];
v     [1,2,3]
```

Рисунок 3.2 – Завдання вектору у СКА Maxima

Нехай задано дві матриці A і B. Базові операції над матрицями наступні:

1) $A + B$ – додавання;
2) $A - B$ – віднімання;
3) $A.B$ – матричне множення (див. рис. 3.3);
4) $k * A$ – множення числа на матрицю (див. рис. 3.3);
5) $A\hat{}\hat{} - 1$ – обчислення оберненої матриці до матриці A;
6) функція `invert(A)` – другий спосіб обчислення оберненої матриці до матриці A (див. рис. 3.4);
7) функція `transpose(A)` – транспонування матриці A;
8) функція `determinant(A)` – обчислення визначника матриці A;
9) функція `eigenvalues(A)` – обчислення власних значень матриці A (див. рис. 3.5);
10) функція `minor(A)` – обчислення мінору матриці A;
11) функція `rank(A)` – обчислення рангу матриці A;
12) функція `submatrix()` використовується для створення нової матриці з існуючої, вилучаючи вказані рядки та стовпці;
13) функція `echelon()` перетворює матрицю у форму рядків, що ступінчасто зменшуються, подібно до методу Гауса.

Щоб використовувати розширену лінійну алгебру, треба підключити модуль `load("linearalgebra")`. Щоб отримати у відповіді наближені значення, використовуйте команду `float(…)`.

```
(%i3) A : matrix([1, 2], [3, 4]);
```

A $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

```
(%i4) B : matrix([2, 0], [1, 3]);
```

B $\begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix}$

```
(%i5) A.B;
```

(%o5) $\begin{pmatrix} 4 & 6 \\ 10 & 12 \end{pmatrix}$

```
(%i6) 5·A;
```

(%o6) $\begin{pmatrix} 5 & 10 \\ 15 & 20 \end{pmatrix}$

Рисунок 3.3 – Матричне множення і множення числа на матрицю

```
(%i7) A^^-1;
```

(%o7) $\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\left(\frac{1}{2}\right) \end{pmatrix}$

```
(%i8) invert(A);
```

(%o8) $\begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\left(\frac{1}{2}\right) \end{pmatrix}$

Рисунок 3.4 – Обчислення оберненої матриці

```
(%i9) eigenvalues(A);
```

(%o9) $\left[\left[-\left(\frac{\sqrt{33}-5}{2}\right),\frac{\sqrt{33}+5}{2}\right],[1,1]\right]$

Рисунок 3.5 – Обчислення власних значень матриці

У СКА Maxima є три типу даних: списки, множини та масиви.

Список – це впорядкована сукупність довільних об'єктів, у тому числі, можливо, і самих списків. Список може бути пустим. Щоб створити список, необхідно через кому перерахувати всі об'єкти у квадратних дужках. До елементів списку можна звертатися після його створення, вказавши у квадратних дужках його порядковий номер (див. рис. 3.6):

```
(%i26) n:[2,4,6,1,3,5];
n      [2,4,6,1,3,5]

(%i27) n[6];
(%o27) 5
```

Рисунок 3.6 – Приклад списку та звертання до елемента списку

Список можна створювати за допомогою функції:

`makelist(form,i,a,b)`,

де `form` – список, який залежить від параметру `i` (належить цілим числам); `i` – індекс; `a` та `b` – діапазон зміни параметру `I`(див. рис. 3.7).

```
(%i28) L:makelist(i^^2,i,1,5);
L      [1,4,9,16,25]
```

Рисунок 3.7 – Приклад форми виклику функції `makelist()`

До елементів масиву можна звертатися після його створення, вказавши у квадратних дужках його порядковий номер елемента, який називається індексом.

Масив – це впорядкована послідовність величин. Якщо виникає необхідність використовувати індексовану змінну `x[k]`, не визначаючи `x`, то `x` буде масивом у Maxima. Масив можна створювати за допомогою функції:

`array(name,n,k)`,

де `name` – ім'я масиву; `n`, `k` (належать цілим числам) – розмірність масиву.

```
(%i29) array(a,1,1);
(%o29) a

(%i33) a[0,0]:-1;a[0,1]:3;a[1,0]:5;a[1,1]:-2;
(%o30) -1
(%o31) 3
(%o32) 5
(%o33) -2

(%i34) a[1,0];
(%o34) 5
```

Рисунок 3.8 – Приклад форми виклику функції `array()`

До елементів масиву можна звертатися після його створення, вказавши у квадратних дужках його порядковий номер елемента, який називається індексом (див. рис. 3.8).

У деяких випадках потрібно масив перетворити у список, для цього використовується функція `listarray()` (див. рис. 3.9):

```
(%i35) listarray(a);
(%o35) [-1,3,5,-2]
```

Рисунок 3.9 – Приклад форми виклику функції `listarray()`

Умовні оператори.

У Maxima умовні оператори дозволяють виконувати різні дії залежно від виконання певної умови. Вони використовуються як у виразах, так і у визначенні функцій.

Основна форма умовного оператора:

```
if cond_1 then expr_1 else expr_0.
```

Якщо умова `cond_1` виконується, то виконується вираз `expr_1`, інакше – виконується вираз `expr_0`. Пакет Maxima надає змогу використати різні форми оператора `if`, наприклад:

```
if cond_1 then expr_1 elseif cond_2 then expr_2 elseif
... else expr_0
```

Якщо виконується умова `cond_1`, то виконується вираз `expr_1`, інакше – перевіряється умова `cond_2`, і якщо вона виконується – виконується вираз `expr_2`, тощо. Якщо жодна з умов не виконується – виконується вираз `expr_0`.

Альтернативні вирази `expr_1`, `expr_2`, ... , `expr_n` – довільні вирази Maxima (зокрема вкладені оператори `if`). Умови – дійсно або потенційно логічні вирази, що зводяться до значень `true` або `false`. Спосіб інтерпретації умов залежить від значення прапорця `prederror`. Якщо `prederror=true`, виводиться помилка, якщо значення якогось із виразів `cond_1`, ... , `cond_n`, відрізняється від `true` або `false`. Якщо `prederror=false` і значення якогось із виразів `cond_1`, ... , `cond_n`, відрізняється від `true` або `false`, результат обчислення `if` – умовний вираз.

Оператори циклу.

Для виконання ітерацій використовується оператор `for`. Можуть використовуватися три варіанти його виклику, що відрізняються умовою закінчення циклу:

– for variable: init_value step increment thru limit do body;
– for variable: init_value step increment while condition do body;
– for variable: init_value step increment unless condition do body.

Тут `variable` – змінна циклу; `init_value` – початкове значення; `increment` – крок (типово дорівнює 1); `limit` – кінцеве значення змінної циклу; `body` - оператори тіла циклу.

Ключові слова `thru`, `while`, `unless` вказують на спосіб завершення циклу:

– за досягнення змінною циклу значення `limit`;
– поки виконується умова `condition`;
– поки не буде досягнута умова `condition`.

Параметри `init_value`, `increment`, `limit`, і `body` можуть бути довільними виразами. Контрольна змінна по завершенні циклу має бути додатною (при цьому початкове значення може бути і від'ємним). Вирази `limit`, `increment`, умови завершення (`condition`) обчислюються на кожному кроці циклу, тому їхня складність впливає на час виконання циклу.

При нормальному завершенні циклу величина, що повертається – `done` (див. рис. 3.10). Примусовий вихід із циклу здійснюється за допомогою оператора `return`, що може повертати довільне значення.

```
(%i2) s:0$ for i:1 while i <= 10 do s: s+i;
(%o2) done

(%i3) s;
(%o3) 55
```

Рисунок 3.10 – Приклад циклу у СКА Maxima

Контрольна змінна циклу – локальна усередині циклу, тому її зміна у циклі не впливає на контекст (навіть при наявності поза циклом змінної з тією ж назвою).

Наприклад, треба написати програму за допомогою операторів циклу, яка буде розкладати функцію $e^{\sin x}$ у ряд Маклорена до 7 порядку (див. рис. 3.11):

```
(%i13) series: 1$
       term: exp(sin(x))$

       for p:1 unless p > 7 do (
          term: diff (term, x)/p,
          series: series + subst (x=0, term)·x^p
       )$;

       series;
```

(%o13) $\frac{x^7}{90}-\frac{x^6}{240}-\frac{x^5}{15}-\frac{x^4}{8}+\frac{x^2}{2}+x+1$

Рисунок 3.11 – Приклад розкладання функції у ряд Маклорена за допомогою оператору циклу

То й же самий результат можна отримати, використовуючи вбудовану команду `taylor`, яка автоматично будує ряд Тейлора (Маклорена) для будь-якої функції. Вона дозволяє перевірити результат ручного алгоритму, який написано через цикл (див. рис. 3.12):

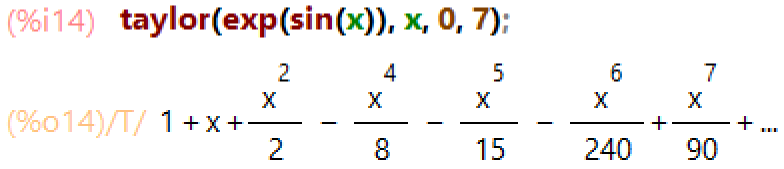

Рисунок 3.12 – Приклад розкладання функції у ряд Маклорена через вбудовану команду `taylor`

Таким чином, система комп’ютерної алгебри Maxima є ефективним засобом для навчання та практичного розв’язання задач лінійної алгебри, дозволяючи поєднувати аналітичні обчислення з наочністю та автоматизацією процесів.

Завдання до лабораторної роботи

1. Для матриць

$$A = \begin{pmatrix} 5 & 16-n & h-4 \\ -2 & -h+n & n+3 \\ 4 & 2h-n & 7 \end{pmatrix},\quad B = \begin{pmatrix} -10 & -n & -h+9 \\ 11 & h & 17-n \\ 18 & n & -2 \end{pmatrix},$$

де n – номер варіанту, h – остання цифра номеру вашої групи, знайти у СКА Maxima: 1) AB; 2) BA; 3) $nA - hB$; 4) $\det A$ та $\det B$; 5) A^{-1} та B^{-1}; 6) ранг матриці A, слід матриці B, власні значення матриць A та B; 7) привести матрицю A до трикутного вигляду, матрицю B до Жорданової форми.

2. Розв’язати матричне рівняння $AX = B$ трьома способами у СКА Maxima (методом Гауса, матричним методом та за формулами Крамера):

$$A = \begin{pmatrix} 5 & 16-n & h-4 \\ -2 & -h+n & n+3 \\ 4 & 2h-n & 7 \end{pmatrix},\quad B = \begin{pmatrix} -10 & -n & -h+9 \\ 11 & h & 17-n \\ 18 & n & -2 \end{pmatrix},$$

де n – номер варіанту, h – остання цифра номеру вашої групи.

3. Знайти суму ряду у СКА Maxima, використовуючи списки:

1) $\sum_{i=0}^{10} \frac{1}{i}$.
2) $\sum_{i=-10}^{10} \frac{i!}{2i+1}$.
3) $\sum_{i=-10}^{10} (2i+1)$.
4) $\sum_{i=0}^{10} \frac{1}{k^i}$, для $k = 10, k = 15$.
5) $\sum_{i=0}^{5} k^{\frac{1}{i}}$, для $k = 2, k = 3$.
6) $\sum_{i=0}^{4} \cos^2(\pi i)$.
7) $\sum_{i=0}^{4} (2^i - 1)$.
8) $\sum_{i=0}^{10} \frac{e^i}{i+1}$.
9) $\sum_{i=1}^{10} (5i^2 - 1)$.

10) $\sum_{i=0}^{5} \sin^2\left(\frac{\pi i}{3}\right)$.

4. Розв'язати задачу, використовуючи списки та масиви СКА Maxima:
1) Вивести значення функції $f(x) = x^2 \cos 3x, x \in [0; 2\pi]$ з шагом $h = 0{,}2\pi$.
2) Знайти суму елементів матриці 4×4.
3) У довільному списку, який містить 10 елементів, замінити всі елементи менші 5 на 0.
4) Знайти добуток мінімальних елементів кожного стовбця матриці 4×4.
5) У довільному списку, який містить 10 елементів, замінити всі елементи більші 3 на 1.
6) У довільному списку, який містить 10 елементів, знайти найменший.
7) Вивести значення функції $f(x) = \sqrt{x} \sin 2x, x \in [0; \pi]$ з шагом $h = 0{,}1\pi$.
8) У довільному списку, який містить 15 елементів, знайти найбільший.

Питання для самоконтролю
1. Охарактеризуйте можливості СКА Maxima для матричних обчислень.
2. Які Ви знаєте базові операції над матрицями?
3. Для чого потрібно підключати модуль `linearalgebra` у СКА Maxima?
4. Чи можна обернену матрицю обчислити наближено у СКА Maxima?
5. Які типи даних є у СКА Maxima?
6. Охарактеризуйте функцію `makelist()`.
7. Для чого використовується функція `array()` у СКА Maxima?
8. Розкажіть про умовні оператори у СКА Maxima.
9. Розкажіть про оператори циклу у СКА Maxima.

4. Лабораторна робота. Візуалізація двовимірної та тривимірної графіки у системі комп'ютерної алгебри Maxima

Мета: засвоїти можливості роботи СКА Maxima до візуалізація двовимірної та тривимірної графіки у СКА Maxima.

Теоретичні відомості та методичні рекомендації

Для побудови двомірних графіків у системі комп'ютерної алгебри Maxima можна використовується функція (див. рис. 4.1, 4.2):

```
plot2d([list_functions], [x, a, b], [y, c, d]),
```

де перший аргумент `list_functions` – список функцій, графік, другий і третій – обмеження по осям координат (якщо не вказати данні обмеження, то графік функції буде побудований автоматично).

