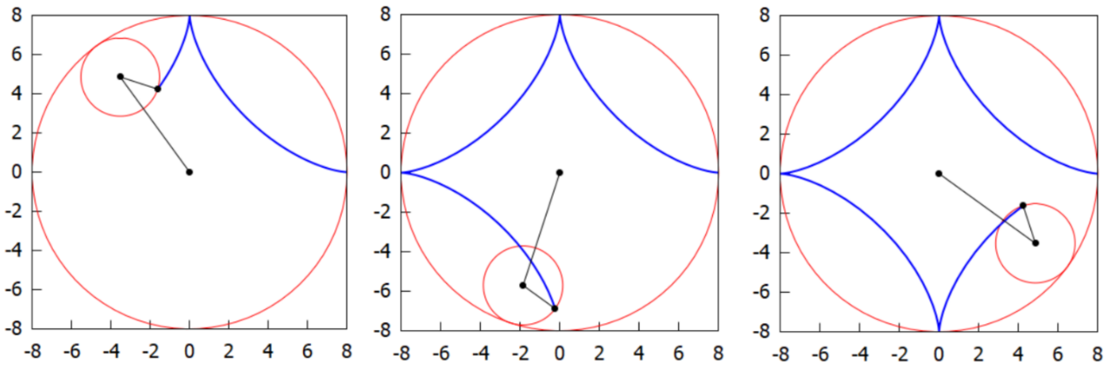
Міністерство освіти і науки України

Ніжинський державний університет імені Миколи Гоголя

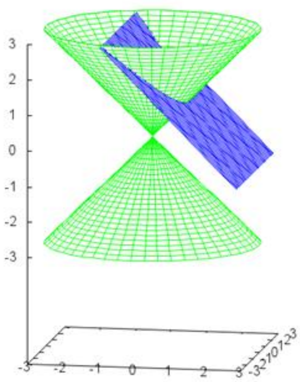
Бугаєць Н.О.

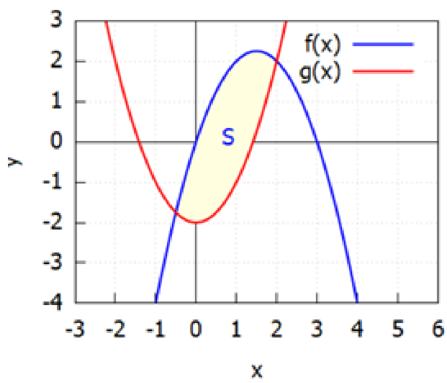
Математика з програмою Maxima

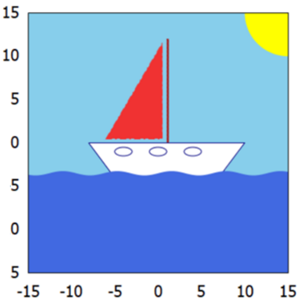
Навчальний посібник



```
(%i26) wxplot2d([x^2, −x+1, x, sin(x)], [x,−2.5, 2.5],
       [box, false], grid2d,
       [yx_ratio, 1], [axes, solid],
       [xtics, −2, 1, 2],  [ytics, −1, 2, 5],
       [label, ["x", 2.3, 0.2], [ "y", −0.2, 6]],
       [title, "Графіки функцій"]);
```







2023

УДК 004.9
Б90

Рекомендовано Вченою радою
Ніжинського державного університету імені Миколи Гоголя
Протокол № 12 від 30.06.2023

Рецензенти:

Шевчук Л.Д. – доктор педагогічних наук, професор, завідувач кафедри математики, інформатики і методики навчання Університету Григорія Сковороди в Переяславі.

Лисенко І.М. – кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних технологій, фізико-математичних та економічних наук Ніжинського державного університету імені Миколи Гоголя.

Бугаєць Н.О.
Б90 Математика з програмою Maxima: навчальний посібник. – Ніжин: НДУ ім. М. Гоголя, 2023. – 163 с.

Даний навчальний посібник розроблений для студентів спеціальностей 014 Середня освіта (Математика), 014 Середня освіта (Інформатика), 014 Середня освіта (Фізика), 122 Комп'ютерні науки. Теоретичний матеріал та практичні завдання викладено в обсязі, достатньому для опанування засобами системи комп'ютерної математики Maxima для проведення чисельних та символьних обчислень, розв'язування задач лінійної алгебри, аналітичної геометрії, матаналізу, побудови двовимірних та тривимірних графіків, математичного та комп'ютерного моделювання.

Навчальний посібник може бути використаний не тільки студентами вказаних спеціальностей, а й учителями та учнями загально-освітніх та професійно-технічних навчальних закладів освіти під час вивчення тем, пов'язаних з математичним моделюванням, проведенням графічних досліджень за допомогою комп'ютерних засобів.

Зміст

Вступ

На межі інтеграції математики та інформатики виникає поняття комп'ютерної математики, яку визначають як сукупність теоретичних, алгоритмічних, апаратних і програмних засобів, призначених для ефективного розв'язування за допомогою комп'ютера всіх видів математичних задач з високою мірою візуалізації всіх етапів обчислень.

Комп'ютерна математика – це перш за все синтетична назва для цілої серії комп'ютерних математичних систем, у яких акумульовані багатовікові знання людства в галузі математичних методів обчислень і розрахунків, що виконуються як у чисельній, так і в аналітичній та графічній формах [6]. Використання таких систем значно підвищує користувачеві можливості та результативність його навчально-пізнавальної, наукової, педагогічної або іншої творчої діяльності. Вони стають потужними засобами діяльності як професійних математиків, так і тих, хто використовує математику для побудови й дослідження математичних моделей в різних предметних галузях.

Maxima – це одна з найвідоміших систем комп'ютерної математики, історія розробки якої почалася у Масачусетському технологічному інституті (MTI) в кінці 60-х років минулого століття у вигляді проєкту MACSYMA (Проект Mac's SYmbolic MAnipulation System), що виконувався групою Mathlab у лабораторії комп'ютерних наук MTI (спочатку відомої як Project MAC). Ліцензування в 70-ті рр. програмних кодів Macsyma призвело до створення інших прикладних математичних пакетів – Maple фірми Waterloo Maple Inc. та Mathematica фірми Wolfram Research та ін. Спільність цих програмних продуктів виражається як у схожому синтаксисі, так і в спільних алгоритмах.

Новий етап у розвитку Maxima настав у 1999 році, коли минув термін дії патенту, і права на Maxima повернулись до одного з її авторів – Вільяма Шелтера, який виконав повну переробку системи та залучив до її відкритої розробки провідних фахівців. З листопада 2001 року проект Maxima підтримується роботою команди на чолі з Джеймсом Амундсоном і Ричардом Фейтманом.

На даний час Maxima – це вільно поширювана відкрита система, що розповсюджується за ліцензією GPL.

Широке застосування система Maxima набуває завдяки можливості її інтегрування у різні середовища (в т.ч. на основі Web-технологій). Це дозволяє користувачу застосовувати різні візуальні інтерфейси до ядра системи та створювати власні.

Сьогодні проєкт продовжує активно розвиватися, і участь у ньому є кращою візитівкою для математиків та програмістів з усього світу. Завдяки зусиллям інтернаціональної команди розробників Maxima набула ряд особливостей:

1) система повністю відкрита, ліцензійно чиста та безоплатна;

2) система незалежна від використовуваної операційної системи й апаратної платформи;

3) багаторічний досвід удосконалення системи призвів до появи в ній повністю відлагоджених, швидких та оптимізованих алгоритмів;

4) система проста в інсталяції, мала за розміром та невимоглива до апаратних ресурсів.

Програма періодично оновлюється і постійно виходять більш досконалі версії. Установити останню версію програми або будь-яку іншу на свій комп’ютер можливо з офіційного сайту https://maxima.sourceforge.io/ та з сайту завантаження дистрибутиву для операційної системи Windows https://maxima.sourceforge.io/download.html.

В даному навчальному посібнику розглядаються задачі, які розв’язані за допомогою програми Maxima версії 5.42.2, яка має українську локалізацію.

1 Основи роботи в середовищі системи Maxima

1.1 Середовище системи Maxima, основи синтаксису, введення найпростіших команд у програмі Maxima, типи даних

Вікно системи Maxima складається з елементів (рис. 1.1) :

1) рядок назви (імені) документу;
2) рядок головного меню;
3) панель інструментів;
4) робоче поле;
5) рядок стану;
6) полоси прокручування.

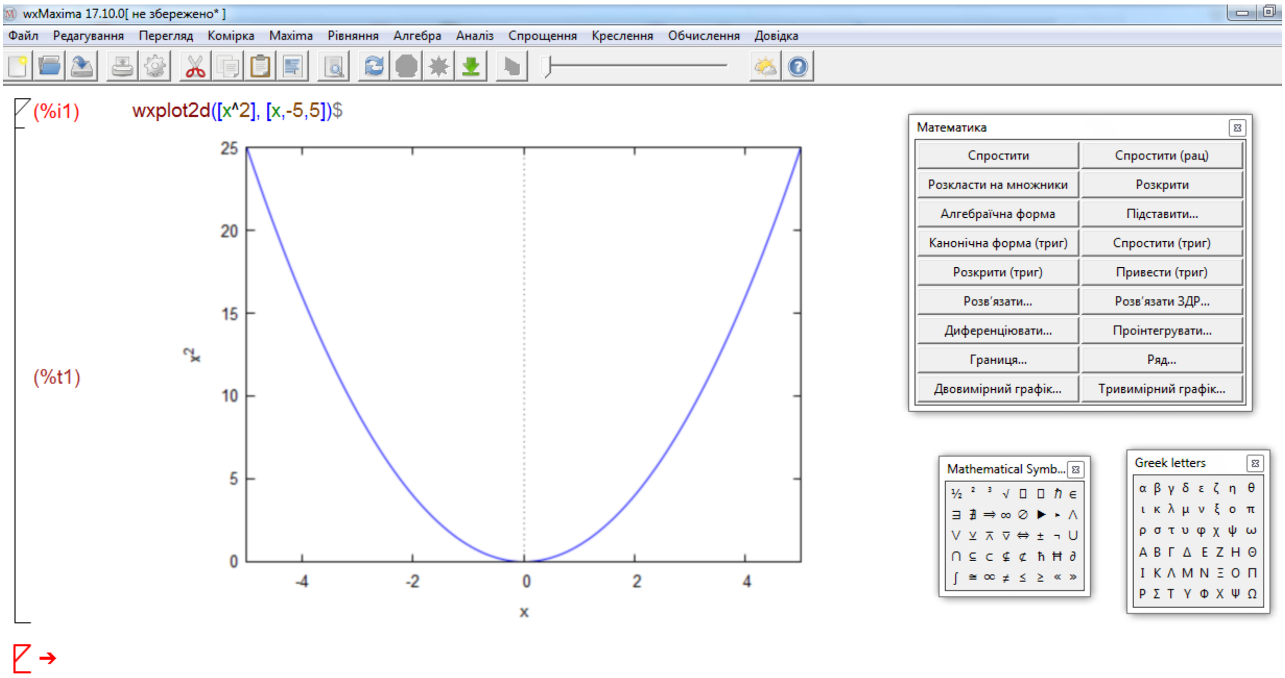


Рис. 1.1 Вікно програми Maxima

Робота в Maxima здійснюється в режимі сесії – користувач вводить команди, вирази, текст, які сприймаються як певні умови та дані і опрацьовуються в системі.