```
(%i1) plot2d([x^2, sin(x)], [x, -2·%pi, 2·%pi], [y, -5, 5]);
```

Рисунок 4.1 – Приклад форми виклику функції `plot2d()` у СКА Maxima

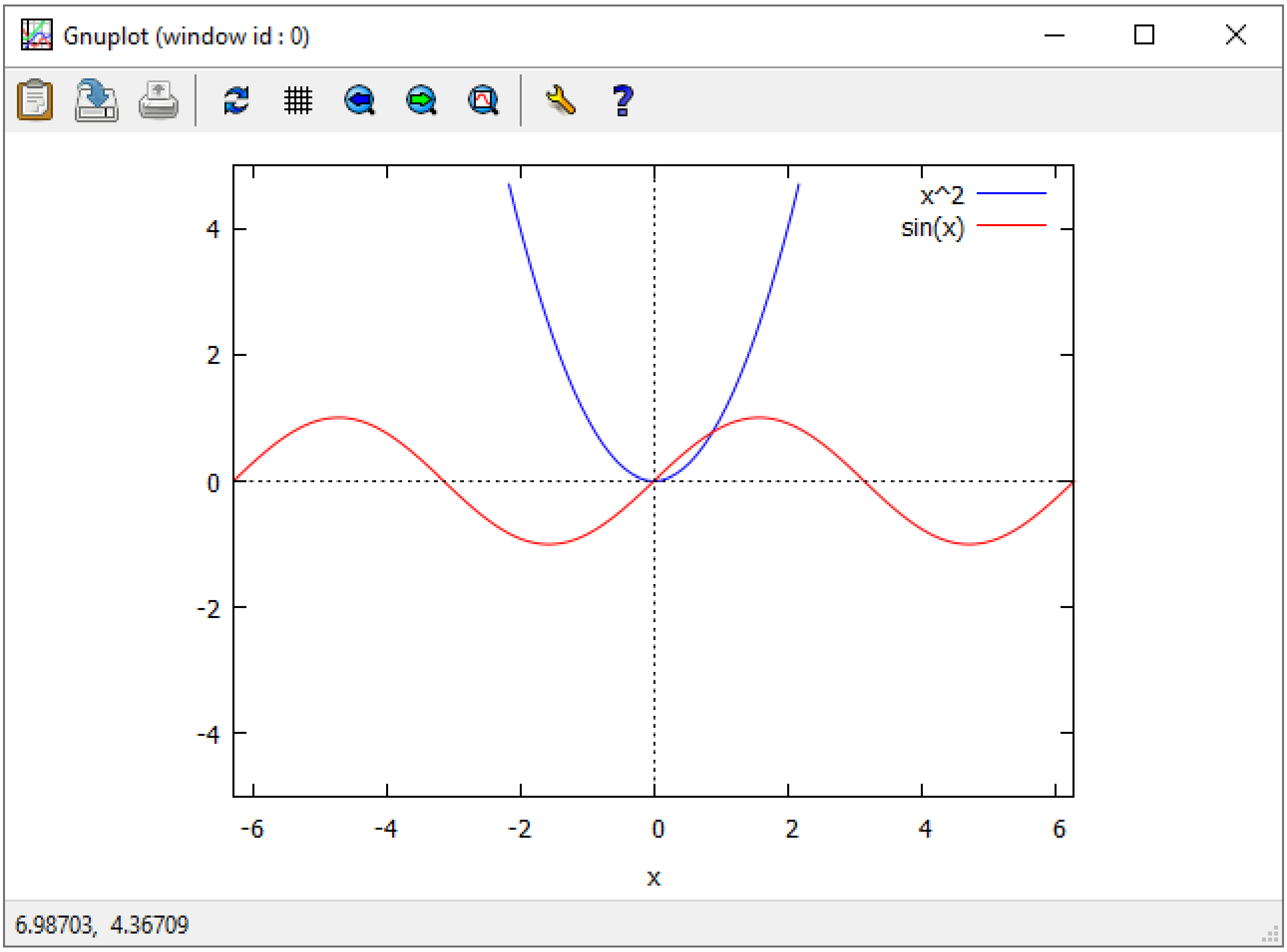

Рисунок 4.2 – Результат виводу функції `plot2d()`

Для побудови графіків функції у СКА Maxima можна використати меню Креслення → Двовимірний графік. Після виконання даної команди з'явиться вікно з формою, яку необхідно заповнити. У цьому випадку графіки будуються автоматично різними кольорами. За допомогою кнопки Додатково можна вибрати в якому вигляді задана функція (графік параметричної функції або графік дискретної функції). Нижче можна налаштувати формат і параметри відображення графіку функції (див. рис. 4.3, 4.4).

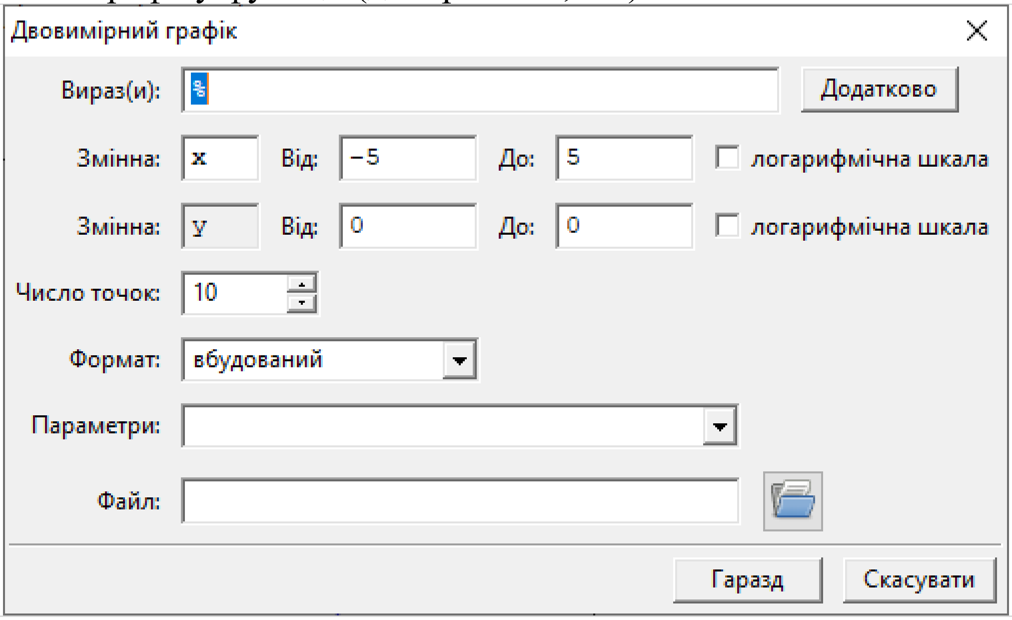

Рисунок 4.3 – Вікно з формою для побудови графіків

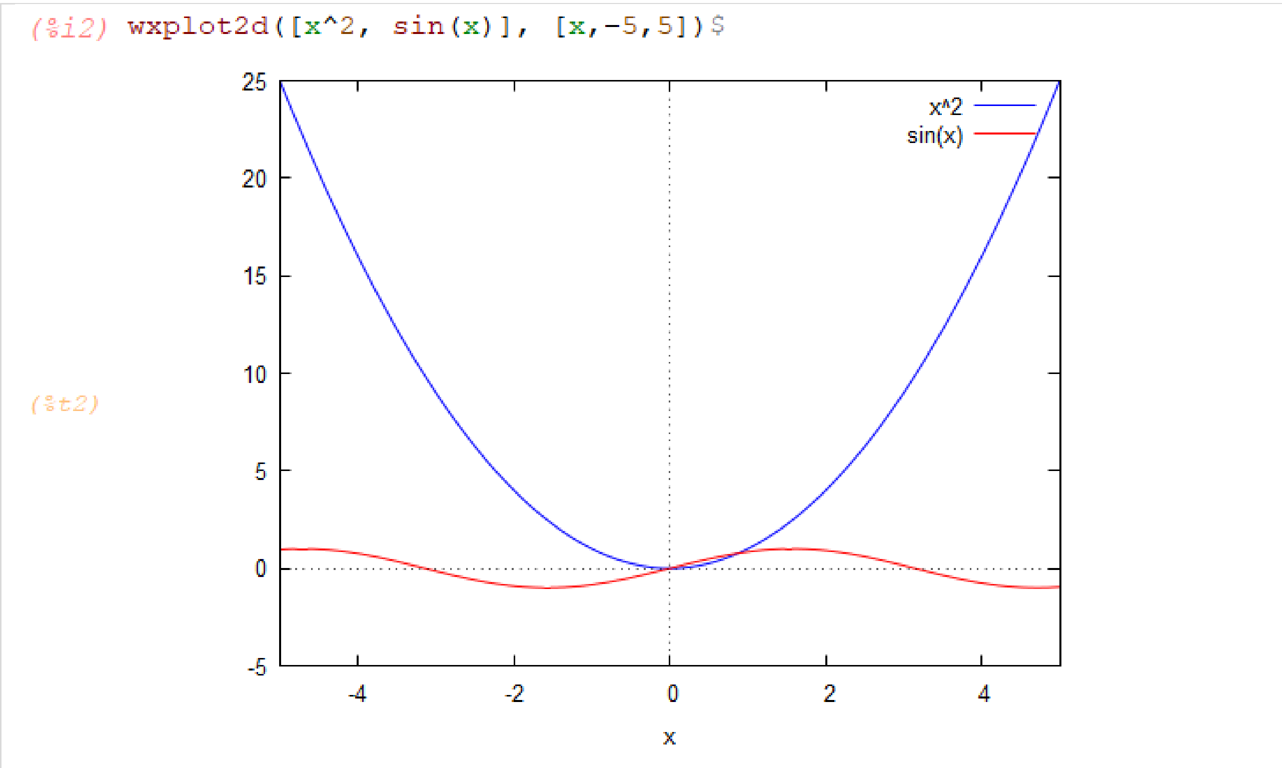

Рисунок 4.4 – Результат виводу функції `plot2d()` через меню Креслення → Двовимірний графік

У СКА Maxima можна будувати лінії у полярній системі координат за допомогою функції `draw2d()`, яка міститься у бібліотеці `draw`. Для цього спочатку треба додатково завантажити цю бібліотеку оператором `load(draw)` (див. рис. 4.5, 4.6).

```
(%i18) load(draw)$

(%i19) draw2d(user_preamble="set grid polar", nticks=500,
        color=blue, xrange=[-2, 2], yrange=[-2, 2],
        polar(2·sin(3·phi), phi, 0, 2·%pi));
(%o19)  [gr2d(polar)]
```

Рисунок 4.5 – Приклад форми виклику функції `draw2d()` у СКА Maxima

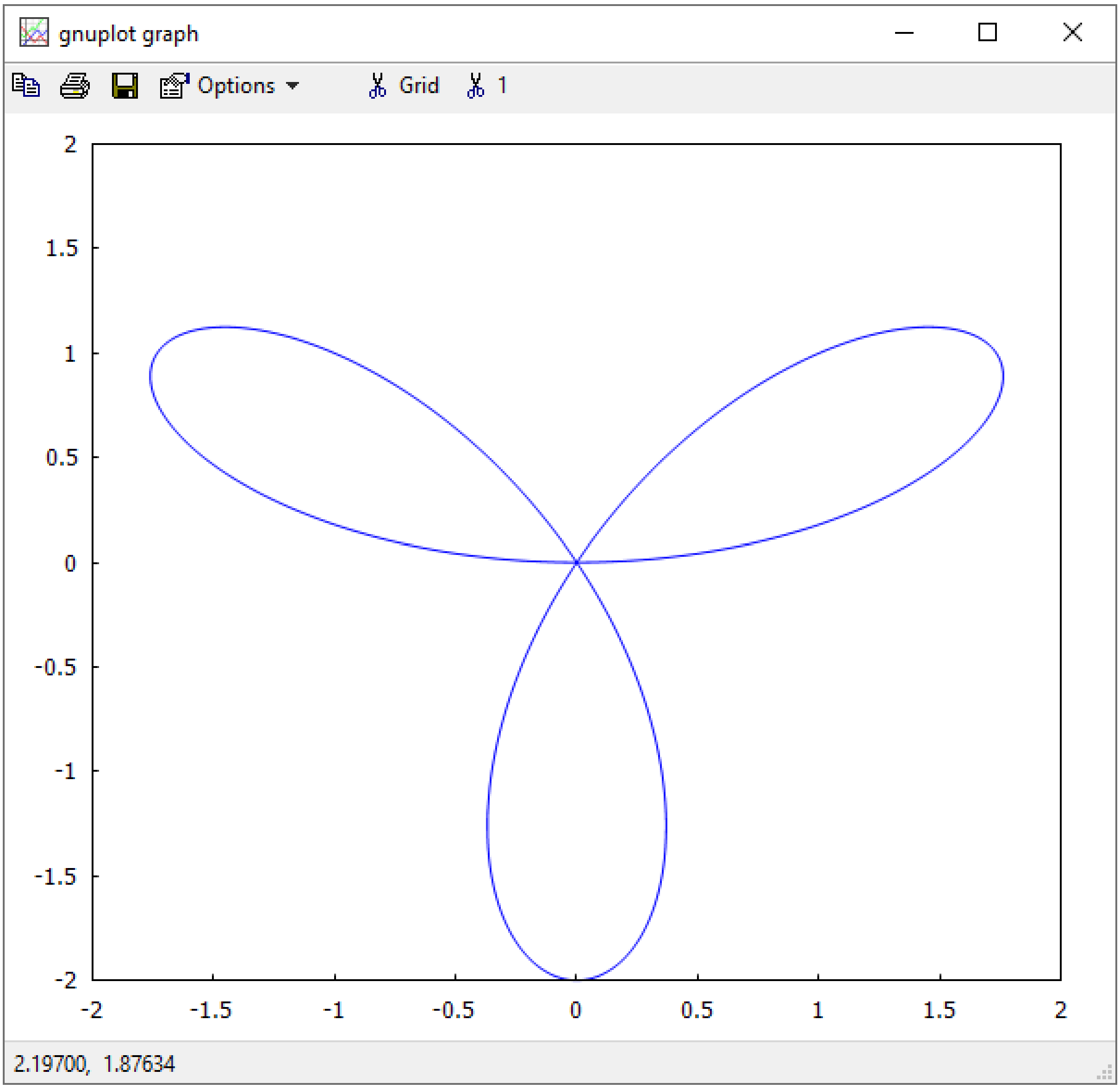

Рисунок 4.6 – Результат виводу функції `draw2d()`.

У СКА Maxima можна будувати графіки функцій, які задано неявно за допомогою функції `implicit_plot()`, яка міститься у бібліотеці `implicit_plot`. Для цього спочатку треба додатково завантажити цю бібліотеку оператором `load(implicit_plot)`.

Для побудови тривимірних графіків функцій використовується функція

```
plot3d(function, [x, a, b], [y, c, d]),
```

де перший аргумент `function` – функція $f(x, y)$, графік якої треба побудувати, другий і третій – обмеження по осям координат (див. рис. 4.7, 4.8).

```
(%i22) plot3d(x·y^2, [x, -5, 5], [y, -5, 5]);
```

Рисунок 4.7 – Приклад форми виклику функції `plot3d()` у СКА Maxima

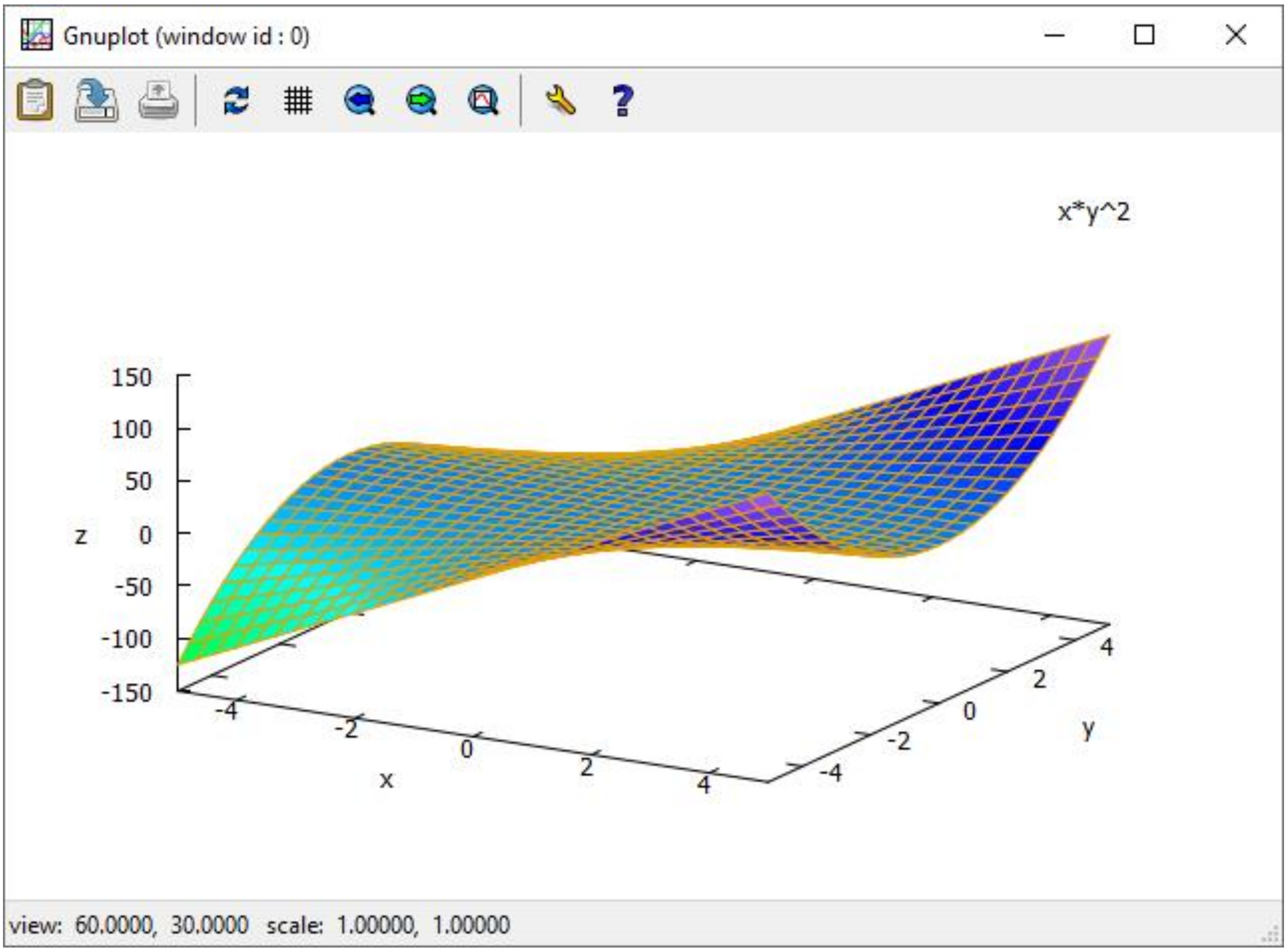

Рисунок 4.8 – Результат виводу функції `plot3d()`

Завдання до лабораторної роботи

1. Побудувати графіки функцій, використовуючи основні опції функції `plot2d()` у декартовій системі координат:

1) a) $y = \frac{x^5}{(x^2-1)^2}$; b) $y = \sin x + \frac{1}{2}\sin 2x + \frac{1}{3}\sin 3x$.

2) a) $y = \sqrt[3]{x(3-x)^2} - x$; b) $y = \left(\frac{x+1}{x-1}\right)^4$.

3) a) $y = \frac{20x^2}{(x-1)^3}$; b) $y = \frac{\ln^2 x}{x}$.

4) a) $y = 8x^2(x^2-1)^3$;b) $y = \frac{x}{2} - \arccos\frac{2x}{1+x^2}$.

5) a) $y = \frac{x^5-8}{x^4}$; b) $y = \arccos\frac{1-x^2}{1+x^2}$.

6) a) $y = \frac{(x-5)^3}{(x-7)^2}$; b) $y = \frac{x}{\ln x}$.

7) a) $y = \frac{x^3-2x^2-x+2}{x}$; b) $y = \frac{x^2\sqrt{x^2-1}}{2x^2-1}$.

8) a) $y = \frac{x^3}{x^2-1}$; b) $y = (x^2-2)e^{-2x}$.

9) a) $y = \frac{1+x^2}{1+(x-2)^2}$; b) $y = \ln\left|\frac{1-x}{1+x}\right| + \frac{6}{1+x}$.

10) a) $y = \frac{x^2+2x-3}{x} - e^{\frac{1}{x}}$; b) $y = \frac{x}{2} + 2\operatorname{arctg} x$.

2. Побудувати графіки функцій, заданих у параметричній системі координат:

1) $x = \frac{9}{2}\cos^3 t, y = \frac{9}{4}\sin^3 t, t \in [0; 2\pi]$.

2) $x = t^3 - 2, y = t - 2, t \in [0; 10]$.

3) $x = 5\left(\cos t + \ln\left(\operatorname{tg}\frac{t}{2}\right)\right), y = 5\sin t, t \in [0; 2\pi]$.

4) $x = 2(t - \sin t), y = 2(1 - \cos t), t \in [0; 2\pi]$.

5) $x = t - \frac{\operatorname{sh} 2t}{2}, y = 2\operatorname{ch} t, t \in [0; 15]$.

6) $x = (t^2-2)\sin t + 2t\cos t, y = (t^2-2)\cos t - 2t\sin t, t \in [0; \pi]$.

7) $x = 3(\operatorname{sh} t - t), y = 3(\operatorname{ch} t - 1), t \in [0; 21]$.

8) $x = \cos^3 t, y = \sin^3 t, t \in \left[0; \frac{\pi}{2}\right]$.

9) $x = 2t^3(1-t^2), y = t\sqrt{7}, t \in [0; 12]$.

10) $x = 2t^2, y = t\left(\frac{1}{4} - t^2\right), t \in [0; 20]$.

3. Побудувати графіки функцій, заданих неявно:

1) $x^2y^2 + \sin x = 1$.

2) $x^2y + y^2 = 2$.

3) $4x^2 + 7y^2 - 10x + 20y = 8$.

4) $x\ln y + y^3 = \sqrt{2}$.

5) $xe^y + 2xy = e$.

6) $3x^2 - y^2 - x + 2y = 4$.

7) $x^5y^3 - \cos 2x = 8$.

8) $-x^4 + 3y^2 - xy + 6y = 1$.

9) $(x^2+y^2)(xy+3x^2) = 1$.

10) $x^2y^2 + \operatorname{tg} y = 1$.

4. Побудувати графіки функцій, заданих у полярній системі координат:

1) $\rho = 5\sin^2 2\varphi$.

2) $\rho = 7\cos^2\varphi$.

3) $\rho = 2(1 + \sin^2\varphi)$.

4) $\rho = 4\sin 2\varphi$.

5) $\rho = 9\cos 5\varphi$.

6) $\rho = 4(1 + \cos\varphi)$.

7) $\rho = 3\cos 3\varphi$.

8) $\rho = 8\cos 4\varphi$.

9) $\rho = 7\sin 5\varphi$.

10) $\rho = 2\sin 4\varphi$.

5. Побудувати поверхні, використовуючи основні опції функції `plot3d()` у декартовій системі координат:

1) $z = 2x^2 + 9y^2$.
2) $z = 3 - (x^2 + y^2)$.
3) $z = 4 - x^2$.
4) $z = \frac{y^2}{5}$.
5) $z = 2x^2 - 3y^2$.
6) $z = 12 + x^2 + 2y^2$.
7) $z = 2x^2 - 3$.
8) $z = 4x^2 + \frac{y^2}{3} + 1$.
9) $z = \frac{x^2}{3} - 6$.
10) $z = 6x^2 - y^2$.

Питання для самоконтролю

1. Які функції дозволяють побудови графіки функцій у СКА Maxima?
2. У яких системах координат можна побудувати графік функції у СКА Maxima?
3. Для чого потрібна бібліотека `draw`? Як її підключити?
4. Чи можна будувати графіки функцій у полярній системі координат?
5. Як побудувати графік функції, заданої неявно?
6. Розкажіть про особливості побудови поверхні у СКА Maxima.

5. Лабораторна робота. Застосування системи комп'ютерної алгебри Maxima до статистичних задач у теорії ймовірностей

Мета: ознайомлення з використанням можливостей роботи СКА Maxima для статистичного аналізу даних, вивчення основних характеристик вибірки, побудова розподілів і використання моделей у задачах теорії ймовірностей.

У Maxima є можливості для обробки статистичних даних, хоча вони не такі розвинені, як у спеціалізованих пакетах типу R або Python (NumPy/Pandas). Проте система дозволяє розв'язувати багато базових і прикладних задач статистики.

Теоретичні відомості та методичні рекомендації

СКА Maxima може наступне:

1. Описова статистика. СКА Maxima може обчислювати:

– середнє арифметичне, наприклад, (див. рис. 5.1):

```
(%i1) mean([1,2,3,4,5]);
(%o1) 3
```

Рисунок 5.1 – Результат виводу функції `mean()`

– медіану, наприклад, (див. рис. 5.2):

```
(%i2) median([1,2,3,4,5]);
(%o2) 3
```

Рисунок 5.2 – Результат виводу функції `median()`

– дисперсію і стандартне відхилення, наприклад, (див. рис. 5.3):

```
→ variance([1,2,3,4,5]);
  std([1,2,3,4,5]);
(%o3) variance([1,2,3,4,5])
(%o4) √2
```

Рисунок 5.3 – Результат виводу функцій `variance()` і `std()`

– мінімум і максимум, наприклад, (див. рис. 5.4):

```
(%i6) lmin([1,2,3,4,5]);
      lmax([1,2,3,4,5]);
(%o5) 1
(%o6) 5
```

Рисунок 5.4 – Результат виводу функцій `lmin()` і `lmax()`

2. Робота з випадковими величинами. СКА Maxima підтримує генерацію випадкових чисел і обчислення ймовірностей:

– випадкове число від 0 до 1, наприклад, `random(1.0);`

– випадкове ціле число від 0 до 100, наприклад, random(100);
– множина випадкових чисел (набір вибірки), наприклад, 20 випадкових чисел від 0 до 99 (див. рис. 5.5):

```
(%i7) makelist(random(100), i, 1, 20);
(%o7) [67,29,85,34,24,27,23,70,88,61,53,7,78,44,3,74,31,28,12,30]
```

Рисунок 5.5 – Результат виводу функції makelist(random())

3. Ймовірнісні розподіли. СКА Maxima може працювати з деякими класичними розподілами:

– біноміальний розподіл, наприклад, кількість комбінацій з 5 по 2 (див. рис. 5.6):

```
(%i8) binomial(5,2);
(%o8) 10
```

Рисунок 5.6 – Результат виводу функції binomial()

– ймовірності для дискретних розподілів через формули (Бернуллі, біноміальний, геометричний);
– для неперервних розподілів можна використовувати інтеграли функцій густини.

4. Кореляція та регресія:

– коваріація та кореляція двох вибірок, наприклад, (див. рис. 5.7):

```
(%i10) covariance([1,2,3],[2,4,6]);
       pearson_correlation([1,2,3],[2,4,6]);
(%o9)  covariance([1,2,3],[2,4,6])
(%o10) pearson_correlation([1,2,3],[2,4,6])
```

Рисунок 5.7 – Результат виводу функцій covariance() і pearson_correlation()

– проста лінійна регресія через lsquares_estimates(<дані>, [змінні], <модель>, <незалежна змінна>) (знаходить коефіцієнти рівняння регресії $y = ax + b$). Ще треба завантажити пакет lsquares командою:

```
load("lsquares")
```

5. Візуалізація статистики. Через draw можна будувати:

– діаграми розподілу (гістограми);
– точки вибірки для аналізу регресії.

Наприклад, побудова випадкової вибірки (див. рис. 5.8):

```
load("draw")$
data : makelist(random(100), i, 1, 20)$
draw2d(
    points(data)
);
```

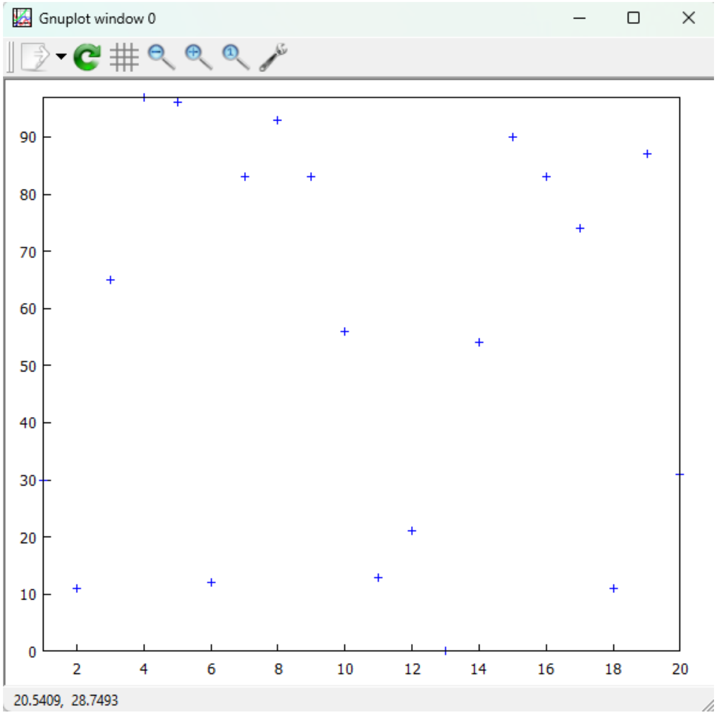

Рисунок 5.8 – Побудова випадкової вибірки

Отже, у Maxima можна розв'язувати такі статистичні задачі:
1) обчислення середнього, медіани, дисперсії, стандартного відхилення;
2) генерація випадкових вибірок і моделювання експериментів;
3) робота з класичними ймовірнісними розподілами;
4) коваріація, кореляція, проста лінійна регресія;
5) візуалізація даних через точки та гістограми.

Завдання до лабораторної роботи

1. Згенерувати випадкову вибірку випадкових величин, об'ємом 100 елементів (цілі числа). Знайдіть:
– середнє значення;
– медіану;
– моду;
– дисперсію;
– стандартне відхилення.

2. Згенеруйте 1000 випадкових значень з біноміального розподілу з параметрами $n = 10, p = 0{,}3$. Обчисліть відносну частоту появи кожного значення. Знайдіть ймовірність того, що випадкова змінна менше або дорівнює 3. Використовуйте команди `binomial_pdf(k,n,p)` або `binomial_cdf(k,n,p)` після `load("distrib")`.

3. Задані експериментальні дані:

```
data: [[1,2], [2,3], [3,5], [4,4], [5,6]];
```

Знайдіть рівняння прямої регресії: $y = ax + b$. Побудуйте графік даних і регресійної прямої.

4. У коробці лежить 5 червоних, 3 синіх і 2 зелених кульки. Витягують 2 кульки без повернення. Знайдіть ймовірність того, що обидві будуть одного

кольору. Моделюйте цей процес 1000 разів і порівняйте експериментальний та теоретичний результат.

Питання для самоконтролю

1. Як обчислити середнє арифметичне засобами СКА Maxima?
2. Як обчислити медіану вибірки засобами СКА Maxima?
3. Як обчислити дисперсію і стандартне відхилення засобами СКА Maxima?
4. Яка функцію допоможе знайти мінімум і максимум у СКА Maxima?
5. Яка функція може згенерувати множину випадкових чисел у СКА Maxima?
6. Чи можна у СКА Maxima обчислювати ймовірнісні розподіли? Охарактеризуйте можливості СКА.
7. Як обчислити кореляцію та регресію засобами СКА Maxima?
8. Охарактеризуйте можливості візуалізація даних через точки та гістограми у СКА Maxima.

6. Лабораторна робота. Реалізація обчислюваних методів у системі комп'ютерної алгебри Maxima

Мета: засвоїти можливості роботи СКА Maxima до написання власних алгоритмів для чисельних та аналітичних розрахунків.

Теоретичні відомості та методичні рекомендації

У СКА Maxima можна реалізовувати обчислювальні методи – і це одна з її сильних сторін. Вона не обмежується лише символьними перетвореннями: Maxima містить повноцінну мову програмування (схожу на Lisp), що дозволяє писати власні алгоритми для чисельних та аналітичних розрахунків.

Можна реалізовувати:

1. чисельні методи аналізу:
   - метод Ньютона (для коренів рівнянь);
   - метод дихотомії;
   - метод січних;
   - чисельне інтегрування (метод трапецій, Сімпсона тощо);
   - чисельне диференціювання;
2. методи лінійної алгебри:
   - Гаусове виключення;
   - LU-, QR-розклад;
   - метод ітерацій для систем рівнянь;
3. методи статистики та теорії ймовірностей:
   - оцінка параметрів (метод моментів, МНК);
   - Монте-Карло моделювання;
   - генерація випадкових вибірок;
4. оптимізаційні алгоритми:
   - градієнтний спуск;
   - лінійне програмування;
   - задачі з обмеженнями.

Можна виділити такі інструменти Maxima для цього:

- цикли та умовні оператори (`for, while, if-then-else`);
- функції користувача (`f(x):= ... або define`);
- пакети з готовими методами, наприклад, `numericalio` (читання/запис чисел і масивів), `lsquares` (метод найменших квадратів), `stats` (статистика), `draw` (візуалізація результатів);
- чисельні функції, наприклад, `find_root` (пошук кореня на інтервалі), `quad_qags` (чисельне інтегрування), `rk` (чисельне розв'язання ОДУ методом Рунге–Кутта).

Приклад. Реалізувати метод Ньютона для рівняння $x^3 - x - 2 = 0$.

```
/* --- оголошуємо функцію і її похідну --- */
f(x) := x^3 - x - 2;
df(x) := diff(f(x), x);   /* або df(x) := 3*x^2 - 1; */
```

```
/* --- реалізація Ньютона (викликаємо через apply) --- */
newton(fun, dfun, x0, eps, maxiter) := block(
  [x : x0, iter : 0, fx, dfx],
  while abs(apply(fun, [x])) > eps and iter < maxiter do (
    fx  : apply(fun, [x]),
    dfx : apply(dfun, [x]),
    if abs(dfx) < 1.0e-12 then (print("Derivative ~ 0 at
x=", x), return(['error, derivative_zero])),
    x : x - fx/dfx,
    iter : iter + 1
  ),
  if iter >= maxiter then (print("Max iterations reached"),
return([x, iter])),
  return([x, iter])
)$
f_lambda  : lambda([t], t^3 - t - 2);
df_lambda : lambda([t], 3*t^2 - 1);