Робоче поле складається з блоків (комірок, англ. *cell*), що обмежені зліва прямокутною дужкою. Блок ділиться на дві частини (рис. 1.1):

1. Поле введення – складається з командних рядків (%i1, %i2,…). Команда вводиться справа від спеціального символу —>, який з'являється на екрані після

введення 1-го символу. В кінці кожної команди ставиться крапка з комою. Якщо замість крапки з комою в кінці команди записати символ $, команда виконується, але результат обчислення не виводиться на екран. Для виконання команди натискається комбінація клавіш Ctrl+Enter. Для того, щоб продовжити введення команд в наступному рядку натискають клавішу Enter. Якщо область введення займає занадто багато місця на екрані, її можна згорнути, клацнувши мишею на трикутнику у верхньому куті квадратної дужки.

2. Поле виведення – містить результати опрацювання введених команд (%o1, %o2,…), графічних об'єктів або повідомлень про помилку.

Для запису будь-яких текстових даних, наприклад, умови задачі, коментарів, пояснень тощо область введення можна перетворити в текстову область, скориставшись послугою *Cell→Insert Text Cell* або натиснути мишею кнопку *Текст* на панелі *Вставити*. За допомогою аналогічних послуг даного меню створюються текстові заголовки та підзаголовки.

Щоб додати текстовий коментар, безпосередньо в області введення команди, текст відокремлюється символами /* */. В такому випадку текст ігнорується інтерпретатором, але є зауваження, область введення не може закінчуватися коментарем.

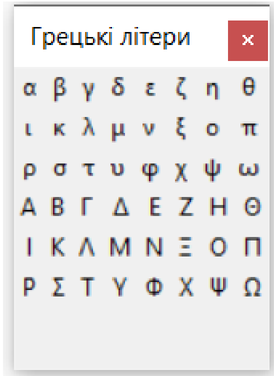
Введення команд в Maxima здійснюється згідно правил синтаксису спеціальної мови програмування. До основних елементів мови програмування в Maxima відносяться ідентифікатори, константи, змінні, функції, процедури і оператори.

Ідентифікатор – основна синтаксична одиниця мови програмування. *Ідентифікатори* – це всеможливі імена констант, змінних, функцій і операторів.

За деякими ідентифікаторами в Maxima закріплений певний зміст, наприклад `%pi` – число π; `limit` – функція для обчислення границі. Такі ідентифікатори називають службовими.

Рядки поля введення та виведення нумеруються, тому ідентифікатори %i1, %i2,… та %o1, %o2,… є посиланнями на відповідний рядок (від англ. *input* – введення, *output* – виведення). Ідентифікатор % посилається на попередній результат. Наприклад:

```
(%i1) 3;
(%o1) 3
(%i2) 5;
(%o2) 5
(%i3) %i1+%i2;
(%o3) 8
(%i4) %+%;
(%o4) 16
```



Для подання букв грецького алфавіту в Maxima використовують спеціальну панель (*Перегляд/Грецькі літери*) або ідентифікатори:

`%alpha, %beta, %gamma, %delta, … , %omega;`

`%Alpha, %Beta, %Gamma, %Delta, … , %Omega.`

Наприклад:

```
(%i6) sin(π·α)+cos(π);
      sin(%pi·%alpha);
(%o5) sin(π α)−1
(%o6) sin(π α)
```

Константи

В Maxima є системні (зарезервовані) змінні або константи:

`%pi`	число π
`%e`	основа натурального логарифма, Ейлерове число
`%i`	уявна одиниця ($\sqrt{-1}$)
`true,`	логічні константи, що позначають хибність чи істинність
`false`	виразу (висловлювання)
`inf`	додатна системна нескінченність
`minf`	від'ємна системна нескінченність

Системні змінні не можна вилучити, або змінити їх значення. Наприклад, під час спроби надати такій змінній якогось значення, програма видасть повідомлення про помилку.

Через `%c` з номером позначаються константи, які використовуються під час інтегрування, для яких використання літери «с» традиційне в математиці.

Змінні

Для збереження результатів розрахунків використовуються змінні. Змінною може бути будь-який ідентифікатор (окрім зарезервованих), який складається з латинських букв і цифр, а також символів кирилиці.

Великі і малі літери розрізняються.

Імена змінних повинні починатися тільки з літери і не можуть містити пробілів. Ім’я змінної не може збігатися з іменами вбудованих функцій і системних змінних.

Змінній може присвоюватися будь-яке значення за допомогою оператора присвоєння (двокрапка). Змінна, якій не присвоєне жодне значення, вважається вільною змінною, її ім’я зберігається в арифметичних обчисленнях.

```
(%i10) a: 5;
       b: 8+2·a;
(a)    5
(b)    18
(%i11) (a+c)·b;
(%o11) 18 (c+5)
(%i12) (a+c)·'b;
(%o12) b (c+5)
```

Якщо необхідно видалити значення змінної з пам’яті (очистити її), то застосовується команда `kill`: `kill(`*x*`)` – видалити значення змінної *x*; `kill(all)` – видалити значення всіх змінних, які використовувалися до цього. Крім того, `kill` розпочинає нову нумерацію для виконуваних команд.

Якщо на змінну потрібно накласти обмеження, використовують функцію `assume(cond)`, де `cond` – умова, що накладається на змінну. Функцією `forget(cond)` знімають обмеження `cond` зі змінної. Наприклад:

```
(%i1) sqrt(a^2);
(%o1) |a|
(%i3) assume(a>=0)$
      sqrt(a^2);
(%o3) a
```

Оператори та функції

Всі обчислення та перетворення виразів здійснюються за допомогою операторів та функцій. В Maxima використовують оператори:

	Арифметичні
`+,-,*,/`	додавання, віднімання, множення ділення
`^` або `**`	піднесення до степеня
	Логічні
`<,>,>=,<=,`	менше, більше, більше або дорівнює, менше або дорівнює,
`=,#`	дорівнює, не дорівнює
	Присвоєння
`:`	присвоєння значення змінній
`:=`	визначення функції
	Апострофи
`'`	застосування перед будь-яким символом або виразом, щоб запобігти його обчисленню (екранування)
`''`	застосовується для примусового обчислення виразу

Функція в системі символьних обчислень – це підпрограма або процедура, що має своє унікальне ім’я і в якій виконуються перетворення над аргументами.

В загальному вигляді функції в системі Maxima записуються наступним чином: `function(x1, x2,…, xn)`, де `function` – ім’я функції (ідентифікатор), `x1, x2,…, xn` – аргументи функції.

Всі функції в системі Maxima діляться на дві групи: основні або вбудовані (викликаються безпосередньо) та додаткові, що містяться в спеціалізованих пакетах. Для використання додаткових функцій з інших пакетів, необхідно завантажити відповідний пакет командою `load(pack);` де pack – назва пакету.

Елементарні математичні функції Maxima

Запис в Maxima	Математичний запис
`abs(x)`	$\lvert x \rvert$
`sqrt(x)`	$\sqrt{x}$
`^(1/n)`	$\sqrt[n]{x}$
`log(x)`	$\ln x$
Тригонометричні функції	
`sin(x)`	$\sin x$
`cos(x)`	$\cos x$
`tan(x)`	$\mathrm{tg}\, x$
`cot(x)`	$\mathrm{ctg}\, x$
Обернені тригонометричні функції	
`asin(x)`	$\arcsin x$
`acos(x)`	$\arccos x$
`atan(x)`	$\mathrm{arctg}\, x$
`acot(x)`	$\mathrm{arcctg}\, x$

В головному меню та на панелі інструментів знаходяться послуги функцій для розв’язування багатьох типових задач, які розділені на групи: Рівняння, Алгебра, Аналіз, Спростити, Чисельні обчислення та ін. При введенні команди за допомогою послуг меню з’являється вікно функції, де необхідно вказати лише потрібні параметри. Це може значно полегшити запис команди. Наприклад, послугу *Аналіз→Диференціювати…* застосовують для знаходження похідної функції, вираз якої вводять у відповідне поле вікна *Диференціювати* (рис. 1.2).

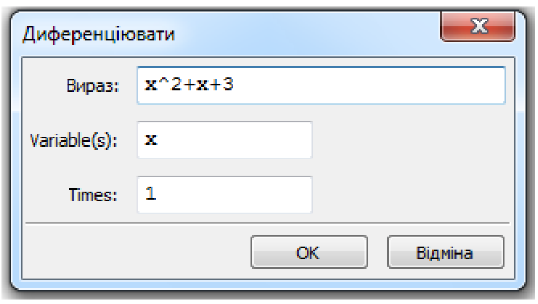


Рис. 1.2 Диференціювання функцій в Maxima

В системі Maxima є можливість визначення функцій користувача (таблиця 1.1). Для цього використовують оператор := або функцію `define`. Визначення функції таким способом дозволяє знаходити її значення в точці.

Таблиця 1.1

Визначення функцій користувача в системі Maxima

Приклад	*Пояснення*
`f(x):=expr`	визначення функції $f(x)$від однієї змінної
`define(f(x),expr)`	Перетворення виразу expr в функцію $f(x)$ від однієї змінної
`f(x1,x2,…,xn):=expr`	визначення функції від n змінних
`define(f(x1,x2,…,xn),expr)`	Перетворення виразу expr в функцію від n змінних
`f(a)`	значення функції в точці $x=a$
`subst(x=a,f(x))`	значення функції в точці $x=a$
`f(a,b)`	значення функції від двох змінних в точці $x=a,\ y=b$
`subst([x=a,y=b],f(x,y))`	значення функції від двох змінних в точці $x=a,\ y=b$

Приклад 1.1. Визначити функції та обчислити значення функції в точках $x=-5$, $x=9$: а) $y = x^3 - 8x^2 + 5x + 1$, б) $y = x^2 + 2x - 3$.

```
(%i3) /*I спосіб*/
      y(x):=x^3−8*x^2+5*x+1; /*визначаємо функцію y(x)*/
      y(−5); y(9); /*обчислюємо значення функції в точках*/
(%o1) y(x):=x^3-8 x^2+5 x+1
(%o2) -349
(%o3) 127
```

```
(%i7)  /·II спосіб·/
       y: x^2+2·x−3; /·присвоюємо змінній y вираз функції·/
       define(y(x),y); /·визначаємо функцію y(x)·/
       y(−5); /·обчислюємо значення функції в x=−5·/
       subst(x=9,y(x)); /·обчислюємо значення функції в x=9·/
```

(y) x^2+2x-3

(%o5) $y(x):=x^2+2x-3$

(%o6) 12

(%o7) 96

Приклад 1.2. Визначити функцію $f(x,y)=1-\sin(x^2+y^2)$ та обчислити значення функції $f(1,2)$, $f(0,a)$.

```
(%i10) f(x,y):=1−x^2+y^2;
       f(1,2);
       f(0,a);
```

(%o8) $f(x,y):=1-x^2+y^2$

(%o9) 4

(%o10) a^2+1

1.2 Типи даних в системі Maxima

В програмі Maxima опрацьовують дані різних типів: числа, рядки, списки, множини, масиви, дані логічного типу. Після присвоювання змінній значення над об'єктом можна виконувати операції, призначені для цього типу даних. Наприклад, не можна до числа додати рядок.

До числового типу даних відносяться натуральні, цілі, раціональні, ірраціональні, дійсні, комплексні числа.

Натуральні числа – це числа, які використовують для лічби: 1,2,3,…

Цілі числа – це натуральні числа, протилежні до них та число нуль: …–2, –1, 0, 1, 2…

Раціональні числа – числа, які можна подати у вигляді $\frac{m}{n}$, де m – ціле число, n – натуральне. Кожне раціональне число можна подати у вигляді десяткового дробу, скінченного або періодичного.

Ірраціональні числа – це нескінченні і неперіодичні десяткові дроби. У вигляді десяткового дробу в системі Maxima можна подати лише з певною точністю, але подання цих чисел в символьному вигляді і можливість виконувати символьні обчислення допомагають одержати результат з високою точністю під час розрахунків. Наприклад, ірраціональними є числа $\sqrt{2}$, $\cos 7$, π тощо.

Дійсні числа – це раціональні та ірраціональні числа.

Розглянемо дії, які виконують з числовим типом даних в програмі Maxima.

Цілі числа

Для роботи з цілими числами в системі Maxima є такі функції:

`evenp(n)`, `oddp(n)`	логічні функції, які перевіряють, чи n парне або непарне число, повертають значення true або false.
`mod(x,y)`	повертає число $z \in [0; y)$, таке що $\frac{x-z}{y}$ – ціле число (остача від ділення)

`ifactors(n)`	повертає список простих дільників p1, p2, … та їх степенів k1,k2,… в наступному форматі $[[p1,k1],[p2,k2],...]$
`factor(n)`	повертає число n у вигляді добутку степенів простих чисел
`divisors(n)`	повертає множину дільників числа n, включаючи 1 і саме число
`lcm(n1,n2,…)`	повертає найменше спільне кратне (НСК) чисел $n1,n2,...$
`gcd( n1,n2,…)`	повертає найбільший спільний дільник (НСД) чисел $n1,n2,...$
`primep(n)`	перевіряє, чи є число n простим.
`next_prime(n)`	повертає найближче просте число, більше за n
`prev_prime(n)`	повертає найближче просте число, менше за n

Раціональні числа, дійсні числа (з плаваючою крапкою)

Для роботи з раціональними та дійсними числами в системі Maxima використовують функції:

`rationalize(x)`	повертає число x у вигляді раціонального дробу
`rat(x)`	повертає число x наближено у вигляді раціонального дробу. Похибка, яку має функція, задається глобальною змінною `ratepsilon` (за замовчуванням $2.0e-8$)
`ev(expr)`	повертає чисельне значення виразу `expr`
`numer`	глобальна змінна, за замовчуванням має значення `false`, обчислення здійснюються у символьному вигляді. Щоб числові функції обчислювались у числах з плаваючою крапкою `numer:true`
`float(expr)`	подання раціонального числа у вигляді числа з плаваючою крапкою (десятковий дріб з деякою точністю, за замовчуванням 16 знаків після коми).

`bfloat(expr)`	виведення числа в експоненціальній формі; буква `b` в записі числа використовується замість *e* для позначення типу *big float* (дійсні числа довільної точності).
`fpprec`	глобальна змінна, яка визначає число значущих цифр в арифметиці чисел з плаваючою крапкою (за замовчув. 16)
`ceiling(x)`	округлення x в більшу сторону
`floor(x)`	округлення x в меншу сторону
`round(x)`	округлення x до найближчого парного числа

У процесі обчислень в Maxima результат обчислень подається в символьному вигляді, а саме у вигляді, наприклад, раціонального або ірраціонального виразу.

```
(%i1) 1/3;
```
(%o1) $\frac{1}{3}$

```
(%i2) sqrt(3)+sqrt(5);
```
(%o2) $\sqrt{5}+\sqrt{3}$

Одержати чисельне значення виразу можна за допомогою функції **`ev`**. Її перший аргумент є виразом, що обчислюється, а другий – опція **`numer`**.

```
(%i3) ev(%, numer);
(%o3) 3.968118785068667
```

Допускається більш зручна форма функції **`ev`**, за якою вказуються тільки її аргументи.

```
(%i4) sqrt(3)+sqrt(5), numer;
(%o4) 3.968118785068667

(%i5) 1/3, numer;
(%o5) 0.3333333333333333
```

Рядки

Рядок (англ. «string») – це набір символів, який розміщений в подвійних лапках `"Математика"`.

Функції для роботи з рядками

`stringdisp`	глобальна змінна, яка визначає виведення на екран рядка з лапками чи без, за замовчуванням має значення false. Якщо надати значення true, рядок буде виведений на екран в лапках
`ascii(int)`	повертає символ за його ASCII-кодом, int=0,1,…255
`cint(char)`	повертає код символа
Рядок як послідовність символів	
`slength(str)`	повертає довжину рядка
`charat(str,n)`	повертає n-ий символ рядка str
`charlist(str)`	повертає список символів рядка str
`sposition(char,str)`	повертає позицію першого входження символа char в рядок str
`simplode(list,del)`	перетворює список list в рядок, використовуючи рядок del як розділювач. За замовчуванням del=`""` (пробіл)
`split(str,del)`	повертає список підрядків, що розділені рядком del
`substring(str,n0,n1)`	Повертає частину рядка str, що відповідає позиціям від n0 до n1, не включаючи останню, або від n0 до кінця рядка, якщо n1 не вказувати
Рядок в цілому	
`string(expr)`	перетворення виразу в рядок, попередньо його спростивши. Спрощення можна відмінити за допомогою глобальної змінної simp: false
`eval_string(str)`	перетворення рядка у вираз та обчислення його

`concat(arg1,arg2,…)`	об'єднання аргументів функції у один рядок, причому аргументи повинні бути атомами
`sconcat(str1,str2,…)`	об'єднання аргументів функції у один рядок, які можуть і не бути атомами

Списки

Список (англ. *list*) – базова структура даних в системі Maxima. Всі інші типи даних (числа, рядки, множини, масиви, хеш-таблиці) можуть подаватися у вигляді списків. Список задається у вигляді переліку елементів через кому в квадратних дужках:

```
[elem1, elem2, …, elemn]
```

Список може бути пустим або складатися з одного елемента. Елементом списку може бути інший список. Наприклад:

```
list1: [1, 2, 3, a, b];
list2: [ ];
list3: [1, 2, 3, [1, 2], [3, 4, 5]];
```

Посилання на елемент списку здійснюється за номером елемента списку.

```
list1[4];
```

```
a
```

```
list3[5];
```

```
[3, 4, 5]
```

Фукції для роботи зі списками

`makelist(expr, i, i1, i2, step)` – створення списку із значень виразу expr, що залежить від значень змінної `i`, яка змінюється від `i1` до `i2` з кроком `step`. За замовчуванням крок вважається рівним одиниці.

`create_list(expr,i1,list1,i2,list2,…,in,listn)` – створення списку із значень виразу expr, що залежить від значень змінних i1,i2,…,in, які набувають значень `list1`, `list2,…`, `listn` відповідно

`length(list)` – визначення числа елементів списку `copylist(list)` – створення копії списку

`append(list1,…,listn)` – створюється один список, який містить послідовно всі елементи списків list1,…,listn

`join(l,m)` – створення нового списку, компонуючи елементи двох списків по-черзі. Довжина одержаного списку обмежується мінімальною довжиною списків l та m.

`cons(expr,list)` – створення списку, першим елементом якого буде expr, а решта – елементи списку list.

`endcons(expr,list)` – створення списку, першими елементами якого є елементи списку list, а останнім – елемент expr.

`reverse(list)` – зміна порядку елементів на зворотній

`rest(list, n)` – повертає залишок від списку, в якому відсутні

`unique(list)` – створення списку, що відповідає списку list, але без співпадання елементів

`map(f, list1,…,listn)` – застосування функції f (або оператора) до частин списків (виразів) list1, …, listn

`apply(f, list)` – застосування заданої функції f до всього списку.

Приклад 1.3. Створити список, елементами якого є квадрати натуральних чисел від 1 до 5.

```
(%i1)     list1: makelist(i^2, i, 1,5);
(list1)   [1,4,9,16,25]
```

Приклад 1.4. Створити список, елементами якого степені числа e з показниками, що дорівнюють елементам списку `list1` з попереднього прикладу.

```
(%i2)     list2: makelist(exp(i), i, list1);
```

(list2) $[\%e, \%e^{4}, \%e^{9}, \%e^{16}, \%e^{25}]$

Приклад 1.5. Створити список змінних від `x1` до `x7`.

```
(%i3)     list3: makelist(concat(x,i), i, 1, 7);
(list3)   [x1,x2,x3,x4,x5,x6,x7]
```

Приклад 1.6. Обчислити довжину списків list1, list2.

```
(%i10)  list1; list2;
        length(list1); length(list2);
(%o7)   [1,4,9,16,25]
(%o8)   [%e,%e^4,%e^9,%e^16,%e^25]
(%o9)   5
(%o10)  5
```

Приклад 1.7. Об’єднати списки list1, list2.

```
(%i11)  append(list1,list2);
(%o11)  [1,4,9,16,25,%e,%e^4,%e^9,%e^16,%e^25]
(%i12)  join(list1,list2);
(%o12)  [1,%e,4,%e^4,9,%e^9,16,%e^16,25,%e^25]
```

Множини

В математиці розглядають як скінченні так і нескінченні множини. В середовищі системи Maxima можна працювати лише зі скінченними множинами. Множина відрізняється від списку тим, що порядок елементів в ній не важливий, а елементи, які співпадають, вважаються одним і тим же елементом.

Множина (англ. «set») задається перерахунком елементів у фігурних дужках (n – ціле число).

```
{elem1, elem2, …, elemn}
```

Функції для роботи з множинами

`makeset(expr,[x1,…,xn],[[a11,…,an1],…,[a1m,…anm]])` – повертає множину, складену із значень виразу expr від векторної змінної, [x1,…,xn], що набуває значень векторів [a11,…,an1],…,[a1m,…anm]

`setp(S)` – логічна функція, яка перевіряє, чи S є множиною

`setequalp(S1,S2)` – логічна функція, яка перевіряє, чи співпадають множини S1, S2

Задачі для самостійного розв'язування

1. Обчисліть значення виразу в символьному вигляді та у вигляді числа з точністю до семи знаків після коми:

1.	$\sin\alpha+\cos\beta$, якщо $\alpha=\frac{1}{\sqrt{2}}$, $\beta=\frac{\pi}{3}$
2.	$\frac{a^2+b^2+c^2}{\sqrt{3}(a-b)}$, якщо $a=1$, $b=2$, $c=3{,}5$.
3.	$\frac{x-a}{x+1}$, якщо $x=-0.17$, $a=\pi$
4.	$x^2+a\lvert x\rvert+3$, $x=-5$, $a=1/3$
5.	$\sqrt[3]{x-6}+\sin\alpha$, якщо $x=-5$, $\alpha=4.2\pi$
6.	$c\sqrt{a^2+b^2}$, якщо $a=1.25$, $b=\sqrt{2}$, $c=0{,}5$.
7.	$\sqrt{a}+\sqrt[3]{b}$, якщо $a=17.8$, $b=-33{,}5$.
8.	$\sqrt{\sin\alpha}+\sqrt{16-x^2}$, якщо $x=1/22$, $\alpha=3.5\pi$
9.	$2\sin(\alpha+\beta)$, якщо $\alpha=1$, $\beta=\frac{\pi}{3}$
10.	$\frac{a+b^2+c^3}{\sqrt{2}(a-b)}$, якщо $a=e$, $b=2$, $c=3{,}5$
11.	Обчисліть об'єм циліндра за формулою $V=\pi r^2h$, якщо $r=2$ – радіус циліндра, $h=3$ – висота циліндра.