newton(f_lambda, df_lambda, 1.5, 1e-6, 100);
```

Результат – див. рис. 6.1:

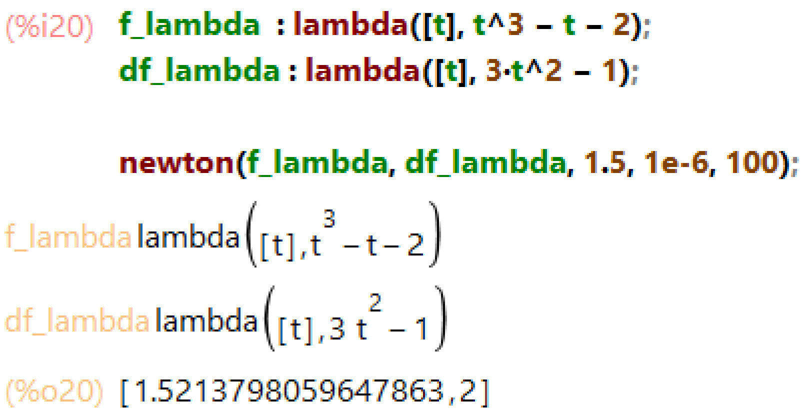

Рисунок 6.1 – Результат роботи програми за методом Ньютона для рівняння

Завдання до лабораторної роботи

1. Ознайомтесь з прикладами реалізації деяких обчислювальних методів з підручника: Чичкарьов Є. А. Підручник-довідник із системи комп’ютерної алгебри Maxima, Розділ 8, стор. 159–177.
2. Реалізуйте наближений метод у системі комп’ютерної алгебри Maxima для заданого рівняння:

1) $x^7 - 5x + 1 = 0$ (метод ділення навпіл);
2) $3x^3 - \sqrt{x+1} = 0$ (метод простих ітерацій);
3) $x^4 - x^3 - 9 = 0$ (модифікований метод Ньютона);
4) $5x^7 - \sqrt{x^3+1} = 0$ (метод хорд);
5) $6x^5 + 11x - 7 = 0$ (метод ділення навпіл);
6) $9x^4 - \sqrt{x^2+1} = 0$ (метод простих ітерацій);

7) $7x^3 + 6x + 1 = 0$ (модифікований метод Ньютона);
8) $6x^2 - \sqrt{x^2 + 9} = 0$ (метод хорд);
9) $4x^5 - 2x^3 + 3 = 0$ (метод ділення навпіл);
10) $3x^5 - \sqrt{x^2 + 11} = 0$ (метод простих ітерацій).

3. Реалізуйте наближений метод у системі комп'ютерної алгебри Maxima для заданого інтеграла:

1) $\int_0^1 \cos(5x^2)\, dx$ (метод середніх прямокутників);
2) $\int_0^1 \sin(11x^2)\, dx$ (метод прямокутників);
3) $\int_2^3 \ln\frac{1}{\sqrt{x}}\, dx$ (метод трапецій);
4) $\int_4^9 \ln\left(\frac{1}{\sqrt{x}+x}\right) dx$ (метод Сімпсона);
5) $\int_0^1 \cos(x^2 + 1)\, dx$ (метод середніх прямокутників);
6) $\int_4^8 \ln\frac{1}{\sqrt{x+1}}\, dx$ (метод прямокутників);
7) $\int_1^2 \sin\left(\frac{1}{x^2}\right) dx$ (метод трапецій);
8) $\int_5^7 \ln\frac{1}{x+1}\, dx$ (метод Сімпсона);
9) $\int_1^2 \cos\left(\frac{1}{x^2}\right) dx$ (метод середніх прямокутників);
10) $\int_2^3 \ln\frac{1}{\sqrt{x-1}}\, dx$ (метод прямокутників).

Питання для самоконтролю

1. Охарактеризуйте метод ділення навпіл.
2. Охарактеризуйте метод простих ітерацій.
3. Охарактеризуйте модифікований метод Ньютона.
4. Охарактеризуйте метод хорд.
5. Охарактеризуйте метод середніх прямокутників.
6. Охарактеризуйте метод прямокутників.
7. Охарактеризуйте метод трапецій.
8. Охарактеризуйте метод Сімпсона.

7. Лабораторна робота. Символьні перетворення у системі комп’ютерної алгебри SageMath

Мета: засвоїти можливості роботи системи комп’ютерної алгебри SageMath до символьних перетворень, таких як спрощення, розкладання, інтегрування, диференціювання.

Теоретичні відомості та методичні рекомендації

SageMath – це потужний приклад математичного програмного забезпечення загального призначення, який поєднує можливості багатьох окремих систем в одному середовищі.

Sage – це безкоштовне і вільно розповсюджуване математичне програмне забезпечення з відкритими вихідними кодом для дослідницької роботи та навчання у різних областях включаючи алгебру, геометрію, теорію чисел, криптографію, чисельні обчислення та багато іншого. Як модель розробки Sage, так і умови його розповсюдження та використання обрані відповідно до принципів відкритої та спільної роботи: ми збираємо машину, а не винаходимо колесо. Однією з основних цілей Sage є створення доступної, безкоштовної та відкритої альтернативи Maple, Matematica та MATLAB. Її головна ідея – об’єднати можливості багатьох уже існуючих математичних пакетів у єдиному зручному середовищі, зберігаючи безкоштовний і відкритий доступ.

У Sage є власне символьне ядро, проте Sage виступає переважно як інтегратор різних систем з єдиним графічним web-інтерфейсом – Jupyter Notebook (див. рис. 7.1).

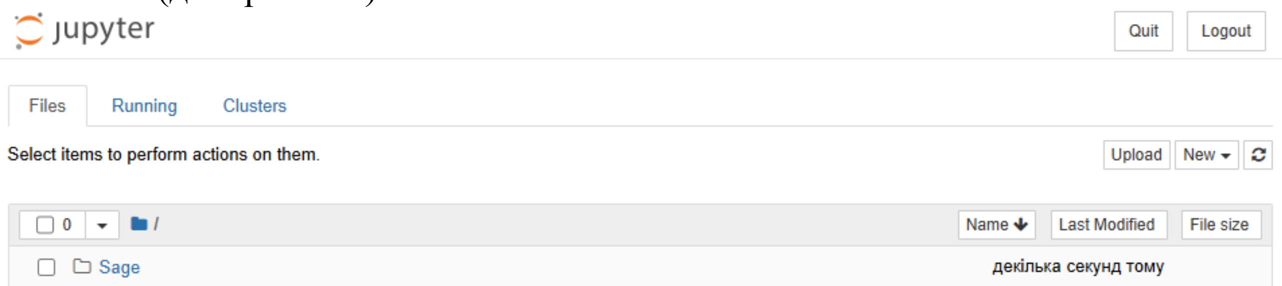

Рисунок 7.1 – Web-інтерфейс Jupyter Notebook

Основні характеристики:

1. відкритий код: SageMath поширюється під ліцензією GPL, тож кожен може вивчати, змінювати і поширювати її;
2. інтеграція з іншими системами – вона поєднує десятки бібліотек і пакетів, зокрема:
   - Maxima (символьна алгебра);
   - GAP (теорія груп);
   - PARI/GP (числова теорія);
   - SymPy (Python-бібліотека для символьної математики);
   - NumPy, SciPy, Matplotlib (наукові обчислення і графіка);
   - R (статистика);

3. мова програмування – використовує синтаксис Python, але з додатковими математичними можливостями;
4. підтримує різні типи обчислень:
 – символьні перетворення (спрощення, розкладання, інтегрування, диференціювання);
 – чисельні обчислення з довільною точністю;
 – розв’язування рівнянь і систем;
 – лінійну алгебру, аналіз, статистику;
 – теорію графів, комбінаторику;
 – диференціальні рівняння;
 – побудову 2D та 3D графіків.

У СКА SageMath є великий набір вбудованих числових і математичних констант, які можна використовувати без додаткового імпорту. Вони мають високу точність і можуть брати участь як у числових, так і в символьних обчисленнях.

У таблиці 7.1 наведено базові числові константи, що підтримуються системою SageMath:

Таблиця 7.1 – Базові числові константи у SageMath

Константа	Значення	Опис
`pi`	$\pi \approx 3.141592653$	Число π
`e`	$e \approx 2.718281828$	Число e. Основа натурального логарифма
`I` або `i`	$\sqrt{-1}$	Уявна одиниця
`oo`	∞	Нескінченність
`NaN`	$-$	«Not-a-Number», невизначене значення
`sqrt(2)`	≈ 1.414213562	Квадратний корінь з двох
`sqrt(3)`	≈ 1.732050807	Квадратний корінь з трьох
`log2`	≈ 0.6931471806	$\ln 2$
`log10`	≈ 2.302585093	$\ln 10$

У таблиці 7.2 наведено математичні та спеціальні константи, що підтримуються системою SageMath:

Таблиця 7.2 – Математичні та спеціальні константи у SageMath

Константа	Приблизне значення	Опис
`EULER_GAMMA` або `euler_gamma`	≈ 0.5772156649	Константа Ейлера–Маскероні
`golden_ratio` або `phi`	≈ 1.6180339887	Золоте відношення
`Catalan`	≈ 0.9159655941	Константа Каталана
`Khinchin`	≈ 2.6854520010	Константа Хінчина

`twinprime_constant`	≈ 0.6601618158	Константа подвійних простих чисел
`zeta(n)`		Функція Рімана, наприклад, `zeta(3)` ≈ 1.202056903
`degree`	$\frac{\pi}{180}$	Один градус у радіанах
`rad`	$\frac{180}{\pi}$	Один радіан у градусах

Усі ці константи можуть бути символьними або числовими з довільною точністю. Можна вказати точність обчислень, наприклад,

```
pi.n(100)  # 100 знаків після коми
```

У SageMath тригонометричні функції задаються так само, як у математиці, тільки в синтаксисі Python, причому вони можуть працювати і з числами, і з символьними виразами. У таблиці 7.3 наведено основні тригонометричні функції, що підтримуються системою SageMath:

Таблиця 7.3 – Основні тригонометричні функції у SageMath

Функція	Опис
`sin(x)`	синус
`cos(x)`	косинус
`tan(x)`	тангенс
`cot(x)`	котангенс
`sec(x)`	секанс
`csc(x)`	косеканс
`asin(x)`	арксинус
`acos(x)`	арккосинус
`atan(x)`	арктангенс
`acot(x)`	арккотангенс
`asec(x)`	арксеканс
`acsc(x)`	арккосеканс

У таблиці 7.4 наведено гіперболічні функції, що підтримуються системою SageMath:

Таблиця 7.4 – Гіперболічні функції у SageMath

Функція	Опис
`sinh(x)`	гіперболічний синус
`cosh(x)`	гіперболічний косинус
`tanh(x)`	гіперболічний тангенс
`coth(x)`	гіперболічний котангенс
`asinh(x)`	арксинус гіперболічний
`acosh(x)`	арккосинус гіперболічний

`atanh(x)`	арктангенс гіперболічний
`acoth(x)`	арккотангенс гіперболічний

У SageMath аргументи тригонометричних функцій за замовчуванням задаються в радіанах.

Арифметичні оператори у Sage:

1) + (додавання);
2) − (віднімання);
3) `*` (множення);
4) `/` (ділення);
5) `//` (ділення націло);
6) `%` (остача від ділення);
7) `^` (піднесення до степеню);
8) `**` (піднесення до степеню);
9) `-` (унарний мінус).

Приклад використання (див. рис. 7.2):

```
In [16]: sin(pi/2)+cos(pi)
Out[16]: 0

In [17]: tan(pi/4).n()
Out[17]: 1.00000000000000
```

Рисунок 7.2 – Тригонометричні обчислення у СКА SageMath

Оператор присвоєння: =

```
In [16]: a = 7
         b = 5

In [17]: a == b
Out[17]: False

In [18]: a != b
Out[18]: True

In [19]: a >= b
Out[19]: True
```

Рисунок 7.3 – Приклади порівняння

Оператори порівняння (див. рис. 7.3):

– == (рівність). Перевіряє, чи два значення рівні;
– != (нерівність). Перевіряє, чи два значення не рівні;

– <= (менше або дорівнює). Перевіряє, чи одне значення менше або дорівнює іншому;
– >= (більше або дорівнює). Перевіряє, чи одне значення більше або дорівнює іншому;
– < (менше). Перевіряє, чи одне значення менше за інше;
– > (більше). Перевіряє, чи одне значення більше за інше.

Часто використовується метод `n()` (аналог функція – `numerical_approx`). Цей метод приймає необов’язкові аргументи `prec`, що визначає кількість бітів точності, і `digits`, що визначає кількість десяткових цифр точності. За замовчуванням застосовується 53 біти точності (див. рис. 7.4).

```
In [1]: exp(1)
Out[1]: e

In [2]: n(exp(1))
Out[2]: 2.71828182845905

In [4]: exp(1).numerical_approx()
Out[4]: 2.71828182845905

In [5]: n(exp(1), digits=10)
Out[5]: 2.718281828

In [6]: n(exp(1), prec=100)
Out[6]: 2.7182818284590452353602874714
```

Рисунок 7.4 – Приклади обчислень з наперед заданою точністю

Якщо у виразі присутня більше ніж одна змінна, то Sage вимагає оголошення всіх їх символьними змінними – `var('x,y')` або `(x,y) = var('x,y')`.

У Sage є багато команд для перетворення алгебраїчних виразів, які дозволяють спрощувати, розкладати, розв’язувати та іншим чином маніпулювати виразами. Ось основні з них:
– функція `sympify()` спрощує вираз різними методами (без розкриття дужок);
– для розкриття дужок використовується функція `expand()`;
– функція `factor()` розкладає вираз на множники;
– функція `simplify_full()` виконує повне спрощення виразу з використанням кількох методів спрощення;
– функція `simplify_trig()` спрощує тригонометричні вирази;

- функція `simplify_rational()` спрощує раціональні вирази;
- функція `simplify_log()` спрощує логарифмічні вирази;
- функція `canonicalize_radical()` спрощує вирази з радикалами;
- функція `expand_trig()` розкриває дужки у тригонометричних виразах;
- функція `log_expand()` розкриває дужки у виразах, які містять логарифми;
- функція `partial_fraction()` розкладає раціональний дріб на прості дроби.

Ці команди дозволяють працювати з різними видами алгебраїчних виразів, спрощуючи, розкладаючи та маніпулюючи ними для досягнення потрібного результату, наприклад, див. рис. 7.5-7.6:

```
In [30]: expr8 = (x^2 - 1)^2/(x + 1)^4
         expr8.simplify_rational()

Out[30]: (x^2 - 2*x + 1)/(x^2 + 2*x + 1)
```

Рисунок 7.5 – Приклад спрощення раціональних виразів

```
In [45]: expr13 = 7/(x^3 - 1)
         expr13.partial_fraction()

Out[45]: -7/3*(x + 2)/(x^2 + x + 1) + 7/3/(x - 1)
```

Рисунок 7.6 – Приклад розкладання раціонального дробу на прості дроби

Завдання до лабораторної роботи

1. Обчислити у СКА SageMath з точністю до 15 знаків після коми:

11) $\dfrac{(7-6{,}35):6{,}5+9{,}9}{\left(1{,}2:36+1{,}2:0{,}25-1\frac{5}{16}\right):\frac{169}{24}}.$

12) $\left(\left(\dfrac{7}{9}-\dfrac{47}{72}\right):1{,}25+\dfrac{7}{40}\right):(0{,}358-0{,}108)\cdot 1{,}6-\dfrac{19}{25}.$

13) $\dfrac{\left(0{,}5:1{,}25+\frac{7}{5}:1\frac{4}{7}-\frac{3}{11}\right)\cdot 3}{\left(1{,}5+\frac{1}{4}\right):18\frac{1}{3}}.$

14) $\left(\dfrac{(2{,}7-0{,}8)\cdot 2\frac{1}{3}}{(5{,}2-1{,}4):\frac{3}{70}}+0{,}125\right):2\dfrac{1}{2}+0{,}43.$

15) $\dfrac{2\frac{3}{4}:1{,}1+3\frac{1}{3}}{2{,}5+0{,}4\cdot 3\frac{1}{3}}:\dfrac{5}{7}-\dfrac{\left(2\frac{1}{6}+4{,}5\right)\cdot 0{,}375}{2{,}75-1\frac{1}{2}}.$

16) $\dfrac{\left(13{,}75+9\frac{1}{6}\right)\cdot 1{,}2}{\left(10{,}3-8\frac{1}{2}\right)\cdot\frac{5}{9}}+\dfrac{\left(6{,}8-3\frac{3}{5}\right)\cdot 5\frac{5}{6}}{\left(3\frac{2}{3}-3\frac{1}{6}\right)\cdot 56}-27\dfrac{1}{6}.$

17) $\dfrac{\left(\frac{1}{6}+0{,}1+\frac{1}{15}\right):\left(\frac{1}{6}+0{,}1-\frac{1}{15}\right)\cdot 2{,}52}{\left(0{,}5-\frac{1}{3}+0{,}25-\frac{1}{5}\right):\left(0{,}25-\frac{1}{6}\right)\cdot\frac{7}{13}}.$

18) $\left(\dfrac{3\frac{1}{3}+2{,}5}{2{,}5-3\frac{1}{3}}\cdot\dfrac{4{,}6-2\frac{1}{3}}{4{,}6+2\frac{1}{3}}\cdot 5{,}2\right):\left(\dfrac{0{,}05}{\frac{1}{7}-0{,}125}+5{,}7\right).$

19) $\dfrac{0{,}4+8\left(5-0{,}8\cdot\frac{5}{8}\right)-5:2\frac{1}{2}}{\left(1\frac{7}{8}\cdot 8-\left(8{,}9-2{,}6:\frac{2}{3}\right)\right)\cdot 34\frac{2}{5}}\cdot 90.$

20) $\dfrac{\left(5\frac{4}{45}-4\frac{1}{6}\right):5\frac{8}{15}}{\left(4\frac{2}{3}+0{,}75\right)\cdot 3\frac{9}{13}}\cdot 34\frac{2}{7}+\dfrac{0{,}3:0{,}01}{70}+\dfrac{2}{7}.$

2. Розкласти дробово-раціональну функцію на суму елементарних дробів у СКА SageMath:

11) $\dfrac{x^2-5}{(x^2-4x+5)(x^2+9)}.$

12) $\dfrac{x+1}{x(x^4+6x^2+8)}.$

13) $\dfrac{1}{x^2(x^2+4)^2}.$

14) $\dfrac{3x^3+2x^2+1}{x(x+1)^2(x^2+4)}.$

15) $\dfrac{x^4-1}{(x^2+9)(x^3+x^2)}.$

16) $\dfrac{x^3+2x^2+3x+4}{x^4+x^3+2x^2}.$

17) $\dfrac{3x^2+4x-1}{(x^2+4x+29)^2}.$

18) $\dfrac{x^3-x-1}{x(x-1)^2(x^2+4)}.$

19) $\dfrac{x^4-1}{(x^2+16)(x^3+x^2)}.$

20) $\dfrac{5}{(x+4)(x^2+4x+20)}.$

3. Спростити вираз у СКА SageMath:

11) $\dfrac{\sqrt{x}+1}{x\sqrt{x}+x+\sqrt{x}}:\dfrac{1}{x^2-\sqrt{x}}.$

12) $\left(\left(\sqrt[4]{p}-\sqrt[4]{q}\right)^{-2}+\left(\sqrt[4]{p}+\sqrt[4]{q}\right)^{-2}\right):\dfrac{\sqrt{p}+\sqrt{q}}{p-q}.$

13) $\dfrac{2x^{-\frac{1}{3}}}{x^{\frac{2}{3}}-3x^{-\frac{1}{3}}}-\dfrac{x^{\frac{2}{3}}}{x^{\frac{5}{3}}-x^{\frac{2}{3}}}-\dfrac{x+1}{x^2-4x+3}.$

14) $\dfrac{\left(\sqrt{a}+\sqrt{b}\right)^2-4b}{(a-b):\left(\sqrt{\frac{1}{b}}+3\sqrt{\frac{1}{a}}\right)}:\dfrac{a+9b+6\sqrt{ab}}{\frac{1}{\sqrt{b}}+\frac{1}{\sqrt{a}}}.$

15) $x\cdot\dfrac{1+\frac{2}{\sqrt{x+4}}}{2-\sqrt{x+4}}+\sqrt{x+4}+\dfrac{4}{\sqrt{x+4}}.$

16) $\left(\dfrac{1+\sqrt{x}}{\sqrt{1+x}}-\dfrac{\sqrt{1+x}}{1+\sqrt{x}}\right)^2-\left(\dfrac{1-\sqrt{x}}{\sqrt{1+x}}-\dfrac{\sqrt{1+x}}{1-\sqrt{x}}\right)^2.$

17) $\left(\dfrac{1}{\sqrt{a}+\sqrt{a+1}}+\dfrac{1}{\sqrt{a}-\sqrt{a-1}}\right):\left(1+\sqrt{\dfrac{a+1}{a-1}}\right).$

18) $\frac{x-1}{x^{\frac{3}{4}}+x^{\frac{1}{2}}}\cdot\frac{x^{\frac{1}{2}}+x^{\frac{1}{4}}}{x^{\frac{1}{2}}+1}\cdot x^{\frac{1}{4}}+1.$

19) $\left(\frac{1+x+x^2}{2x+x^2}+2-\frac{1-x+x^2}{2x-x^2}\right)^{-1}\cdot(5-2x^2).$

20) $\sqrt{\frac{2a}{(1+a)\sqrt[3]{1+a}}}\cdot\sqrt[3]{\frac{4+\frac{8}{a}+\frac{4}{a^2}}{\sqrt{2}}}.$

Питання для самоконтролю

1. Охарактеризуйте систему комп'ютерної алгебри SageMath.
2. Які Ви знаєте базові числові константи у SageMath?
3. Які Ви знаєте математичні та спеціальні константи у SageMath?
4. З якої точністю можна проводити обчислювання у SageMath?
5. У яких одиницях беруться аргументи тригонометричних функцій за замовчуванням у SageMath?
6. Коли зручно використовувати функцію numerical_approx()?
7. Як оголошуються символьні змінні у SageMath?
8. Охарактеризуйте команди для перетворення алгебраїчних виразів, які дозволяють спрощувати, розкладати і розв'язувати у SageMath?

8. Лабораторна робота. Робота з графікою у системі комп’ютерної алгебри SageMath

Мета: засвоїти можливості роботи системи комп’ютерної алгебри SageMath до побудови графіків у декартовій, параметричній і полярній системі координат, а також поверхонь у просторі.

Теоретичні відомості та методичні рекомендації

У SageMath побудова графіків у декартовій, параметричній та полярній системах робиться різними функціями, але всі вони підтримують як числові, так і символьні вирази.

SageMath має дуже широкі можливості для побудови графіків – від простих 2D-кривих до складних 3D-поверхонь. Система інтегрує інструменти на основі Matplotlib, Jmol, Three.js та інших візуалізаційних бібліотек, тому підтримує і наукову, і навчальну графіку.

1. Декартова система координат. Використовується функція:

```
plot(f, (x, xmin, xmax), options...)
```

де `f` – вираз або функція від `x`; `(x, xmin, xmax)` – змінна і діапазон побудови (див. рис. 8.1).

```
x = var('x')
plot(sin(x), (x, -2*pi, 2*pi), color='blue', thickness=2)
```

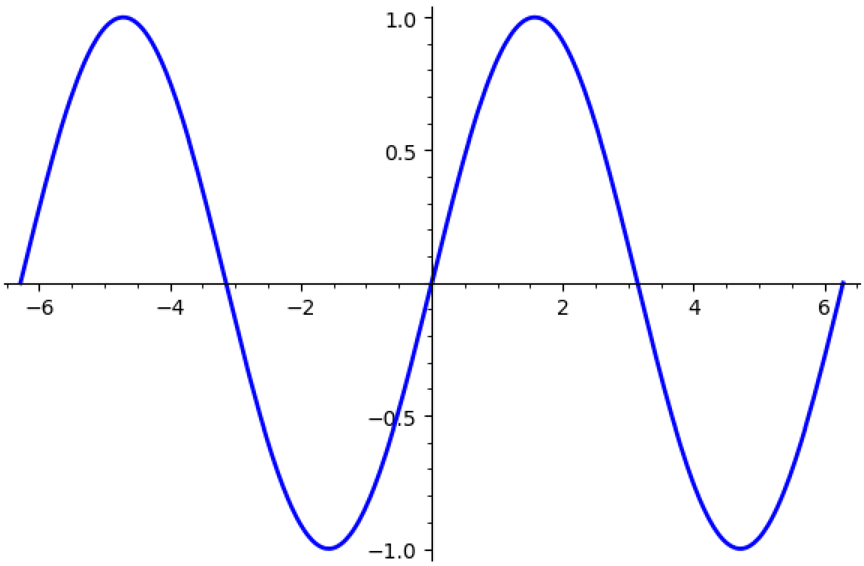

Рисунок 8.1 – Приклад 2D-графіки у декартовій СК

Можна будувати кілька функцій разом:

```
p1 = plot(sin(x), (x, -pi, pi), color='red')
p2 = plot(cos(x), (x, -pi, pi), color='green')
p1 + p2
```

Результат зображено на рис. 8.2:

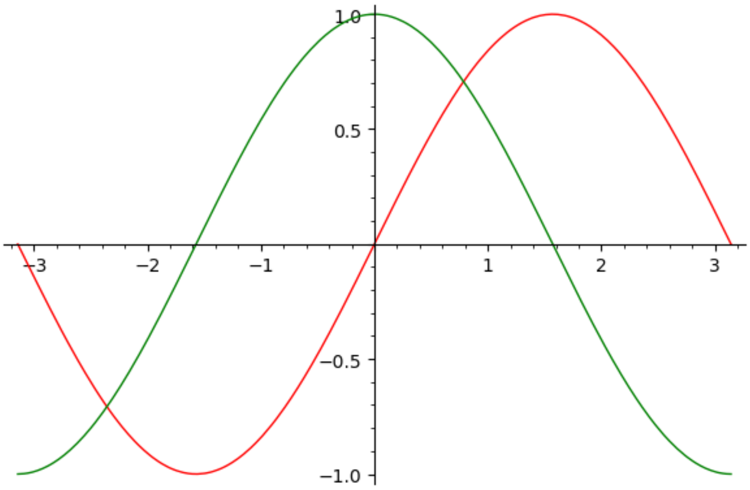

Рисунок 8.2 – Приклад двох графіків на одному рисунку

2. Параметрична система координат. Використовується функція:

```
parametric_plot((x_expr, y_expr), (t, tmin, tmax), options...)
```

де `(x_expr, y_expr)` – вирази для координат `X` і `Y` через параметр `t`; `(t, tmin, tmax)` – діапазон параметра (див. рис. 8.3).

```
t = var('t')
parametric_plot((2*(cos(t))^3, 2*(sin(t))^3), (t, 0, 2*pi), color='purple')
```

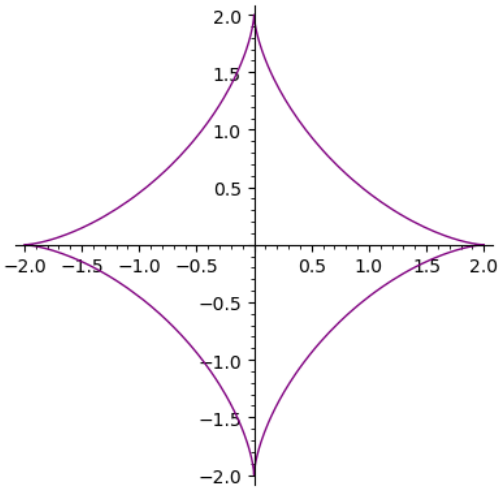

Рисунок 8.3 – Астроїда

Більш складна крива (спіраль Архімеда, див. рис. 8.4):

```
parametric_plot((t*cos(t), t*sin(t)), (t, 0, 6*pi), color='orange')
```

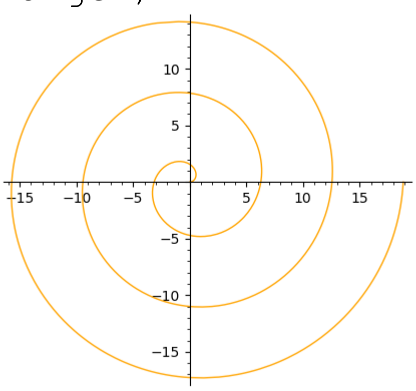

Рисунок 8.4 – Спіраль Архімеда

3. Полярна система координат. SageMath може двома способами побудувати лінію у полярній системі координат – полярну криву можна задати параметрично:

$$x = r(\theta)\cos\theta, y = r(\theta)\sin\theta,$$

або використати polar_plot з sage.plot.polar_plot (у старих версіях SageMath (≈ до 9.0). У новіших версіях функція polar_plot стала доступна напряму з головного простору імен SageMath (без імпорту з підмодулів).

Наприклад (див. рис. 8.5):

```
theta = var('theta')
r = 2*sin(4*theta)
parametric_plot((r*cos(theta), r*sin(theta)), (theta, 0, 2*pi), color='magenta')
```

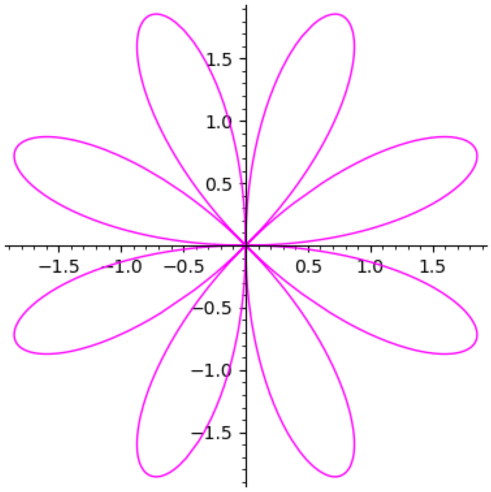

Рисунок 8.5 – Полярна роза

Або з вбудованим `polar_plot`. Результат буде той же, що і на рис. 8.5:

```
theta = var('theta')
polar_plot(2*sin(4*theta), (theta, 0, 2*pi), color='magenta')
```

Додаткові можливості графіків:

- товщина лінії: `thickness=2`;
- колір: `color='red'`;
- додати сітку: `gridlines=True`;
- підписати осі: `axes_labels=['x', 'y']`;
- експорт у файл: `.save('name.png')`.

4. Кілька графіків разом. Графіки можна додавати оператором +, щоб порівнювати різні функції (див. рис. 8.6):

```
p1 = plot(x^2, (x, -5, 5), color='red')
p2 = plot(-2*x+3, (x, -5, 5), color='green')
p1 + p2
```

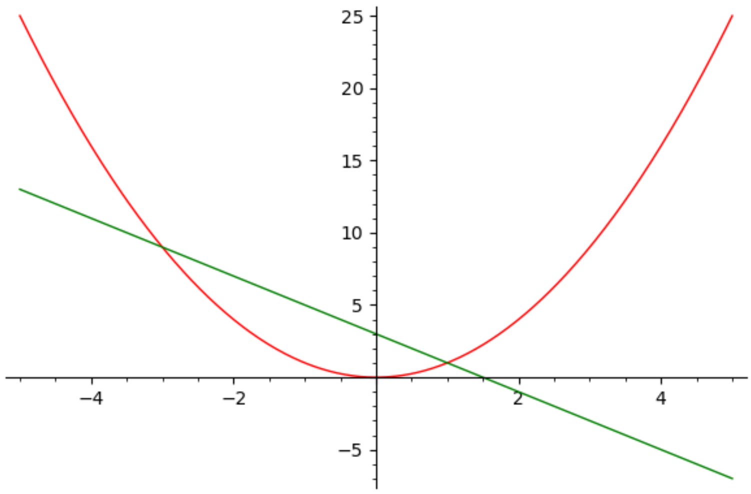

Рисунок 8.6 – Кілька графіків разом

5. Побудова поверхонь у 3D. SageMath підтримує як статичні, так і інтерактивні 3D-графіки. Функції двох змінних:

```
plot3d(f, (x, xmin, xmax), (y, ymin, ymax))
```

Наприклад, див. рис. 8.7

```
x, y = var('x y')
plot3d(sin(sqrt(x^2 + y^2))/(sqrt(x^2 + y^2)), (x, -5, 5),
(y, -5, 5))
```

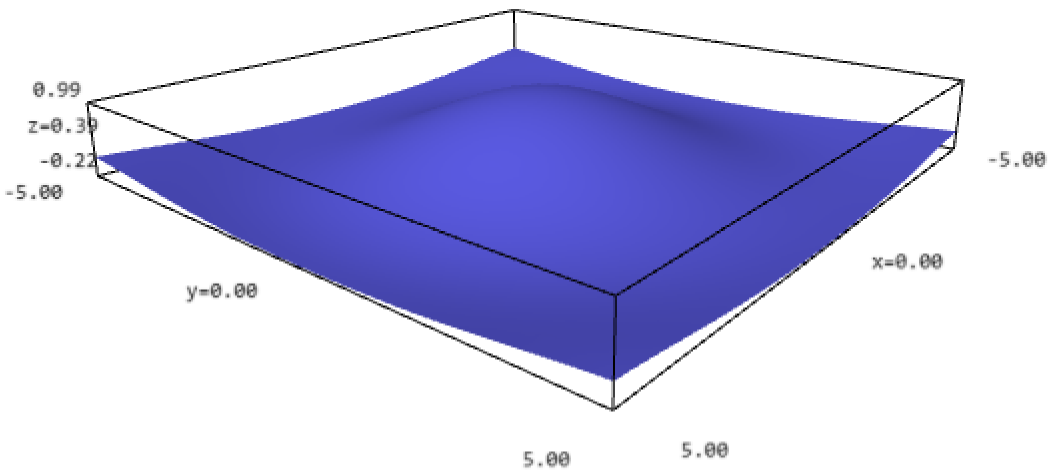

Рисунок 8.7 – Поверхня $\frac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}$

Параметричні поверхні:

```
parametric_plot3d((X(u,v), Y(u,v), Z(u,v)), (u, umin,
umax), (v, vmin, vmax))
```

Наприклад, див. рис. 8.8