2. Визначте функцію, обчисліть значення функції в точці.

1.	$f(x)=-8x^2+5\lvert x\rvert-\sqrt{-x}+1$, $f(-1{,}5)$
2.	$f(x)=\frac{1}{2}\sin 7x-lnx$, $x=2{,}3$
3.	$f(x)=\frac{x}{x^2+3}-e^x$, $x=1{,}7$
4.	$f(x)=-2^x+\lvert x\rvert-1$, $f(4{,}3)$
5.	$f(x)=e^x+\cos\frac{1}{2}x$, $f(-3)$

6.	$f(x) = x^3 \cos x - \cos 3x, \ \ f(-1{,}25)$
7.	$f(x) = \lvert x-2 \rvert + \lvert 3x-4 \rvert, \ \ f(-0{,}33)$
8.	$f(x) = \frac{x^3}{2\sqrt{x+25}}, \ \ x = -5{,}1$
9.	$f(x) = e + tg\lvert 2x \rvert, \ \ x = -5{,}1$
10.	$f(x) = \sqrt[3]{x} - \ln(x-2), \ \ x = 8{,}2$

3. Створіть список

1.	Створіть список, елементами якого є куби парних чисел від 2 до 12. Знайдіть довжину списку.
2.	Створіть список із значень функції $f(x) = x^2 + 1$ в точках $x \in [-3; 3]$ з кроком 0,5. Знайдіть довжину списку.
3.	Створіть список, елементами якого є перші десять членів арифметичної прогресії, якщо перший член дорівнює 1,25, другий елемент дорівнює 1. Запишіть елементи створеного списку у зворотному порядку.
4.	Створіть список із значень функції $f(x) = x^2 + 1$ в точках $x \in [-3; 3]$ з кроком 0,25. Знайдіть довжину списку.
5.	Створіть список із значень функції $f(x) = x^3$ в точках $x \in [1; 10]$ з кроком 1. Зі створеного списку створіть новий список, компонуючи точку та значення функції в точці.
6.	Створіть список, елементами якого є квадрати натуральних чисел від 10 до 20. Знайдіть довжину списку.
7.	Створіть список, елементами якого є перші десять членів геометричної прогресії, $b_1 = 15,. b_2 = 5.$ Запишіть елементи створеного списку в зворотному порядку.
8.	Створіть список із значень функції $f(x) = \sin(x)$ в точках $x \in [0; 2\pi]$ з кроком $\frac{\pi}{3}$. Зі створеного списку створіть новий список, компонуючи точку та значення функції в точці.

9.	Перетворіть слово «Математика» в список, елементами якого є літери цього слова. Запишіть новий список, елементами якого є коди літер з таблиці символів ASCII.
10.	Створіть список, елементами якого є перші десять членів арифметичної прогресії, в якій $a_1 = -5,$. $a_2 = -3{,}5$. Створіть новий список, що складається з квадратів елементів першого списку.

Контрольні питання

1. Як правильно записати в Maxima команду для обчислення квадратного кореня числа *x*?

a) root(x, 2)   b) sqrt(x, 2)   c) sqrt(x)   d) sqr(x)

2. Яка команда в Maxima використовується для обчислення натурального логарифма числа?

a) ln(x)   b) log(x, *e*)   c) log(x)   d) lg(x)

3. Встановіть відповідність між типами даних:

1) [2, 4, 6, 8]	a) масив
2) {2, 4, 6, 8}	b) рядок
3) “2, 4, 6, 8”	c) список
	d) множина

4. Виберіть правильний запис команди, якою створюється список квадратів непарних чисел від 1 до 15:

a) makelist(i^2, 1, 15, 2);

b) makelist(i^2, i, 1, 15, 2);

c) makelist(i^2, i, 1, 15);

d) makelist(i^2, i, 1, 2, 15);

5. Встановіть відповідність між функцією (1-4) та її призначенням (a-d):

1) primep()	a) найбільший спільний дільник;
2) divisors()	b) найменше спільне кратне;
3) lcm()	c) множина дільників
4) gcd()	d) перевірка, чи є число простим;
	e) розклад числа на прості множники.

6. Виберіть можливі правильні варіанти чисельного обчислення виразу

a) ev($\sqrt{2} + \sqrt{3}$);

b) ev(sqrt(2)+sqrt(3), numer);

c) sqrt(2)+sqrt(3); numer;

d) sqrt(2)+sqrt(3), numer;

7. Встановіть відповідність між функцією (1-4) та її призначенням (a-d):

1) evenp()	a) повертає true, якщо число парне;
2) mod()	b) повертає true, якщо число непарне;
3) ceiling()	c) подання виразу у вигляді десяткового дробу з деякою точністю;
4) float()	d) округлення числа в більшу сторону;
	e) остача від ділення.

8. Що задає глобальна змінна fpprec?

a) кількість знаків після коми у числах, з плаваючою крапкою;

b) визначення функції;

c) параметр округлення;

d) параметр для прощення виразу;

9. Встановіть відповідність між функцією (1-4) та її призначенням (a-d):

1) sconcat()	a) довжина рядка;
2) string()	b) довжина списку;
3) length()	c) створення списку з елементів інших списків;
4) append()	d) перетворення виразу в рядок;
	e) об’єднання кількох рядків у один.

10. Вкажіть функцію, за допомогою якої створюють список з елементів інших списків, компонуючи їх по-черзі.

a) append();

b) makelist();

c) cons();

d) join().

1.3 Символьні перетворення та обчислення в системі Maxima

1.3.1 Вирази, перетворення виразів

Раціональні вирази – це вирази, складені із чисел і змінних за допомогою операцій додавання, віднімання, множення, ділення, піднесення до натурального степеня. Раціональний вираз називають *цілим*, якщо він не містить ділення на вираз зі змінною, інакше – дробовим.

Якщо вираз, крім чисел, змінних, дужок і знаків дій додавання, віднімання, множення, ділення і піднесення до степеня з раціональним показником або добування кореня, не містить нічого іншого, його називають *алгебраїчним виразом***.** Алгебраїчний вираз, який містить корені, називають *ірраціональним виразом***.** Усі інші алгебраїчні вирази – *раціональні***.**

Вирази із числами або змінними, які не є алгебраїчними, називають *трансцендентними***.** Такими, зокрема, є тригонометричні вирази.

Функції для перетворення виразів

`expand(expr)` – розкриття дужок виразу `expr`

`factor(expr)` – розклад на множники виразу `expr`

`rat(expr)` – зведення виразу до канонічної форми, розкриття дужок, зведення до спільного знаменника, скорочення і перетворення скінченних десяткових дробів у звичайні

`ratsimp(expr)` – спрощення виразу

`fullratsimp(expr)` – спрощення виразу

`radcan(expr)` – перетворення виразу, що містить логарифмічні, показникові і степеневі вирази

`trigsimp(expr)` – перетворення тригонометричних виразів

`trigexpand(expr)` – тригонометричне розкриття дужок; використовуються формули перетворення суми двох кутів для подання введеного виразу в якомога більш простому вигляді, де в якості аргумента тільки одна змінна

`at(expr, x=a)` – обчислення значення виразу, при `x=a` `divide(expr1, expr2)` – знаходження частки та остачі від ділення двох многочленів

`partfrac(expr, x)` – перетворення дробу у прості дроби за змінною x

Розкладання на прості дроби алгебраїчного дробу (такого дробу, що чисельник і знаменник обидва многочлени) — це операція, яка складається з вираження дробу як суми многочлена (можливо нуля) і одного або кількох дробів з простішими знаменниками.

Приклад 1.8. Розкрийте дужки $(x^2-2)(3x+2)(x-1.5)$.

```
(%i1) a: (x^2-2)·(3·x+2)·(x-1.5);
(a)   (x-1.5) (3 x+2) (x^2-2)
(%i2) expand(a);
(%o2) 3 x^4-2.5 x^3-9.0 x^2+5.0 x+6.0
```

Приклад 1.9. Розкладіть на множники x^3+2x^2-5x-6.

```
(%i1) m: x^3+2·x^2-5·x-6;
(m)   x^3+2 x^2-5 x-6
(%i2) factor(m);
(%o2) (x-2) (x+1) (x+3)
```

Приклад 1.10. Знайдіть частку та остачу від ділення многочленів x^4-x^2+x+1 та x^3-x^2+1.

```
(%i2) m1: x^4-x^2+x+1;
      m2: x^3-x^2+1;
(m1)  x^4-x^2+x+1
(m2)  x^3-x^2+1
(%i3) divide(m1,m2);
(%o3) [x+1,0]
```

Отже, часткою від ділення даних многочленів є двочлен $x+1$, а остача дорівнює нулю.

Приклад 1.11. Розкласти дріб $\frac{x^3+16}{x^3-4x^2+8x}$ на прості дроби.

```
(%i1) partfrac((x^3+16)/(x^3-4·x^2+8·x),x);
```

(%o1) $\frac{2x}{x^2-4x+8}+\frac{2}{x}+1$

Приклад 1.12. Спростити вираз а $+ \frac{81}{a-9} + 9$. Обчислити значення виразу, якщо $a = 0{,}5$.

```
(%i1) /*змінній expr присвоюємо значення даного виразу*/
      expr: a+81/(a-9)+9;
```

(expr) $a+\frac{81}{a-9}+9$

```
(%i2) rat(expr); /*спрощення виразу expr*/
```

(%o2)/R/ $\frac{a^2}{a-9}$

```
(%i3) /*обчислюємо значення виразу expr, якщо a=1/2 */
      at(expr, a=1/2);
```

(%o3) $-\frac{1}{34}$

Приклад 1.13. Спростити $\sqrt{a} + \frac{b-\sqrt{ab}}{\sqrt{a}+\sqrt{b}}$

Даний вираз містить корені, тому, щоб спростити вираз обираємо функцію `radcan()`. Для запису кореня квадратного використовуємо функцію `sqrt()`.

```
(%i1) expr: sqrt(a)+(b-sqrt(a*b))/(sqrt(a)+sqrt(b));
```

(expr) $\frac{b-\sqrt{a\,b}}{\sqrt{b}+\sqrt{a}}+\sqrt{a}$

```
(%i2) radcan(expr);
```

(%o2) $\frac{b+a}{\sqrt{b}+\sqrt{a}}$

Приклад 1.14. Спростити вираз $\frac{\cos\alpha-\cos 3\alpha}{\sin\alpha+\sin 3\alpha}$.

```
(%i3) k: (cos(a)-cos(3*a))/(sin(a)+sin(3*a));
      trigexpand(k);
      trigsimp(%);
```

(k) $\frac{\cos(a)-\cos(3\,a)}{\sin(3\,a)+\sin(a)}$

(%o2) $\frac{3\cos(a)\sin(a)^2-\cos(a)^3+\cos(a)}{-\sin(a)^3+3\cos(a)^2\sin(a)+\sin(a)}$

(%o3) $\frac{\sin(a)}{\cos(a)}$

1.3.2 Розв’язування рівнянь

В програмі Maxima для розв’язування рівнянь різного типу використовується послуга головного меню *Рівняння* (рис. 1.3). Наприклад, за послугою *Рівняння/Розв’язати…* застосовується функція *solve()* і у вікні *Розв’язати*, яке відкривається потрібно ввести вираз рівняння, що прирівнюється до нуля, та змінну, відносно якої буде розв’язуватися рівняння. В полі введення та виведення Maxima з’являється відповідна команда та розв’язок рівняння.

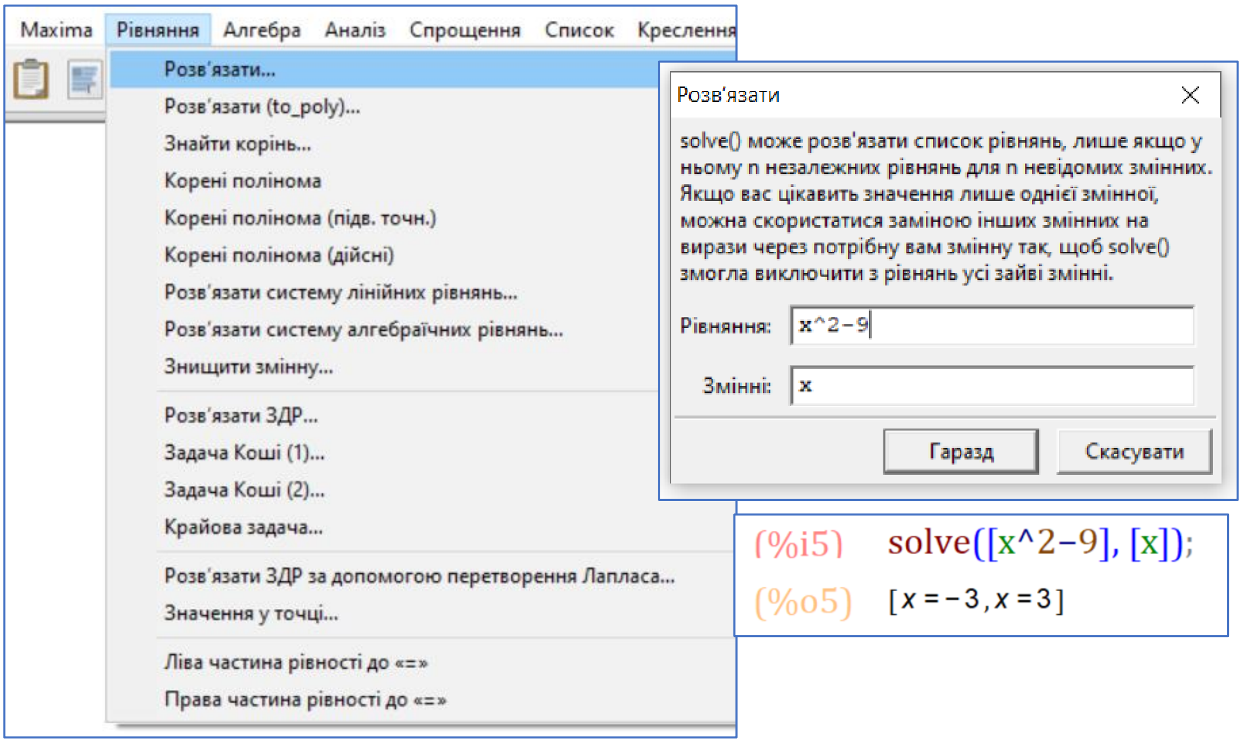


Рис. 1.3 Послуги меню Рівняння

Розглянемо функції в Maxima, які призначені для розв’язування рівнянь:

`solve(expr, x)`
`solve(eqn, x)`
`solve(expr)`
`solve(eqn)`
	розв’язування рівняння `expr` відносно змінної x; рівняння можна записувати двома способами: вираз рівняння `expr`, що дорівнює нулю, або як рівність `eqn`; якщо рівняння містить тільки одну змінну, то другий аргумент можна не вказувати.

`allroots(expr)`	пошук коренів алгебраїчних рівнянь n-го степеня, повертаються наближені значення коренів.

`find_root(eqn,x,a,b)` – пошук коренів за допомогою чисельних методів, використовуючи відокремлення змінної x на проміжку $[a; b]$

Приклад 1.15. Розв'язати рівняння $3x^2 + 4x - 7 = 0$.

Вираз рівняння присвоїмо змінній `eq`, а корені збережем у змінну `koreni`.

```
(%i4)    eq: 3·x^2+4·x−7=0;
         koreni: solve(eq);
(eq)     3 x^2+4 x−7=0
(koreni) [x=−7/3,x=1]
```

Одержали корені, один з яких – звичайний дріб, а другий – ціле число. Це саме рівняння можемо розв'язати за допомогою функції `allroots()`.

```
(%i5)    koreni: allroots(eq);
(koreni) [x=1.0,x=−2.333333333333333]
```

Слід звернути увагу, що, використовуючи `allroots()`, корені подаються у вигляді десяткових дробів із певною точністю.

Приклад 1.16. Розв'язати рівняння $x^3 + 1 = 0$.

```
(%i5)    kill(all)$
(%i2)    eq: x^3+1=0;
         koreni: solve(eq);
(eq)     x^3+1=0
```

(koreni) $[x = -\frac{\sqrt{3}\,\%i - 1}{2}, x = \frac{\sqrt{3}\,\%i + 1}{2}, x = -1]$

Одержали, що дане рівняння окрім дійсного кореня має і два комплексні корені.

Приклад 1.17. Розв'язати рівняння $\frac{1}{x-4} - \frac{1}{x-2} = \frac{1}{4}$

Дане рівняння має область допустимих значень – всі дійсні числа, окрім $x = 4$ та $x = 2$, що слід враховувати під час розв'язування. Для збереження розв'язків рівняння введемо змінну `roots`.

```
(%i1)    roots: solve(1/(x−4)−1/(x−2)=1/4, x);
(roots)  [x=0,x=6]
```

Одержали два корені, що задовольняють область допустимих значень. Слід звернути увагу, що розв'язок подається у програмі Maxima у вигляді списку, елементами якого і є корені рівняння. Це дає можливість звернутися до кореня як до елементу списку і, наприклад, виконати перевірку розв'язку. Для перевірки розв'язку далі запишемо команди:

```
(%i3)  1/(x-4)-1/(x-2)=1/4, roots[1];
```

(%o3) $\frac{1}{4}=\frac{1}{4}$

```
(%i3)  1/(x-4)-1/(x-2)=1/4, roots[2];
```

(%o3) $\frac{1}{4}=\frac{1}{4}$

Отже, як бачимо, одержані корені задовольняють дане рівняння.

Розв'яжемо за допомогою функції solve() тригонометричне рівняння.

Приклад 1.18. Розв'язати рівняння $\cos x = \frac{\sqrt{3}}{2}$.

```
(%i2)  eq: cos(x)=sqrt(3)/2;
       koreni: solve(eq);
```

(eq) $\cos(x)=\frac{\sqrt{3}}{2}$

solve: using arc-trig functions to get a solution.
Some solutions will be lost.

(koreni) $[x=\frac{\pi}{6}]$

Під час розв'язування тригонометричних рівнянь в середовищі Maxima слід звернути увагу, що в розв'язку подається тільки один окремий корінь, але насправді рівняння має нескінченну множину коренів внаслідок періодичності тригонометричних функцій.

Функція solve() в програмі Maxima також використовується для розв'язування систем рівнянь. Рівняння системи та змінні системи подають у вигляді списку, тобто записують у квадратних дужках через кому. Розглянемо приклад.

Приклад 1.19. Розв'язати систему рівнянь $\begin{cases} x^2 + xy = 10, \\ xy + y^2 = 15. \end{cases}$

```
(%i1)  solve([x^2+x·y=10, x·y+y^2=15], [x,y]);
```

(%o1) $[[x=2, y=3], [x=-2, y=-3]]$

Одержали два розв'язки $(2; 3)$, $(-2; -3)$.

Задачі для самостійного розв'язування

1. Спростіть вираз, обчисліть значення виразу для даного a.

1.	$\left(1-\frac{1}{1-a}\right):\left(1-\frac{1-2a^2}{1-a}+a\right)$, $a=1{,}2$
2.	$\left(1+\frac{1}{a}+\frac{1}{a^2}+\frac{1}{a^3}\right):\left(1-\frac{1}{a}+\frac{1}{a^2}-\frac{1}{a^3}\right)$, $a=5{,}3$
3.	$\frac{a}{a^2-1}+\frac{a^2+a-1}{a^3-a^2+a-1}-\frac{2a^3}{a^4-1}$, $a=1{,}5$
4.	$1-\frac{a}{1+\frac{1}{1-\frac{1}{1+\frac{1}{a}}}}$, $a=8{,}5$
5.	$(32-a)\sqrt[4]{\frac{81}{(a-32)^4}}$, $a>32$
6.	$\frac{a}{\sqrt{2a}+2}+\frac{2}{\sqrt{2a}-a}-\frac{a+2}{\sqrt{2a}}$, $a=1{,}25$
7.	$\left(\frac{a^{0,5}+2}{a^{0,5}-2}+\frac{a^{0,5}-2}{a^{0,5}+2}-\frac{16}{a-4}\right)^4+a^{0,5}$, $a=2{,}25$
8.	$\frac{a-1}{a^{0.75}+a^{0.5}}\cdot\frac{a^{0.5}+a^{0.25}}{a^{0.5}+1}$, $a=16$
9.	$\left(\sqrt[3]{a}+2\right)\left(\sqrt[3]{a}-2\right)-(\sqrt[3]{a}+3)^2$, $a=125$
10.	$\frac{\sqrt{a}}{\sqrt{a}-1}-\frac{\sqrt[4]{a}}{\sqrt[4]{a}+1}$, $a=16$

2. Розкрийте дужки.

1.	$(x+2)(x-1)(2x-3)(2{,}4x-1{,}3)$
2.	$(2x+6)(x-5)(4x-6)(7{,}3x+2{,}2)$
3.	$(x^2-2x+3)(x^2+2x-3)(x^2-1{,}7)$
4.	$(a^2-2b+c)(a-3b+2)(0{,}5c-2)$
5.	$(x-1)(x-3{,}5)(x^4+x^3+x^2-2x+1)$
6.	$(0{,}2x+3)(2x-5)(0{,}7x-6)(2x^2+3)$
7.	$(3a^2+2{,}5)(a-5)(4a-6{,}8)(a+8{,}2)$
8.	$(m^2+5{,}5n+n^2)(m-5n+2)(0{,}1m-2)$
9.	$(a+3)(2a-7)(a^4-a^3+a^2-a+4{,}5)$
10.	$(7{,}2x+7{,}8)(5{,}7x+5)(3{,}4x-9)(7x+5)(x-2)$

3. Розкладіть на множники вираз.