```
u, v = var('u v')
parametric_plot3d((cos(u)*sin(v), sin(u)*sin(v), cos(v)),
(u, 0, 2*pi), (v, 0, pi))
```

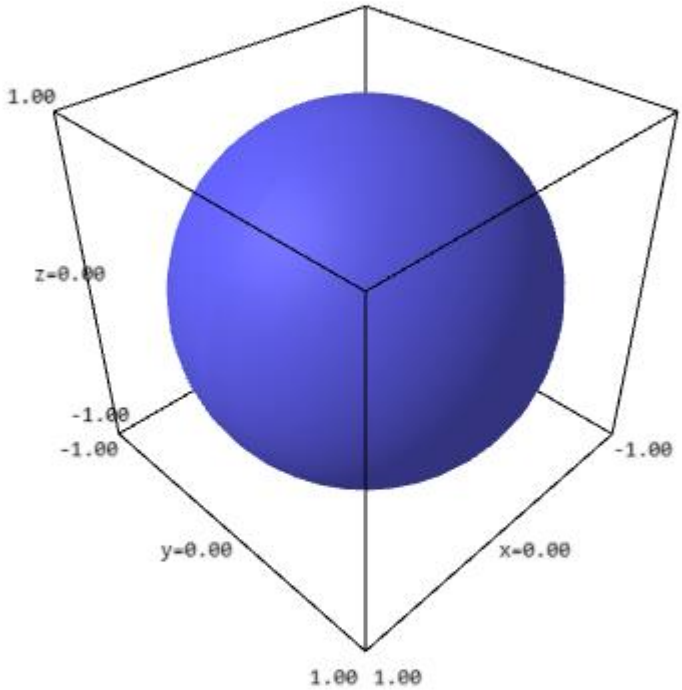

Рисунок 8.8 – Поверхня, яка задана параметрично

Полярні та циліндричні форми: можна використовувати параметричне задання через `r, theta, z`.

Отже, переваги побудови графіків у SageMath такі:

– єдиний синтаксис для 2D та 3D;
– можливість працювати одночасно з числовими та символьними виразами;
– підтримка складних параметричних і полярних форм;
– легка інтеграція з Python-бібліотеками для візуалізації (Matplotlib, Plotly).

Завдання до лабораторної роботи

1. Побудувати графіки функцій у декартовій системі координат у СКА SageMath (спочатку по одному, а потім два графіка на одному рисунку різного кольору):

1) a) $y = \frac{x^5}{(x^2-1)^2}$; b) $y = \sin x + \frac{1}{2}\sin 2\,x + \frac{1}{3}\sin 3\,x$.
2) a) $y = \sqrt[3]{x(3-x)^2} - x$; b) $y = \left(\frac{x+1}{x-1}\right)^4$.
3) a) $y = \frac{20x^2}{(x-1)^3}$; b) $y = \frac{\ln^2 x}{x}$.
4) a) $y = 8x^2(x^2-1)^3$; b) $y = \frac{x}{2} - \arccos\frac{2x}{1+x^2}$.
5) a) $y = \frac{x^5-8}{x^4}$; b) $y = \arccos\frac{1-x^2}{1+x^2}$.
6) a) $y = \frac{(x-5)^3}{(x-7)^2}$; b) $y = \frac{x}{\ln x}$.
7) a) $y = \frac{x^3-2x^2-x+2}{x}$; b) $y = \frac{x^2\sqrt{x^2-1}}{2x^2-1}$.

8) a) $y = \frac{x^3}{x^2-1}$; b) $y = (x^2-2)e^{-2x}$.

9) a) $y = \frac{1+x^2}{1+(x-2)^2}$; b) $y = \ln\left|\frac{1-x}{1+x}\right| + \frac{6}{1+x}$.

10) a) $y = \frac{x^2+2x-3}{x} - e^{\frac{1}{x}}$; b) $y = \frac{x}{2} + 2\operatorname{arctg} x$.

2. Побудувати графіки функцій, заданих у параметричній системі координат у СКА SageMath:

1) $x = \frac{9}{2}\cos^3 t, y = \frac{9}{4}\sin^3 t, t \in [0; 2\pi]$.
2) $x = t^3 - 2, y = t - 2, t \in [0; 10]$.
3) $x = 5\left(\cos t + \ln\left(\operatorname{tg}\frac{t}{2}\right)\right), y = 5\sin t, t \in [0; 2\pi]$.
4) $x = 2(t - \sin t), y = 2(1 - \cos t), t \in [0; 2\pi]$.
5) $x = t - \frac{\operatorname{sh} 2t}{2}, y = 2\operatorname{ch} t, t \in [0; 15]$.
6) $x = (t^2-2)\sin t + 2t\cos t, y = (t^2-2)\cos t - 2t\sin t, t \in [0; \pi]$.
7) $x = 3(\operatorname{sh} t - t), y = 3(\operatorname{ch} t - 1), t \in [0; 21]$.
8) $x = \cos^3 t, y = \sin^3 t, t \in \left[0; \frac{\pi}{2}\right]$.
9) $x = 2t^3(1-t^2), y = t\sqrt{7}, t \in [0; 12]$.
10) $x = 2t^2, y = t\left(\frac{1}{4} - t^2\right), t \in [0; 20]$.

3. Побудувати графіки функцій, заданих у полярній системі координат у СКА SageMath:

1) $\rho = 5\sin^2 2\varphi$.
2) $\rho = 7\cos^2 \varphi$.
3) $\rho = 2(1 + \sin^2 \varphi)$.
4) $\rho = 4\sin 2\varphi$.
5) $\rho = 9\cos 5\varphi$.
6) $\rho = 4(1 + \cos\varphi)$.
7) $\rho = 3\cos 3\varphi$.
8) $\rho = 8\cos 4\varphi$.
9) $\rho = 7\sin 5\varphi$.
10) $\rho = 2\sin 4\varphi$.

4. Побудувати поверхні, використовуючи основні опції функції `plot3d()` у декартовій системі координат у СКА SageMath:

1) $z = 2x^2 + 9y^2$.
2) $z = 3 - (x^2 + y^2)$.
3) $z = 4 - x^2$.
4) $z = \frac{y^2}{5}$.
5) $z = 2x^2 - 3y^2$.
6) $z = 12 + x^2 + 2y^2$.
7) $z = 2x^2 - 3$.
8) $z = 4x^2 + \frac{y^2}{3} + 1$.
9) $z = \frac{x^2}{3} - 6$.

10) $z = 6x^2 - y^2$.

5. Побудувати поверхні, задані параметрично, у СКА SageMath:

1) Площина $\begin{cases} x = u, \\ y = v, \\ z = 2u + 3v. \end{cases}$, де $u, v \in \mathbb{R}$.

2) Параболоїд обертання $\begin{cases} x = u\cos v\,, \\ y = u\sin v\,, \\ z = u^2. \end{cases}$, де $u \geq 0, v \in [0; 2\pi]$.

3) Однопорожнинний гіперболоїд $\begin{cases} x = \cosh v \cos v\,, \\ y = \cosh v \sin v\,, \\ z = \sinh u\,. \end{cases}$, де $u \in \mathbb{R}, v \in [0; 2\pi]$.

4) Тор $\begin{cases} x = (7 + 3\cos u)\cos v\,, \\ y = (7 + 3\cos u)\sin v\,, \\ z = r\sin u\,. \end{cases}$, де $u, v \in [0; 2\pi]$.

5) Сфера $\begin{cases} x = 2\sin u \cos v\,, \\ y = 2\sin u \sin v\,, \\ z = 2\cos u\,. \end{cases}$, де $u, v \in [0; 2\pi]$.

6) Мобіусова смуга $\begin{cases} x = \left(1 + \frac{v}{2}\cos\frac{u}{2}\right)\cos u\,, \\ y = \left(1 + \frac{v}{2}\cos\frac{u}{2}\right)\sin u\,, \\ z = \frac{v}{2}\sin\frac{u}{2}. \end{cases}$, де $u \in [0; 2\pi], v \in [-1; 1]$.

7) Конус $\begin{cases} x = u\cos v\,, \\ y = u\sin v\,, \\ z = 7u. \end{cases}$, де $u \geq 0, v \in [0; 2\pi]$.

8) Циліндр $\begin{cases} x = 4\cos u\,, \\ y = 4\sin u\,, \\ z = v. \end{cases}$, де $u \in [0; 2\pi], v \in \mathbb{R}$.

9) Гіперболічний параболоїд $\begin{cases} x = u, \\ y = v, \\ z = u^2 - v^2. \end{cases}$, де $u, v \in \mathbb{R}$.

10) Поверхня синусоїдального гребеня $\begin{cases} x = u, \\ y = v, \\ z = \sin u \cdot \cos v\,. \end{cases}$, де $u, v \in \mathbb{R}$.

Питання для самоконтролю

1. Охарактеризуйте можливості побудови графіків у СКА SageMath.
2. Розкажіть про синтаксис функції plot().
3. Розкажіть про синтаксис функції parametric_plot().
4. Розкажіть про синтаксис функції polar_plot().
5. Розкажіть про синтаксис функції plot3d().
6. Розкажіть про синтаксис функції parametric_plot3d().
7. Як змінити підписи осей на графіку?
8. Як можна покращити оформлення графіків у СКА SageMath?
9. Чи можна будувати кілька графіків одночасно? Якщо так, як?

9. Лабораторна робота. Чисельні обчислення у системі комп'ютерної алгебри Scilab

Мета: ознайомлення з використанням можливостей роботи СКА Scilab для чисельних обчислень.

Теоретичні відомості та методичні рекомендації

Scilab – це вільно поширювана система комп'ютерної математики (computer algebra system, CAS) та технічних обчислень, яку можна розглядати як аналог MATLAB із відкритим кодом. Інтерфейс Scilab має доволі простий вигляд. Вікно містить меню, панель інструментів і консоль з знаком запрошення --> (див. рис. 9.1). Ввід команд у Scilab виконують з клавіатури. Натиснення клавіші Enter заставляє систему виконати команду і вивести результат. Вона орієнтована насамперед на чисельні обчислення, але має й певні можливості для символьної математики через підключення зовнішніх пакетів (наприклад, SciMax чи інтерфейс до Maxima). Файл зберігається з розширенням «sod». Слід зазначити, що можливості збереження всього тексту сесії команда «Зберегти середовище» не дає.

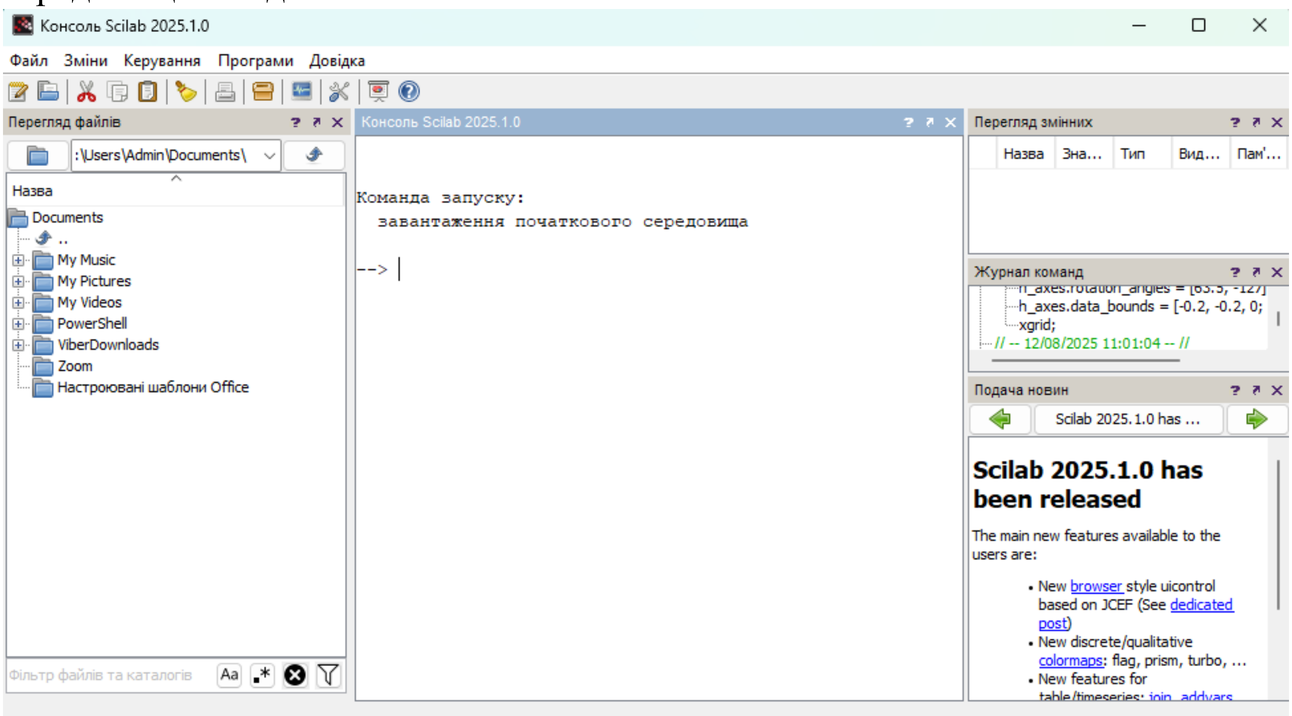

Рисунок 9.1 – Інтерфейс Scilab

Основні можливості СКА Scilab:

1. чисельні обчислення:

– лінійна алгебра (матриці, вектори, розв'язання систем рівнянь, власні значення/вектори);
– нелінійна оптимізація та рівняння;
– диференціальні рівняння;

– статистичні обчислення;
– обробка сигналів та зображень;

2. графіка та візуалізація:

– 2D та 3D графіки;
– анімації та інтерактивні побудови;
– візуалізація даних із файлів і сенсорів;

3. моделювання та симуляція

– Xcos (див. рис. 9.2) – аналог Simulink, дозволяє створювати блок-схеми для моделювання динамічних систем;
– підтримка моделювання у реальному часі (через спеціальні модулі);

4 символьна математика:

– базово відсутня (Scilab – це більше про чисельні методи), але можна підключити: SciMax – модуль, який інтегрує систему Maxima.

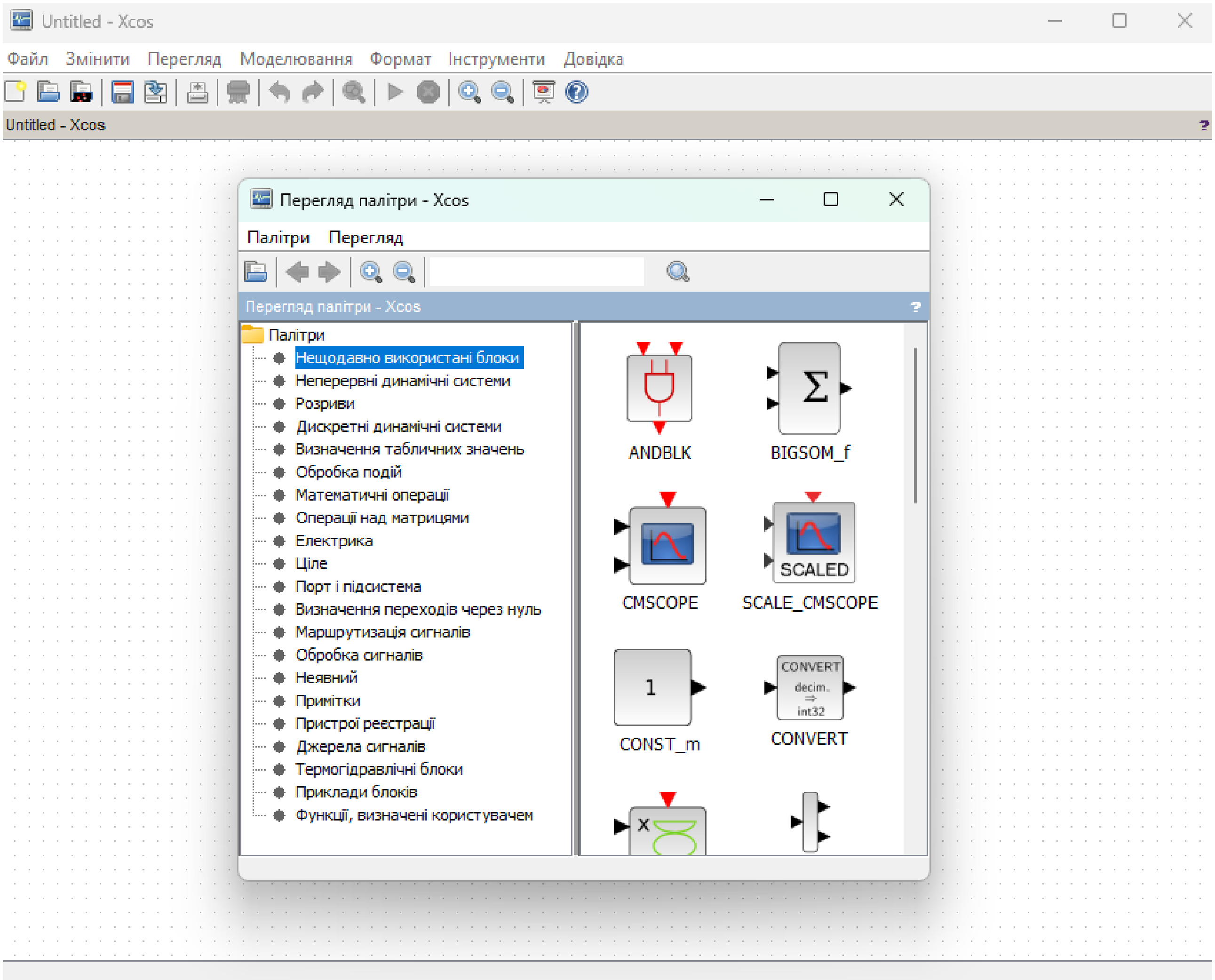

Рисунок 9.2 – Інтерфейс Xcos

У Scilab введення команд має кілька особливостей, які відрізняють його від інших систем комп’ютерної математики:

– команди можна вводити пакетом (декілька команд в одному рядку), розділяючи їх комою «,» або крапкою з комою «;»;

– якщо команда закінчується «;», то результат не виводиться;
– якщо «;» немає, результат відображається у консолі;
– однорядковий коментар починається з //;
– багаторядкові коментарі пишуться між /* та */;
– для визначення змінної треба набрати ім’я змінної, знак «=» і значення змінної. Для очистки значення змінної можна виконати команду `clear name`, де `name` – ім’я змінної.

Наприклад, див. рис. 9.3:

```
--> // Кілька команд в одному рядку

--> x = 5; y = x^2, z = y + 10

 y =

   25.

 z =

   35.

--> |
```

Рисунок 9.3 – Виконання команд пакетом

У Scilab є набір вбудованих системних констант (починаються з символу `%`), які можна використовувати без додаткового визначення. Вони стосуються як математичних значень, так і системних параметрів середовища і наведені у таблиці 9.1.

Таблиця 9.1 – Вбудовані системні константи

Константа	Значення	Призначення
`%pi`	π	Число пі
`%e`	e	Основа натуральних логарифмів
`%i`	$\sqrt{-1}$	Уявна одиниця
`%nan`	Not-a-Number	Результат некоректних обчислень, наприклад, $\frac{0}{0}, \frac{\infty}{\infty}$
`%inf`	∞	Нескінченність
`%eps`	$2.220D-16$	Машинний епсилон або умовний нуль (межа точності `double`)
`%t`	true	Логічне «істина»
`%f`	false	Логічне «хибний»

Наприклад, див. рис. 9.4:

```
--> %t
 ans =
  T
--> %eps
 %eps =
   2.220D-16
  "Environment saved."
--> %inf
 %inf =
   Inf
```

Рисунок 9.4 – Приклади деяких вбудованих системних констант

У таблиці 9.2 наведено елементарні математичні функції, що підтримуються системою Scilab:

Таблиця 9.2 – Елементарні математичні функції у Scilab

Функція	Опис
`sin(x)`	Синус
`cos(x)`	Косинус
`tan(x)`	Тангенс
`cotg(x)`	Котангенс
`asin(x)`	Арксинус
`acos(x)`	Арккосинус
`atan(x)`	Арктангенс
`exp(x)`	Експонента e^x
`sqrt(x)`	Корінь квадратний з x
`round(x)`	Округлення
`log(x)`	Натуральний логарифм з числа x
`log10(x)`	Десятковий логарифм з числа x
`log2(x)`	Логарифм за основою 2 від числа x
`abs(x)`	Модуль числа x
`sinh(x)`	Синус гіперболічний
`cosh(x)`	Косинус гіперболічний
`tanh(x)`	Тангенс гіперболічний

Наприклад, див. рис. 9.5:

```
--> x = %pi/3;

--> y1 = sin(x) + cos(x);

--> y2 = log10(100) + sqrt(16);

--> disp("sin+cos:", y1);

  "sin+cos:"

   1.3660254

--> disp("log10+sqrt:", y2);

  "log10+sqrt:"

   6.