1.	$x^4 - 5x^2 + 4$	6.	$x^8 + x^4 - 2$
2.	$x^4 + x^2 + 1$	7.	$x^4 + 5x^2 + 9$
3.	$4x^4 - 12x^2 + 1$	8.	$x^4 - 8x^2 - 2$
4.	$x^5 + x + 1$	9.	$9x^2 - 30xy + 16y^2$
5.	$x^4 + 4$	10.	$y^2 + 12y + 35$

4. Розкладіть дріб на прості множники.

1.	$\frac{-3x+1}{x^2+2x-3}$	6.	$\frac{3x+1}{x^2-2x-3}$
2.	$\frac{x^2+5}{x^3-x^2+3x-4}$	7.	$\frac{2x^2-4x+3}{x^4+4}$
3.	$\frac{x^2+2x-5}{x^3+5x^2+3x-4}$	8.	$\frac{-2x^2-3x+2}{x^3-x^2+x+2}$
4.	$\frac{-x^2-3x+3}{x^5+x+1}$	9.	$\frac{3x+1}{x^2-x-6}$
5.	$\frac{1-3x}{x^2+x-6}$	10.	$\frac{2x+8}{-x^2-x+5}$

5. Розв’яжіть рівняння, виконайте перевірку, проаналізуйте одержаний розв’язок.

1.	$\frac{2x^2-3x+1}{x^2+2x-3} = 1$	6.	$\frac{2x^2+3x+1}{x^2-2x-3} = 1$
2.	$x^2 + 3x + \frac{2}{x-1} = 4 - \frac{2}{1-x}$	7.	$\frac{2x^2-3x-2}{x-2} = 5$
3.	$x^2 - 2x + \frac{3}{x-3} = 3 - \frac{3}{3-x}$	8.	$\frac{-2x^2-3x+2}{x+2} = 5$
4.	$\frac{x+2}{x-1} + \frac{x+3}{x+1} + \frac{x+5}{1-x^2} = 0$	9.	$\frac{2x-1}{x+2} + \frac{3x+1}{x^2-x-6} = \frac{x-1}{x-3}$
5.	$\frac{2x+1}{x-2} + \frac{1-3x}{x^2+x-6} = \frac{x+1}{x+3}$	10.	$\frac{x-1}{x+2} + \frac{x+1}{x-2} + \frac{2x+8}{4-x^2} = 0$

6. Розв’яжіть рівняння, виконайте перевірку, проаналізуйте одержаний розв’язок.

1.	$x^4 + 5x^3 - x^2 - 17x + 12 = 0$	6.	$x^4 + 4 = 0$
2.	$x^5 + x + 1 = 0$	7.	$x^4 - 8x^2 - 2 = 0$

3.	$4x^4 - 12x^2 + 1 = 0$	8.	$x^4 + 5x^2 + 9 = 0$
4.	$x^4 + x^2 + 1 = 0$	9.	$y^2 + 12y + 35 = 0$
5.	$x^8 + x^4 - 2 = 0$	10.	$9x^2 - 30xy + 16y^2 = 0$

Контрольні питання

1. Яка функція призначена для обчислення значення виразу для деякого значення змінної?

a) ad(); b) define(); c) at(); d) subst().

2. Яка функція призначена для знаходження частки та остачі від ділення многочлена на многочлен?

a) div(); b) mod(); c) mode(); d) divide().

3. Який запис команди для спрощення виразу $\frac{x^2+2x+1}{x^2-1}$ правильний?

a) rat((x^2+2x+1)/(x^2-1));

b) ratsimp((x^2+2x+1)/(x^2-1));

c) rat((x^2+2*x+1)/(x^2-1));

d) rat([x^2+2*x+1]/[x^2-1]);.

4. Яка функція використовується, щоб видалити значення змінної з пам’яті?

a) assume(); b) kill(); c) cond(); d) del().

5. Яка функція використовується, щоб обмежити значення змінної відповідно деякій умові?

a) assume(); b) kill(); c) cond(); d) del().

6. Встановіть відповідність між функцією (1-4) та її призначенням (a-e):

1) radcan()	a) розклад виразу на множники;
2) fullratsimp()	b) перетворення раціонального виразу;
3) factor()	c) перетворення тригонометричного виразу;
4) partfrac()	d) перетворення показникового виразу;
	e) розклад дробу на прості дроби.

7. Вкажіть функцію для пошуку коренів рівняння за допомогою чисельних методів, використовуючи відокремлення змінної x на проміжку $[a; b]$.

a) allroots(); b) solve(); c) find_root(); d) findroot().

8. Яка функція повертає точні значення коренів у символьному вигляді?

a) allroots(); b) solve(); c) find_root(); d) findroot().

9. Вкажіть правильний запис команди для розв'язування системи рівнянь $\begin{cases} 3x - y = -5, \\ 12x + 15y = 7 \end{cases}$.

a) solve((3x-y, 12x+15y), [-5,7]);

b) solve([3*x-y, 12*x+15*y], [-5,7]);

c) solve([3x-y=-5, 12x+15y=7], [x,y]);

d) solve([3*x-y=-5, 12*x+15*y=7], [x,y]);.

10. В якому вигляді функція solve повертає знайдені корені рівняння?

a) у вигляді списку;

b) у вигляді множини;

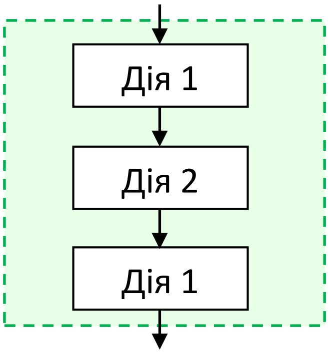
c) у вигляді рядка.

1.4 Програмування в системі Maxima

В середовищі програми Maxima є всі необхідні базові алгоритмічні конструкції для написання програм. Базові алгоритмічні структури – це конструкції, за допомогою яких створюється алгоритм для розв'язування задачі. Існує три базові (основні) алгоритмічні конструкції: лінійна, розгалуження, повторення.

Лінійний алгоритм (структура слідування) – це алгоритм, у якому команди виконуються послідовно, одна за одною, незалежно від вхідних даних і проміжних результатів.

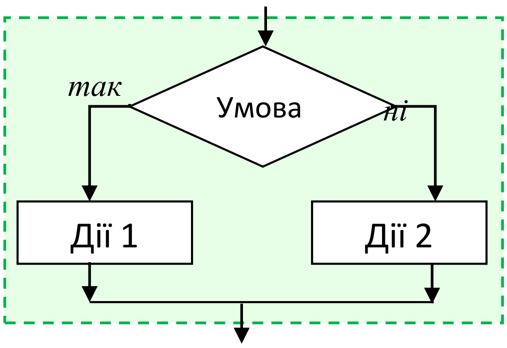
Наприклад, щоб обчислити площу прямокутника зі сторонами 5 см і 7 см потрібно записати послідовність команд:

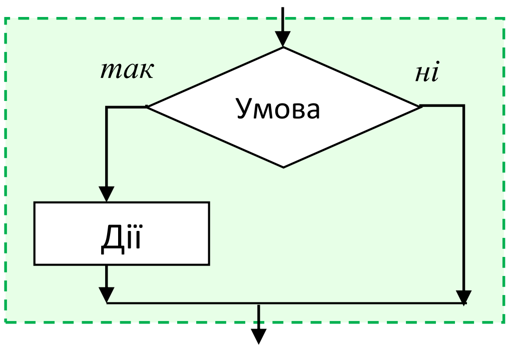



```
→   a: 5;
    b: 7;
    S: a·b; /·площа прямокутника·/
(a) 5
(b) 7
(S) 35
```

Алгоритм розгалуження – форма організації дій, коли залежно від виконання деякої умови відбувається одна або інша послідовність дій.

Неповне розгалуження – це розгалуження, в якому дії визначені тільки у разі виконання (невиконання) умови.





В програмі Maxima умовний алгоритм записується за допомогою операторів:

`if` *умова* `then` *дія1* `else` *дія2*

Тобто, якщо умова істинна, то виконується дія1, інакше – виконується дія2.

Неповний умовний оператор в Maxima має вигляд:

`if` *умова* `then` *дія*

Тобто, якщо умова істинна, то виконується деяка дія, якщо умова хибна – дія не виконується.

Для запису виразу умови використовують оператори порівняння >, <, ≥, ≤, =, # , а також логічні оператори `and, or, not`.

Приклад. Присвоїти змінній *n* більше з двох чисел *a* і *b*.

```
(%i6)  a: −12; b: 5;
       if a>b then n: a else n: b;
(a)    −12
(b)    5
(%o6)  5
```

Цей самий приклад можна виконати інакше:

```
(%i9)  a: −12; b: 5;
       n: if a>b then a else b;
(a)    −12
(b)    5
(n)    5
```

Приклад. Обчислити значення функції $f(x)$ в точках $a = -2$, $b = 3$

$$f(x) = \begin{cases} x^3, & \text{якщо } x < 0, \\ x^2 - 5, & \text{якщо } x \geq 0. \end{cases}$$

```
(%i14) a: −2$ b: 3$
       f(x):=if x<0 then x^3 else x^2−5$
       f(a);
       f(b);
(%o13) −8
(%o14) 4
```

Приклад 1.20. Написати програму, яка в залежності від введеного значення температури *t* повертає стан води, в якому вона знаходиться.

Для написання цієї програми використаємо вкладений оператор розгалуження, тобто після else, щоб виконати наступну дію, потрібно перевірити ще одну умову.

```
(%i11) t: -5$
       if t<0 then print("Лід")
              else if t<100 then print("Вода")
              else  print("Пара") $
```

Лід

В цій програмі також використали функцію `print(a)`, яка виводить на екран значення свого аргументу, тобто змінної a, наприклад. В наведеному вище прикладі аргументом функції print є рядкова величина у вигляді текстового повідомлення. Аналогічно використовується функція `display(a)`.

Приклад 1.21. Дано двозначне натуральне число. Написати програму для визначення, чи є число десятків більше за число одиниць чи вони рівні.

Для розв'язування цієї задачі дане двозначне число потрібно розкласти на десятки та одиниці. Для цього скористаємося функцією `mod(x,y)`, яка повертає остачу від ділення числа x на y.

```
(%i28) a: 28;
       od: mod(a,10); /·число одиниць·/
       des: (a-mod(a,10))/10; /·число десятків·/
       if od>des then print("Число одиниць більше за число десятків")
       else if od=des then print("Число одиниць дорівнює числу десятків")
       else ("Число одиниць менше за число десятків")$
(a)    28
(od)   8
(des)  2
       Число одиниць більше за число десятків
```

Алгоритм повторення (цикл) – це алгоритм, у якому деяка послідовність дій повторюється певну кількість раз.

Багаторазове виконання деякої частини алгоритму для різних значень деяких змінних називається циклічним обчислювальним процесом. Кожне таке окреме виконання повторюваних дій називають ітерацією циклу.

Для організації циклів необхідно задати початкове значення змінної, що визначає цикл (параметр циклу), змінювати цю змінну перед кожним повторенням циклу і перевіряти умову продовження (закінчення циклу).

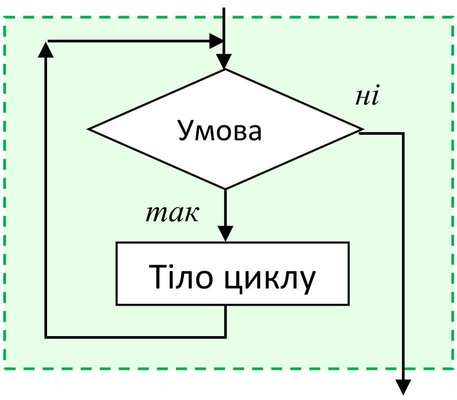
Розрізняють три типи циклів за способом кількості повторень.

1. Цикл з параметром (цикл «для») – цикл, у якому кількість повторень наперед відома. Синтаксис циклу з параметром в програмі Maxima:

`for` *змінна: початкове_значення* `step` крок `thru` *кінцеве_значення*
`do` *тіло циклу*;

Змінну циклу ще називають параметром або лічильником циклу. Конструкції `step` та `thru` не є обов'язковими. Якщо опустити конструкцію step, то за замовчуванням крок зміни параметру циклу буде дорівнювати 1.

2. Цикл із заданою умовою продовження роботи (цикл «поки», цикл з передумовою).

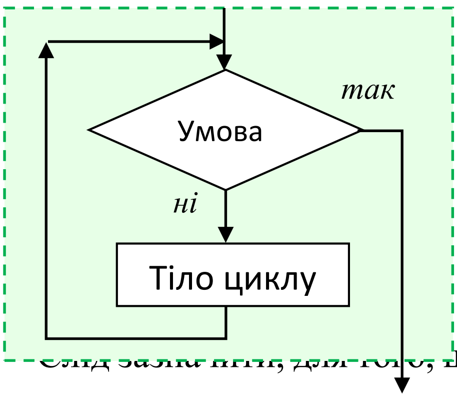


Синтаксис в програмі Maxima циклу з передумовою:

`while` умова `do` тіло_циклу;

Ключове слово while вказує на те, що команди тіла циклу будуть виконуватися поки виконується умова.

3. Цикл із заданою умовою закінчення роботи (цикл «до», цикл з післяумовою). Синтаксис в програмі Maxima циклу з передумовою:



`unless` умова `do` тіло_циклу;

Ключове слово unless вказує на те, що команди тіла циклу будуть виконуватися поки умова не виконується.

Слід зазначити, для того, щоб у тілі умовної конструкції або циклу потрібно виконати кілька команд, ці команди записують в круглих дужках через кому. Такий запис називають блоком команд. Наприклад, блок команд для обміну місцями змінних a і b виглядає так: `(t:a, a:b, b:t)`.

Приклад 1.22. Задати перші п'ять членів арифметичної прогресії, в якій перший член дорівнює –3,5 і різниця 1,5.

```
(%i23) a:-3.5$ d: 1.5$ /*задаємо перший член прогресії та різницю*/
       n:1$ /*задаємо номер члена арифметичної прогресії*/
       while n<=5 do (print("a",n,"=",a), a: a+d, n: n+1);
a 1 = -3.5
a 2 = -2.0
a 3 = -0.5
a 4 = 1.0
a 5 = 2.5
(%o23) done
```

Оскільки в задачі задано кількість членів арифметичної прогресії, то дану задачу також можна розв'язати, використовуючи цикл з параметром:

```
(%i27) a:-3.5; d: 1.5;
       display(a);
       for i:2 thru 5 do (a:a+d, display(a));
(a)    -3.5
(d)    1.5
       a=-3.5
(%o26) done
       a=-2.0
       a=-0.5
       a=1.0
       a=2.5
(%o27) done
```

Приклад 1.23. Обчислити суму натуральних чисел від 1 до 15.

```
(%i31) s:0$ /*змінна для обчислення суми*/
       i: 1$ /*лічильник циклу*/
       while i<=15 do (s: s+i, i: i+1) $
       print("сума чисел від 1 до 15 :  ", s);
       сума чисел від 1 до 15 :    120
(%o31) 120
```

Задачі для самостійного розв'язування

1. Обчислити значення функції $f(x)$ в точках a, b, c :

1.	$a=-8{,}4$, $b=-5$, $c=1{,}5$	$f(x)=\begin{cases} x^3-x^2-1, & \text{якщо } x<-5, \\ x^2-5, & \text{якщо } x=-5, \\ \sqrt{x^2+7}, & \text{якщо } x>-5. \end{cases}$

2.	$a = -6{,}1,$ $b = -0{,}3,$ $c = 5{,}96$	$f(x) = \begin{cases} -x^2 + 3\lvert x\rvert - 1, & \text{якщо } x < -5, \\ x^3 + 1, & \text{якщо } -5 \le x < 5, \\ \sqrt{x-2}, & \text{якщо } x \ge 5. \end{cases}$
3.	$a = -12,$ $b = -2,$ $c = 3{,}65$	$f(x) = \begin{cases} x^2 + 1, & \text{якщо } x < -2{,}5, \\ \sin x, & \text{якщо } -2{,}5 \le x < 1, \\ \sqrt[3]{x-2}, & \text{якщо } x \ge 1. \end{cases}$
4.	$a = -7{,}7,$ $b = -0{,}4,$ $c = 9{,}5$	$f(x) = \begin{cases} -x^2 + 3\lvert x\rvert - 1, & \text{якщо } x < -5, \\ x^3 + 1, & \text{якщо } -5 \le x < 5, \\ \sqrt{x-2}, & \text{якщо } x \ge 5. \end{cases}$
5.	$a = -6,$ $b = 1{,}88,$ $c = 3{,}75$	$f(x) = \begin{cases} \sin x - x^2, & \text{якщо } x < -1, \\ \sin x - x, & \text{якщо } -1 \le x < 2{,}5, \\ \sqrt{x^2 - 1}, & \text{якщо } x \ge 2{,}5. \end{cases}$
6.	$a = -1,$ $b = 1{,}35,$ $c = 9{,}25$	$f(x) = \begin{cases} \lvert x^2 + 3\lvert x\rvert - 1\rvert, & \text{якщо } x < 5, \\ x^3 \cos x, & \text{якщо } 5 \le x < 8{,}5, \\ \dfrac{2}{x}, & \text{якщо } x \ge 8{,}5. \end{cases}$
7.	$a = -7,$ $b = 2{,}1,$ $c = 6{,}5$	$f(x) = \begin{cases} \sin 2x + \cos 3x, & \text{якщо } x < -3{,}5, \\ x^2 - 5, & \text{якщо } -3{,}5 \le x < 4{,}5, \\ \sqrt{3x}, & \text{якщо } x \ge 4{,}5. \end{cases}$
8.	$a = -5{,}3,$ $b = -0{,}5,$ $c = 5{,}8$	$f(x) = \begin{cases} 2\sqrt{-x}, & \text{якщо } x < -5, \\ e^{3x}, & \text{якщо } -5 \le x < 5, \\ \dfrac{1}{x+2}, & \text{якщо } x \ge 5. \end{cases}$
9.	$a = -8,$ $b = -5,$ $c = 1{,}5$	$f(x) = \begin{cases} 5x^2 - 2\lvert x\rvert - 1, & \text{якщо } x < -1, \\ e^x - 1, & \text{якщо } -1 \le x < 4{,}5, \\ 8x\sqrt{x-2}, & \text{якщо } x \ge 4{,}5. \end{cases}$
10.	$a = -7{,}5,$ $b = -5,$ $c = 1{,}5$	$f(x) = \begin{cases} \dfrac{2}{x^2 - 5}, & \text{якщо } x < -7, \\ \sin x^2 + \cos x, & \text{якщо } -7 \le x < -3, \\ x^3 \cos x, & \text{якщо } x \ge -3. \end{cases}$

2. Розв’яжіть задачу, використовуючи оператор розгалуження.

1.	Дано натуральне тризначне число. Написати програму, яка перевіряє, чи є дане число паліндромом (десятковий запис числа читається однаково зліва направо і навпаки).
2.	Дано натуральне число. Написати програму, яка перевіряє, чи ділиться число на 8.
3.	Дано тризначне натуральне число. Перевірити, чи є серед його цифр нуль.
4.	Дано натуральне число. Написати програму, яка визначає максимальну цифру цього числа.
5.	Написати програму, яка за номером місяця визначає пору року.
6.	Написати програму, яка за введеною годиною доби виводить відповідне привітання: доброго ранку, добрий день, добрий вечір.
7.	Дано натуральне число. Якщо число парне, піднести його до квадрату. Якщо непарне – знайти корінь квадратний.
8.	Дано натуральне число. Якщо число просте, вивести відповідне повідомлення. Якщо число не є простим, записати його у вигляді добутку простих множників.
9.	Дано двозначне натуральне число. Написати програму, яка округлить число до десятків зі збільшенням на одиницю числа десятків, якщо число одиниць ≥ 5.
10.	Дано натуральне число. Написати програму, яка визначає мінімальну цифру цього числа.

3. Розв’яжіть задачу, використовуючи оператор повторення.

1.	Дано натуральне число. Обчислити кількість цифр цього числа.
2.	Знайти всі двозначні числа, в яких є цифра 2 і число ділиться на 3.
3.	Знайти всі тризначні числа, у яких всі три цифри різні.
4.	Обчислити суму кубів перших 12 натуральних чисел.
5.	Обчислити добуток квадратів перших 10 натуральних чисел.

6.	Перший член геометричної прогресії дорівнює 2,8. Наступний член у півтора рази більший. Знайти перших сім членів геометричної прогресії.
7.	Знайти всі тризначні числа, які діляться на 7, обчислити їх кількість.
8.	Знайти всі чотиризначні числа, які діляться на 12, обчислити їх кількість.
9.	Знайти всі тризначні числа, сума цифр яких дорівнює 9, обчислити їх кількість.
10.	Натуральне число з $\mathbf{n}$ цифр є числом Армстронга, якщо сума його цифр, піднесених до $\mathbf{n}$-го степеня, дорівнює самому числу (наприклад, $153 = 1^3 + 5^3 + 3^3$. Отримати всі числа Армстронга, що складаються з трьох і чотирьох цифр.

4. Якщо числа n і n+2 є простими, то вони називаються простими числами-близнюками. Знайдіть всі прості близнюки, що не перевищують 100. Порахуйте кількість пар простих близнюків, що не перевищують 1000.

5. Розглянемо четвірки простих чисел, які відрізняються лише останньою цифрою (очевидно, це будуть числа виду 10n+1, 10n+3, 10n+7, 10n+9). Знайдіть всі такі четвірки, які не більші за 1000.

6. Число називається досконалим, якщо сума всіх його дільників, окрім самого числа, дорівнює цьому числу, тобто divsum(n)=2n. Знайдіть перші чотири досконалі числа.

7. Два числа називаются дружніми, якщо сума всіх дільників кожного з них (окрім самих цих чисел) дорівнює другому числу, наприклад, 220 и 284. Знайдіть всі пари дружніж чисел, не більших за 10000.

8. Гіпотеза Гольдбаха (не доведена і не спростована до цих пір) говорить, що будь-яке парне число (крім 2) можно подати у вигляді сумми двох простих чисел. Перевірте гіпотезу Гольдбаха для всіх чисел, не більших 106.