--> |
```

Рисунок 9.5 – Приклади з функціями

У Scilab функції, визначені користувачем (user-defined functions) – це спосіб створити власні команди, які можна багаторазово викликати з різними аргументами. Вони дуже схожі на функції у MATLAB або у мовах програмування, але мають свої особливості синтаксису.

Загальний синтаксис:

```
function [вихідні_змінні] = ім’я_функції(вхідні_змінні)
    // тіло функції
endfunction
```

- вхідні_змінні – список параметрів, які передає користувач;
- вихідні_змінні – значення, які функція повертає (може бути одне або кілька);
- імена функцій не повинні збігатися з іменами вбудованих команд;
- можна створювати анонімні функції (без збереження в окремому файлі) або зберігати їх у файлах з розширенням .sci.

Розглянемо приклади.

Функція з одним аргументом і одним результатом:

```
function y = square(x)
    y = x^2;
endfunction

// Виклик функції
res = square(5)
```

Результат див. рис. 9.6:

```
--> res = square(5)

 res =                    res =

  25.                       25.
```

Рисунок 9.6 – Результат роботи функції з одним аргументом

Функція з кількома аргументами і кількома результатами:

```
function [summa, diff] = sum_and_diff(a, b)
    summa = a + b;
    diff  = a - b;
endfunction

[s, d] = sum_and_diff(7, 3)
```

Результат див. рис. 9.10:

```
--> [s, d] = sum_and_diff(7, 3)

 s =

   10.

 d =

   4.
```

Рисунок 9.10 – Результат роботи функції з кількома аргументами

Функція без аргументів:

```
function msg = hello()
    msg = "Привіт, Scilab!";
endfunction

disp(hello());
```

Результат див. рис. 9.11:

```
--> disp(hello());

  "Привіт, Scilab!"
```

Рисунок 9.11 – Результат роботи функції без аргументів

Функція для обчислення факторіала:

```
function f = factorial_sc(n)
    f = 1;
    for k = 1:n
        f = f * k;
    end
endfunction

disp(factorial_sc(5))
```

Результат див. рис. 9.12:

```
--> disp(factorial_sc(5))

    120.
```

Рисунок 9.12 – Результат роботи функції для обчислення факторіала

У Scilab команда `deff` – це скорочений спосіб швидко створити функцію, визначену користувачем, без окремого файлу і без повного синтаксису. Загальний синтаксис:

```
deff('вихідні_змінні = ім’я_функції(вхідні_змінні)',
                    'тіло_функції')
```

- перший аргумент – рядок, який містить заголовок функції;
- другий аргумент – рядок з виразом або кількома командами тіла функції;
- якщо тіло функції складається з кількох команд, їх розділяють крапкою з комою «;».

Наприклад, написати функцію, за допомогою якої можна обчислити площу трикутника за формулою Герона:

```
deff('S=G(a,b,c)','p=(a+b+c)/2;S=sqrt(p*(p-a)*(p-b)*(p-c))');
```

Результат див. рис. 9.13:

```
--> G(3,4,5)

 ans =

   6.
```

Рисунок 9.13 – Результат роботи функції для обчислення площі трикутника за формуло. Герона

Запам’ятайте, що `deff` визначає анонімну функцію, яка зберігається в пам’яті на час роботи сесії.

У Scilab для розв’язання алгебраїчного рівняння існує команда:

```
r = roots(p)
```

де `p` – вектор коефіцієнтів многочлена, починаючи з найстаршого степеня, `r` – вектор усіх коренів (дійсних і комплексних).

Наприклад, знайти корні рівняння $x^3 - 2x - 7 = 0$. Коефіцієнти многочлена: $[1,0,-2,-7]$, тоді див. рис. 9.14:

```
--> p = [1 0 -2 -7];

--> r = roots(p)

 r = [3x1 double]

   2.2582589 + 0.i
  -1.1291294 + 1.3508515i
  -1.1291294 - 1.3508515i
```

Рисунок 9.14 – Знаходження коренів алгебраїчного рівняння

Але команда `roots(p)` не підходить для трансцендентних і нелінійних рівнянь та їх систем. У цьому випадку використовується команда:

```
x = fsolve(x0, f)
```

де `x0` – початкове наближення кореня (може бути число або вектор), `f` – функція, яка повертає значення рівняння $f(x) = 0$, `x` – знайдений корінь.

Наприклад, розв'яжемо систему:

$$\begin{cases} x^2 + y^2 - 4 = 0, \\ e^x + y - 1 = 0. \end{cases}$$

```
deff('F = sys(X)', 'F(1) = X(1)^2 + X(2)^2 - 4; F(2) =
exp(X(1)) + X(2) - 1')

// Початкове наближення
X0 = [1; 1];

// Виклик fsolve
sol = fsolve(X0, sys)
disp(sol)
```

Результат див. рис. 9.15:

```
--> disp(sol)

  -1.8162641
   0.8373678
```

Рисунок 9.15 – Знаходження розв'язку системи

Початкове наближення критично важливе – від нього залежить, чи знайде метод корінь і який саме (особливо, якщо коренів кілька). Якщо функція погано поводиться, можна:

– побудувати графік, щоб оцінити, де знаходиться корінь;
– використати кілька різних початкових наближень.

У Scilab чисельне інтегрування використовується, коли неможливо (або складно) знайти інтеграл аналітично. Система має кілька вбудованих функцій для цього, головна з яких:

```
integrate('вираз', 'змінна', нижня_межа, верхня_межа)
```

Якщо Scilab може знайти аналітичний результат – отримаєте точний інтеграл. Якщо ні – використає чисельний метод. Межі можна задавати числами або `%inf` / `-%inf`. Наприклад:

```
integrate('sin(x)', 'x', 0, %pi)
```

Результат див. рис. 9.16:

```
 ans =

   2.
```

Рисунок 9.16 – Результат знаходження визначеного інтеграла

У Scilab `inttrap` – це функція чисельного інтегрування методом трапецій. Вона підходить, коли ми вже маємо табличні значення функції (тобто не формулу, а масив точок). Той же самий інтеграл $\int_0^{\pi} \sin x\, dx$ розв'яжемо чисельним методом трапецій:

```
// Масив точок
x = 0:%pi/10:%pi;
y = sin(x);

// Чисельне інтегрування методом трапецій
S = inttrap(x, y);
disp(S);
```

Результат див. рис. 9.17:

```
--> disp(S);

   1.9835235
```

Рисунок 9.17 – Результат знаходження визначеного інтеграла методом трапецій

Ще є функція `intg()`, яка виконує лише чисельний розрахунок.

Отже, СКА Scilab має такі переваги та недоліки.

Плюси:

– безкоштовність та відкритий код;
– добра підтримка матричних і чисельних методів;
– Xcos для моделювання без програмування;
– кросплатформеність.

Мінуси:

– менша популярність порівняно з MATLAB – менше прикладів і готових бібліотек;
– символьні обчислення потребують окремого встановлення;
– інтерфейс іноді менш зручний, ніж у комерційних аналогів.

Завдання до лабораторної роботи

1. За допомогою оператора `deff` створити функцію і обчислити її значення у заданій точці:

1) $f(x) = e^{\frac{x}{y}} \cdot \sqrt{\left|\frac{-x^2-7y-120}{\ln 3x}\right|}, x = 5, y = 7.$

2) $f(x) = \sin(xy) \cdot \log_2(x^2 + y^2 + 5), x = -2, y = -1.$

3) $f(x) = \mathrm{ch}(x - y) \cdot \sqrt{\left|\ln \frac{x^3-5}{y^4}\right|}, x = 1, y = 8.$

4) $f(x) = e^{x^2+y^8} \cdot \mathrm{tg}\left(\cos\left(\frac{x^3+2x-8}{141}\right)\right), x = -3, y = 6.$

5) $f(x) = \cos\left(\frac{x-y}{10}\right) \cdot \ln\left|\ln \frac{x^4-1}{y^4+1}\right|, x = -5, y = -1.$

6) $f(x) = \mathrm{ctg}\left(\left|\frac{x-y}{xy}\right|\right) \cdot e^{x^{12}+x^6+3}, x = -6, y = 6.$

7) $f(x) = \operatorname{sh}\left(\frac{x^3}{x+y}\right) \cdot \log_2(10x^2 + 10y^2)\,, x = 2, y = 9.$

8) $f(x) = \operatorname{ch}(x^3 - y^3) \cdot \sqrt{\left|\ln \frac{x+y}{x^2+y^2}\right|}\,, x = -1, y = \frac{1}{2}.$

9) $f(x) = e^{-x^2 - xy^3} \cdot \operatorname{tg}\left(\sin\left(\frac{x^3+3x}{\sqrt{x+y}}\right)\right), x = 3, y = 11.$

10) $f(x) = \operatorname{ctg}\left(\left|\frac{e^{xy}}{xy}\right|\right) \cdot e^{x^2 - 9x^6 - 5}, x = -5, y = 5.$

2. Обрати з геометрії або стереометрії довільну формулу і написати функцію, яка реалізує обчислення за обраною формулою.

3. Розв'язати рівняння СКА Scilab:

1) $e^x - 3x = 0.$
2) $\sin x - \frac{x^2}{4} = 0.$
3) $\ln x + x^2 - 3 = 0.$
4) $\operatorname{tg} x - x^3 = 0.$
5) $\cos x - |x| = 0.$
6) $e^{-x} - \cos x = 0.$
7) $\ln x - \sin x = 0, x > 0.$
8) $e^{x^2} - 5 = 0.$
9) $\sin^2 x - 0.5x = 0.$
10) $xe^x - 4 = 0.$

4. Обчислити чисельно визначений інтеграл, використовуючи функції `integrate`, `intg` і `inttrap`. Порівняйте отримані результати:

1) a) $\int_0^{\frac{\pi}{2}} x \cos x \, dx$; b) $\int_1^{+\infty} \frac{dx}{\sqrt{x}}$.

2) a) $\int_{\frac{\pi}{6}}^{\frac{\pi}{4}} \frac{\cos x}{x} dx$; b) $\int_0^{+\infty} \frac{\sin x \cos x dx}{x}$.

3) a) $\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{dx}{1+\cos x}$; b) $\int_1^{+\infty} \frac{dx}{1+x^2}$.

4) a) $\int_0^{\frac{\pi}{2}} x^3 \cos x \, dx$; b) $\int_0^{+\infty} \frac{1-e^{-2x^2}}{xe^{x^2}} dx$.

5) a) $\int_0^{\frac{\pi}{2}} \frac{dx}{5 - 4\sin x + 3\cos x}$; b) $\int_1^{+\infty} \frac{\operatorname{arctg} 4x}{x^2} dx$.

6) a) $\int_0^{\frac{\pi}{2}} \frac{\sin x \cos x}{(\cos^2 x + \sin^2 x)^2} dx$; b) $\int_1^{+\infty} \frac{\ln(x^2+1)}{x} dx$.

7) a) $\int_0^{\pi} x^3 \sin x \, dx$; b) $\int_0^{+\infty} e^{-2x} dx$.

8) a) $\int_0^{e} \ln^3 x \, dx$; b) $\int_0^{+\infty} \frac{dx}{x^2+4}$.

9) a) $\int_4^{9} \frac{x-1}{\sqrt{x}+1} dx$; b) $\int_0^{+\infty} e^{-x}\sqrt{x} dx$.

10) a) $\int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \frac{x}{\sin^2 x} dx$; b) $\int_1^{+\infty} \frac{x^3+1}{x^4} dx$.

Питання для самоконтролю

1. Охарактеризуйте можливості системи комп’ютерної алгебри Scilab.
2. Чи можна виконувати у Scilab символьні обчислення?
3. Як задаються вбудовані системні константи у Scilab?
4. Наведіть приклади синтаксису елементарних математичних функцій у Scilab.
5. Для чого потрібен Xcos?
6. Розкажіть про особливості введення команд у Scilab.
7. Охарактеризуйте різні способи створення функції у Scilab.
8. Яка команда підходить для розв’язання трансцендентних і нелінійних рівнянь та їх систем?
9. Які функції Ви знаєте для чисельного інтегрування у Scilab?

10. Лабораторна робота. Графіка у системі комп'ютерної алгебри Scilab

Мета: засвоїти можливості роботи з графікою у Scilab до розв'язування задач візуалізації функцій.

Теоретичні відомості та методичні рекомендації

Scilab має досить розвинені засоби для побудови графіків функцій, кривих і поверхонь, що дозволяють робити як швидкі наочні візуалізації, так і досить складну графіку для наукових публікацій.

1. Графіки функції однієї змінної можна побудувати командою

`plot(x, y),`

де `x` – масив абсцис, `y` – функція. Наприклад, побудуємо функцію $y = \cos(\sin x), x \in [0; 2\pi]$.

```
x = 0:0.1:2*%pi;
y = cos(sin(x));
plot(x, y);

xlabel("X");
ylabel("Y");
title("function y = cos(sin(x))");
```

Результат див. рис. 10.1:

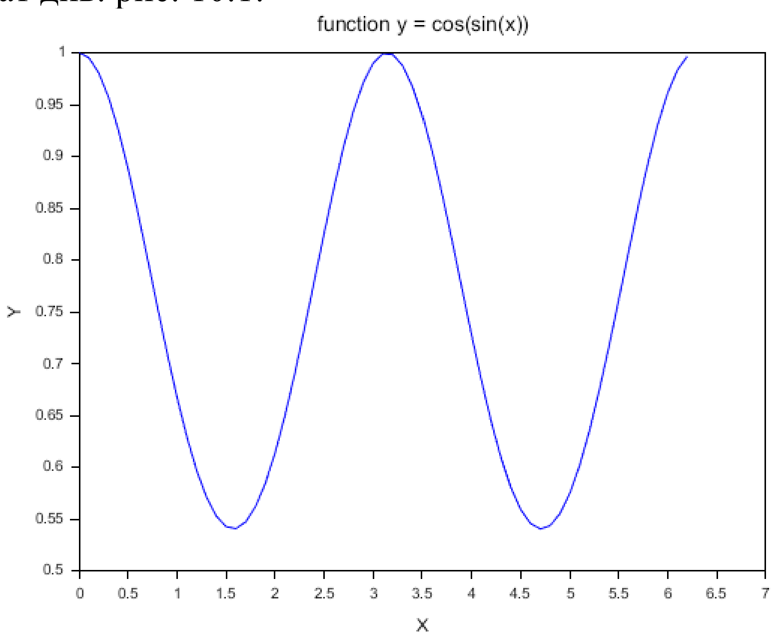

Рисунок 10.1 – Результат роботи функції `plot(x, y)`

Різниця з MATLAB у тому, що Scilab розділяє побудову графіка та його оформлення. Тому, все, що стосується підписів та заголовків, задається окремими командами (`xlabel, ylabel, title, legend`).

У Scilab є ще одна команда `plot2d()`, яка теж малює двовимірні графіки. Обидві команди `plot()` і `plot2d()` малюють 2D-графіки, але вони з'явилися в різний час і працюють трохи по-різному. `Plot()` сучасніша і простіша команда, яка була створена для зручності, схожа на команду у MATLAB. Автоматично визначає масштаб і формат. Підходить для швидкого виводу кривих без детального налаштування. `Plot2d()` більш «старий» та гнучкий інструмент (залишився з ранніх версій Scilab). Має багато параметрів для керування масштабом, маркерами, осями. Використовується, коли потрібно точно налаштувати вигляд графіка в момент побудови, а не окремими командами `xlabel`, `title` тощо:

```
x = 0:0.1:2*%pi;
plot2d(x, cos(sin(x)), style=color("green"))
```

Результат див. рис. 10.2:

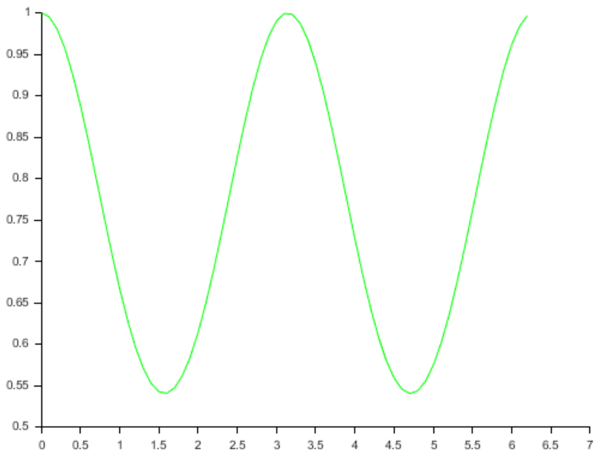

Рисунок 10.2 – Результат роботи функції `plot2d()`

2. Графік у параметричній формі:

```
t = 0:0.1:10*%pi;
x = t .* cos(t);
y = t .* sin(t);
plot(x, y);
```

Результат див. рис. 10.3:

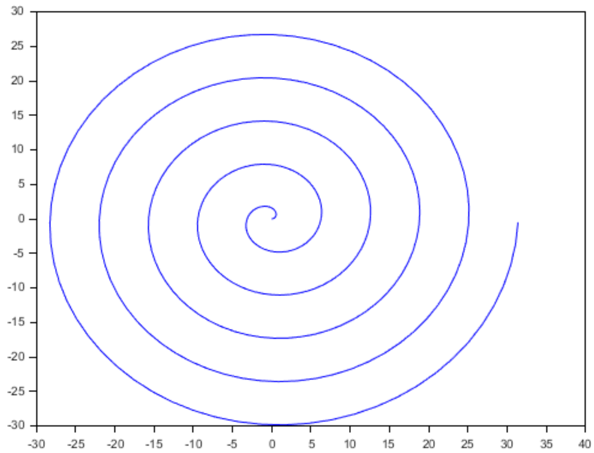

Рисунок 10.3 – Графік у параметричній формі

3. Графік у полярних координатах будується за допомогою функції `polarplot(theta, r)`.

```
theta = 0:0.01:2*%pi;
r = 1 + sin(theta);
polarplot(theta, r);
```

Результат див. рис. 10.4:

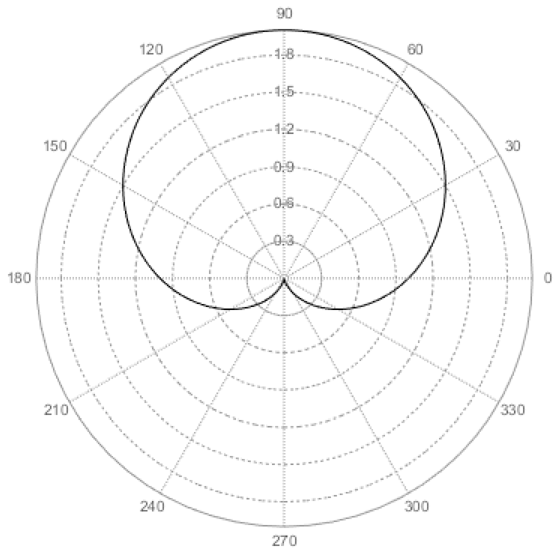

Рисунок 10.4 – Графік у полярних координатах

4. Поверхні та 3D графіки. У Scilab є команда відображення функції $z = f(x, y)$ у вигляді сітки або поверхні:

```
plot3d(x, y, z)
```

Наприклад:

```
x = -2:0.2:2;
y = -2:0.2:2;
[X, Y] = ndgrid(x, y);
Z = sin(X) .* cos(Y);
plot3d(x, y, Z);
```

Результат див. рис. 10.5:

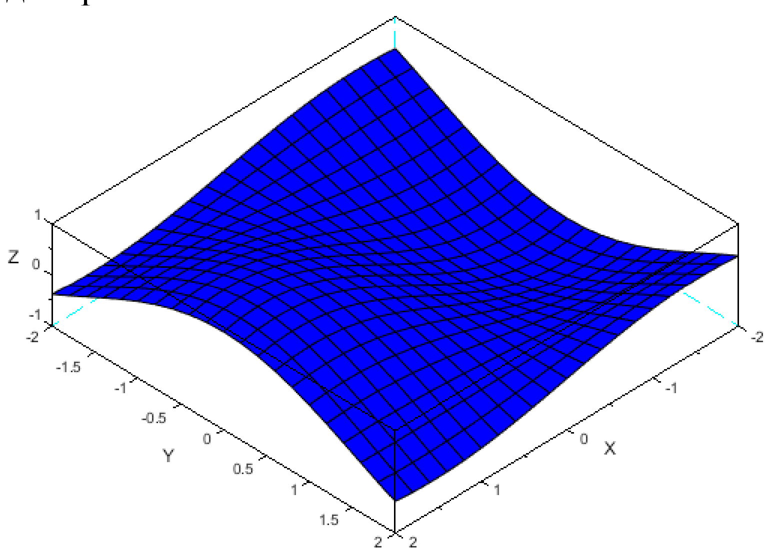

Рисунок 10.5 – Результат роботи функції `plot3d()`

Є ще функції для 3D графіки, такі як: `surf(x, y, z)` – гладка заливка поверхні, `mesh(x, y, z)` – каркасна модель без заливки.

```
x = -2:0.2:2;
y = -2:0.2:2;
[X, Y] = ndgrid(x, y);
Z = sin(X) .* cos(Y);
surf(x, y, Z);
```

Результат див. рис. 10.6:

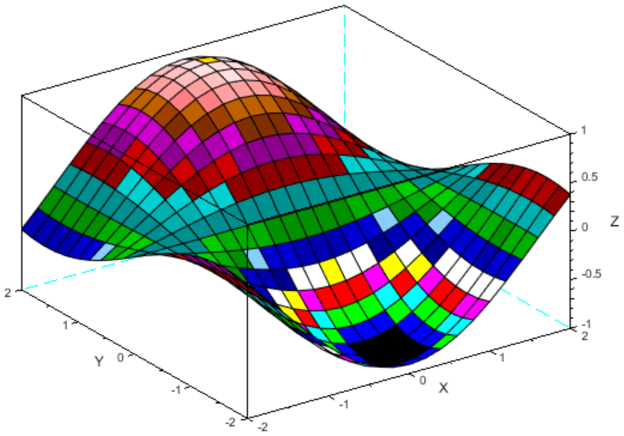

Рисунок 10.6 – Результат роботи функції `surf()`

```
x = -2:0.2:2;
y = -2:0.2:2;
[X, Y] = ndgrid(x, y);
Z = sin(X) .* cos(Y);
mesh(x, y, Z);
```

Результат див. рис. 10.7:

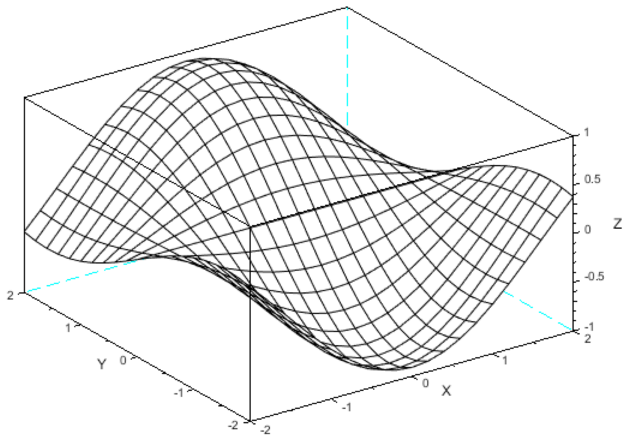

Рисунок 10.7 – Результат роботи функції `mesh()`

У Scilab гістограми та кругові діаграми можна будувати вбудованими командами, і це робиться досить просто.

1. Гістограма. Використовується команда `histplot` або `hist3d` (для 3D). Наприклад:

```
// Дані
```

```
data = grand(1, 100, "uin", 0, 10); // 100 випадкових чисел
від 0 до 10

// Побудова гістограми
histplot(10, data);  // 10 стовпчиків
xlabel("Значення");
ylabel("Частота");
title("Гістограма випадкових чисел");
```

Результат див. рис. 10.8:

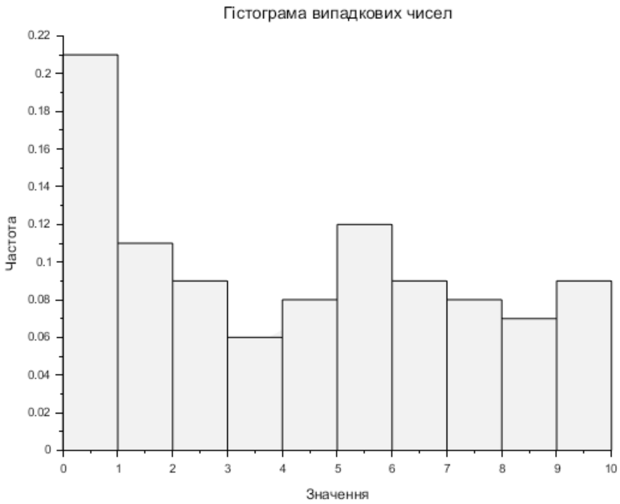

Рисунок 10.8 – Гістограма випадкових чисел

2. Кругова діаграма. Scilab має функцію `pie()`:

```
// Дані
values = [40, 25, 15, 20];
labels = ["A", "B", "C", "D"];

// Побудова кругової діаграми, values – числові значення
секторів, labels – підписи секторів.
pie(values, labels);
title("Кругова діаграма");
```

Результат див. рис. 10.9:

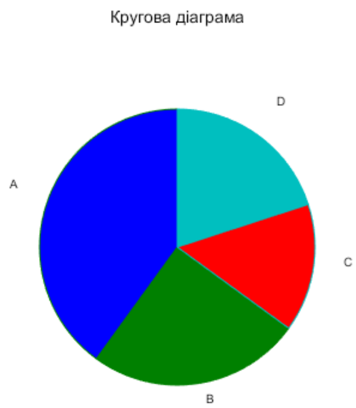

Рисунок 10.9 – Кругова діаграма у Scilab

Отже, Scilab має широкі можливості для побудови графіків — від простих 2D-кривих і багатосерійних діаграм до складних 3D-поверхонь та контурних карт. Система підтримує гістограми, кругові діаграми, а також анімацію та інтерактивне масштабування. Для візуалізації доступні численні параметри налаштування кольорів, маркерів, підписів та стилів ліній, що дозволяє створювати як наукові графіки, так і наочні ілюстрації для навчання та презентацій.

Завдання до лабораторної роботи

1. Побудувати графіки функцій, використовуючи основні опції функцій `plot()` і `plot2d()` у декартовій системі координат:

1) a) $y = \frac{x^5}{(x^2-1)^2}$; b) $y = \sin x + \frac{1}{2}\sin 2x + \frac{1}{3}\sin 3x$.

2) a) $y = \sqrt[3]{x(3-x)^2} - x$; b) $y = \left(\frac{x+1}{x-1}\right)^4$.

3) a) $y = \frac{20x^2}{(x-1)^3}$; b) $y = \frac{\ln^2 x}{x}$.

4) a) $y = 8x^2(x^2-1)^3$;b) $y = \frac{x}{2} - \arccos\frac{2x}{1+x^2}$.

5) a) $y = \frac{x^5-8}{x^4}$; b) $y = \arccos\frac{1-x^2}{1+x^2}$.

6) a) $y = \frac{(x-5)^3}{(x-7)^2}$; b) $y = \frac{x}{\ln x}$.

7) a) $y = \frac{x^3-2x^2-x+2}{x}$; b) $y = \frac{x^2\sqrt{x^2-1}}{2x^2-1}$.

8) a) $y = \frac{x^3}{x^2-1}$; b) $y = (x^2-2)e^{-2x}$.

9) a) $y = \frac{1+x^2}{1+(x-2)^2}$; b) $y = \ln\left|\frac{1-x}{1+x}\right| + \frac{6}{1+x}$.

10) a) $y = \frac{x^2+2x-3}{x} - e^{\frac{1}{x}}$; b) $y = \frac{x}{2} + 2\,\mathrm{arctg}\, x$.

2. Побудувати графіки функцій, заданих у параметричній системі координат:

1) $x = \frac{9}{2}\cos^3 t\,, y = \frac{9}{4}\sin^3 t\,, t \in [0; 2\pi]$.

2) $x = t^3 - 2, y = t - 2, t \in [0; 10]$.

3) $x = 5\left(\cos t + \ln\left(\mathrm{tg}\frac{t}{2}\right)\right), y = 5\sin t\,, t \in [0; 2\pi]$.

4) $x = 2(t - \sin t), y = 2(1 - \cos t), t \in [0; 2\pi]$.

5) $x = t - \frac{\mathrm{sh}\, 2t}{2}, y = 2\,\mathrm{ch}\, t\,, t \in [0; 15]$.

6) $x = (t^2-2)\sin t + 2t\cos t\,, y = (t^2-2)\cos t - 2t\sin t\,, t \in [0; \pi]$.

7) $x = 3(\mathrm{sh}\, t - t), y = 3(\mathrm{ch}\, t - 1), t \in [0; 21]$.

8) $x = \cos^3 t\,, y = \sin^3 t\,, t \in \left[0; \frac{\pi}{2}\right]$.

9) $x = 2t^3(1-t^2), y = t\sqrt{7}, t \in [0; 12]$.

10) $x = 2t^2, y = t\left(\frac{1}{4} - t^2\right), t \in [0; 20]$.

3. Побудувати графіки функцій, заданих у полярній системі координат:

1) $\rho = 5\sin^2 2\varphi$.

2) $\rho = 7\cos^2\varphi$.
3) $\rho = 2(1 + \sin^2\varphi)$.
4) $\rho = 4\sin 2\varphi$.
5) $\rho = 9\cos 5\varphi$.
6) $\rho = 4(1 + \cos\varphi)$.
7) $\rho = 3\cos 3\varphi$.
8) $\rho = 8\cos 4\varphi$.
9) $\rho = 7\sin 5\varphi$.
10) $\rho = 2\sin 4\varphi$.

4. Побудувати поверхні, використовуючи основні опції функції `plot3d()` у декартовій системі координат:

1) $z = x^2\sin(2x + 2y)$.
2) $z = \sin x + \cos y + e^{-(x^2+y^2)}$.
3) $z = e^{-(x^2+y^2)}$.
4) $z = e^{-y^2}\cdot\sin x$.
5) $z = e^{-x^2}\cdot\cos y$.
6) $z = \sin x\cdot\cos y$.
7) $z = e^{(\sin x+\cos y)}$.
8) $z = e^{-0.1(x^2+y^2)}\sin\sqrt{x^2 + y^2}$.
9) $z = e^{-x^2}\cdot\cos(x^2 + y^2)$.
10) $z = e^{\cos x\cdot\sin y}$.

5. Під час опитування 50 учнів школи з’ясували, які фрукти вони найбільше люблять: яблуко – 14; банан – 10; апельсин – 8; груша – 12; виноград – 6. Побудуйте гістограму, що відображає кількість учнів, які віддають перевагу кожному фрукту. Побудуйте кругову діаграму, щоб показати відсотковий розподіл симпатій до фруктів.

Питання для самоконтролю

1. Охарактеризуйте можливості візуалізаціїфункцій у СКА Scilab.
2. Розкажіть про способи побудови двомірних графіків функцій у СКА Scilab.
3. Чи є окрема функція для побудови графіка функції у параметричній системі координат у СКА Scilab?
4. Чи є можливість у СКА Scilab будувати функції, які задані неявно?
5. Поясніть особливості побудови гістограм та кругових діаграм у СКА Scilab.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. Анісімов А. В., Дорогий Я. Ю., Погорілий С. Д., Дорогий Я. Ю. Програмування числових методів мовою Python : підручник. Київ : Київський університет, 2014. 640 с.
2. Affouf M. Scilab by Example. Publisher : CreateSpace Independent Publishing Platform, 2012. 114 p.
3. Maxima – A Computer Algebra System. URL : https://maxima.sourceforge.io/documentation.html
4. Maxima Manual Version 5.47.0. URL : https://maxima.sourceforge.io/ext/maxima.pdf
5. Roux P. Scilab from Theory to Practice. Publisher : Editions D-Booker, 2016. 414 p.
6. Sage Documentation. URL : https://doc.sagemath.org/
7. Sage Tutorial. URL : https://doc.sagemath.org/html/en/tutorial/index.html
8. SageMathCell. URL : https://sagecell.sagemath.org/
9. Scilab Tutorial. URL : https://www.scilab.org/tutorials
10. Verma R. Introduction to Scilab (Student Edition). Publisher : Independently published, 2018. 280 p.
11. Zimmermann P. Computational Mathematics with SageMath. Publisher : SIAM- Society for Industrial and Applied Mathematics, 2018. 464 p.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Собчук В. В., Чичурін О. В., Кальчук І. В., Жигалло Т. В. Розв’язування задач аналізу та диференціальних рівнянь засобами комп’ютерної алгебри Mathematica. Київ : Міленіум, 2021. 420 с.
2. Фетісов В. С. Математична система Scilab : навчально-методичний плсібник, 2-ге видання, перероблене і доповнене. Ніжин : Ніжинський державний університет імені Миколи Гоголя, 2022. 82 с.
3. Чичкарьов Є. А. Підручник-довідник із системи комп’ютерної алгебри Maxima. Київ : ALT Linux, 2020. 186 с.
4. Esslinger B. Introduction into the CAS SageMath. Publisher : CrypTool Project, 2024. 47 p.
5. Kishor K. Scientific Computing Using Scilab. Publisher : Independently published, 2022. 81 p.
6. Kumar V. Basics of SageMath : Mathematics (Practical). Publisher : Amazon KDP, 2022. 248 p.
7. Öchsner A., Makvandi R. Finite Elements Using Maxima : Theory and Routines for Rods and Beams. Publisher : Springer, 2020. 256 p.

Навчальне видання
(*українською мовою*)

Панасенко Євген Валерійович

МАТЕМАТИЧНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації до лабораторних занять
для здобувачів ступеня вищої освіти бакалавра
спеціальності Е7 Математика
освітньо-професійної програми «Математика»

Рецензент *А.О. Лісняк*
Відповідальний за випуск *С.М. Гребенюк*
Коректор *Є.В. Панасенко*