Контрольні питання

1. Яким оператором в Maxima виконується перевірка рівності двох значень?

a) = =   b) equal   c) =   d) :=

2. Яким оператором в Maxima виконується перевірка нерівності двох значень?

a) <>   b) !=   c) /=   d) #

3. Яке значення буде результатом виразу при x = 4?

```
if x > 0 then sqrt(x) else x^2
```

a) 4   b) – 4   c) 2   d) – 2

4. Який оператор використовують, щоб записати блок команд в алгоритмах розгалуження і повторення?

a) ()   b) {}   c) block   d) <>

5. Яка буде вихідна послідовність чисел при виконанні наступного коду?

```
for i: 1 thru 5 do (
   if evenp(i) then print(i^2)
   else print(-i^2));
```

a) -1, 4, -9, 16, -25

b) 1, 2, 3, 4, 5

c) 1, 4, -9, 16, -25

d) -1, -2, -3, -4, -5

6. Який оператор не можна використати в алгоритмах з повторенням, якщо наперед невідомо кількість повторень?

a) repeat   b) while   c) for   d) unless

7. Який оператор використовується в алгоритмах з повторенням, якщо тіло циклу виконується, коли умова хибна?

a) repeat   b) while   c) for   d) unless

8. Яка команда використовується для виведення даних?

a) output()

b) input()

c) print()

d) displaying()

9. Що визначає оператор thru в циклі for?

a) крок зміни параметра циклу

b) поточне значення змінної параметра циклу

c) такий синтаксис циклу

d) останнє значення змінної параметра циклу

10. Скільки разів в конструкції розгалуження може використовуватися вкладене розгалуження?

a) тільки один раз

b) не може використовуватися

c) двічі

d) багато разів.

2 Побудова графічних зображень в Maxima

Для створення графічних зображень різних об’єктів в систему комп’ютерної математики Maxima вбудовано потужний інструментарій, який може широко застосовуватися в процесі графічного моделювання та розв’язування навчально-дослідницьких задач.

Побудова графічних зображень математичних об’єктів в Maxima здійснюється під управлінням програми Gnuplot, яка запускається автоматично під час виконання процедур створення графічних зображень, або під управлінням пакету Openmath, який розробляється разом з Maxima. При цьому під час виконання графічних процедур відкривається окреме вікно Gnuplot або Openmath з побудованим графічним об’єктом. В даних вікнах можна виконати певний набір операцій щодо перетворення та перегляду створеного графічного об’єкта (зміна масштабу, копіювання до буфера обміну), зображення ліній сітки графічної області, використання Gnuplot-консолі та ін.. Для продовження роботи в середовищі Maxima необхідно згорнути або ж закрити вікно Gnuplot.

Якщо до назв графічних процедур дописуються літери «wx», то графічний об’єкт відтворюється на робочому аркуші програми з урахуванням роздільних характеристик екрану. При цьому графіки залишаються видимими протягом всієї сесії роботи з Maxima. В контекстному меню графіка доступні послуги його копіювання до буфера обміну та збереження графіка у вигляді файлу.

Для створення графічних зображень в Maxima є кілька бібліотек. За замовчуванням використовується вбудована в систему бібліотека графічних процедур Plot. Для зображення та налаштування складних графічних об’єктів більш зручною є бібліотека Draw, яка додатково при потребі завантажується за командою `load(draw)$`.

Якщо функція задана рівнянням $y = f(x)$, розв’язаним відносно залежної змінної y, то кажуть, що функція задана у явній формі або є явною.

Під неявним заданням функції розуміють задання функції у вигляді рівняння $F(x, y) = 0$, не розв’язаного відносно залежної змінної. Це рівняння задає функцію лише тоді, коли множина пар чисел $(x; y)$, які є розв’язком даного рівняння, така, що будь-якому числу x_0 у цій множині відповідає не більше однієї пари $(x_0; y_0)$ з першим елементом x_0.

Рівняння $F(x,y) = 0$ називається *рівнянням лінії* l, яка задана на площині відносно певної системи координат, якщо це рівняння задовольняють координати x і y кожної точки лінії l і не задовольняють координати x і y жодної точки, яка не лежить на цій лінії.

Одну й ту саму лінію можна задати різними рівняннями. Вигляд рівняння лінії залежить від вибору системи координат. Рівняння лінії змінюється як при переході від однієї декартової системи до іншої, тобто при перетворенні координат, так і при переході від декартових до будь-яких інших координат.

Параметричні рівняння лінії.

Нехай залежність між змінними x і y виражена через третю змінну t, тобто

$$x = x(t),\ y = y(t).$$

Змінна t називається параметром і визначає положення точки $(x; y)$ на площині. Наприклад, якщо $x = 2t + 3, y = t^2 + 1$, то значенню параметра $t = 3$ відповідає на площині точка $(9; 10)$, тому що $x = 2 \cdot 3 + 3 = 9$, $y = 3^2 + 1 = 10$.

Якщо t змінюється, то точка на площині переміщується, описуючи деяку лінію l. Такий спосіб задання лінії називається параметричним, а рівняння $x = x(t),\ y = y(t)$ – *параметричними рівняннями лінії* l.

Можна вважати, що звичайний графік функції $y = f(x)$ також є параметричним, якщо записати його у вигляді $\begin{cases} y = f(t), \\ x = t \end{cases}$. Всяка параметрично задана функція визначає на площині Oxy деяку криву лінію, проте не всяка параметрично задана крива визначає функцію.

Рівняння лінії в полярних координатах.

Рівняння $\Phi(\rho, \varphi) = 0$ називається рівнянням лінії *l* в полярних координатах, або полярним рівнянням, якщо його задоволняють полярні координати ρ і φ будь-якої точки лінії *l* і не задоволняють координати жодної точки, яка не лежить на цій лінії

2.1 Графічний інтерфейс Plot

2.1.1 Команди інтерфейсу Plot для побудови графічних зображень

Графічний інтерфейс Plot має фактично дві основні команди (wx)plot2d та (wx)plot3d, за допомогою яких будуються двовимірні та тривимірні графіки функцій, ліній та поверхонь, які задані явно, точками або параметрично. Правила запису основних plot-процедур побудови графічних зображень:

`plot2d(f(x),[x,x1,x2],opts)` – побудова графіка явно заданої функції *f(x)* на проміжку $x1 \le x \le x2$. Проміжок вздовж осі Oy вказувати необов'язково.

`plot2d([discrete,[x1,x2,…],[y1,y2,…]],opts)` – побудова графіка, що заданий точками з координатами $(x_i; y_i)$, $i = 1,2,...$ За замовчуванням точки на графіку з'єднуються відрізками.

`plot2d([discrete,[[x1,y1],[x2,y2],…]],opts)` – спосіб побудови графіка, що заданий точками з координатами $(x_i; y_i)$, $i = 1,2,...$, коли координати точок подаються у вигляді вкладеного списку. За замовчуванням точки на графіку з'єднуються відрізками.

`plot2d([parametric,x(t),y(t),[t,t1,t2],opts)` – побудова лінії, заданої параметрично $\{x(t), y(t)\}$ з параметром t, $t1 \le t \le t2$. Якщо параметр незаданий, то він набуває значень за замовчуванням, які визначені в set_plot_option.

`plot3d(f(x,y),[x,x1,x2],[y,y1,y2],opts)` – побудова графіка явно заданої функції *f(x,y)* від двох змінних в тривимірній декартовій системі координат *Oxyz*, $x1 \le x \le x2$, $y1 \le y \le y2$.

`plot3d([x(u,v),y(u,v),z(u,v)],[u,u1,u2],[v,v1,v2],opts)` – побудова поверхні, заданої параметрично $\{x(u,v), y(u,v), z(u,v)\}$ з параметрами $u1 \le u \le u2$, $v1 \le v \le v2$.

`contour_plot(f(x,y),[x,x1,x2],[y,y1,y2],opts)` – побудова контурних ліній явно заданої функції *f(x,y)* в проєкції на площину *Oxy* на проміжках $x1 \le x \le x2$, $y1 \le y \le y2$.

```
load(implicit_plot)$
```

`implicit_plot(f(x,y),[x,x1,x2],[y,y1,y2],opts)` – побудова контурних ліній неявно заданої функції *f(x,y)* в проекції на площину *Oxy* на проміжках $x1 \le x \le x2$, $y1 \le y \le y2$.

Щоб спростити введення команд побудови графіків та їх налаштування в Maxima передбачено меню *Plot*. Виконавши послугу Креслення/Двовимірний кграфік (*Plot/Plot2d…)*, відкривається вікно *Графік 2D* (рис.2 1) з формою, яку необхідно заповнити.

В першій стрічці вводять вираз функції або її назву, якщо функція задана раніше. Щоб зобразити графіки декількох функцій на одному рисунку, їх вирази або імена відокремлюються комами.

За допомогою кнопки *Додатково* (рис.2 1) можна перейти до побудови графіка функції, яка задана параметрично або точками (дискретний графік).

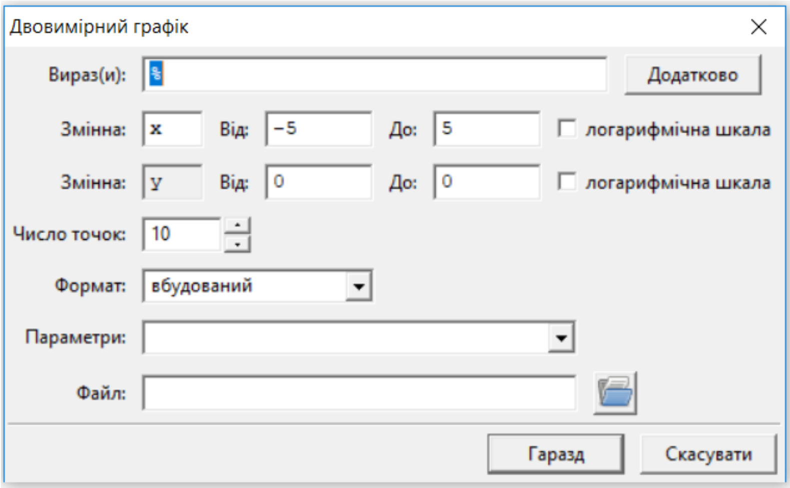


Рис. 2.1 Вікно програми Maxima для побудови графіків за допомогою функцій Plot

В другому та третьому рядках задають діапазон змінних вздовж осей.

Для забезпечення більшої гладкості ліній графіка збільшують число точок.

В полі *Формат* вибирають один з методів побудови графіка:

1) *убудований* – графік зображується в тому ж вікні, що і командний рядок (аналогічно до того, якщо до процедур *plot* дописати *wx*);

2) *за замовчанням* – побудова виконується за допомогою *gnuplot*;

3) *gnuplot* – графік будується в окремому вікні *Gnuplot*, де можна за допомогою кнопок масштабувати, зберігати графік, налаштовувати деякі опції, скопіювати до буфера обміну, роздрукувати та ін. (рис. 2.2);

4) *xmaxima (openmath)* – графік відкривається у вікні *Xmaxima*:*Plot2d*, За допомогою послуг меню даного вікна можна налаштувати опції графічного зображення (шрифти, стилі ліній, колір тощо), масштабувати, зберегти графік у вигляді файлу.

В рядку *Параметри* вибирають налаштування зображення системи координат:

`set zeroaxis` – осі проходять через початок координат;

`set grid` – зображення координатної сітки;

`set size ratio1` – вирівнювання масштабів вздовж осей координат (наприклад, щоб коло було круглим на рисунку, а не овальним).

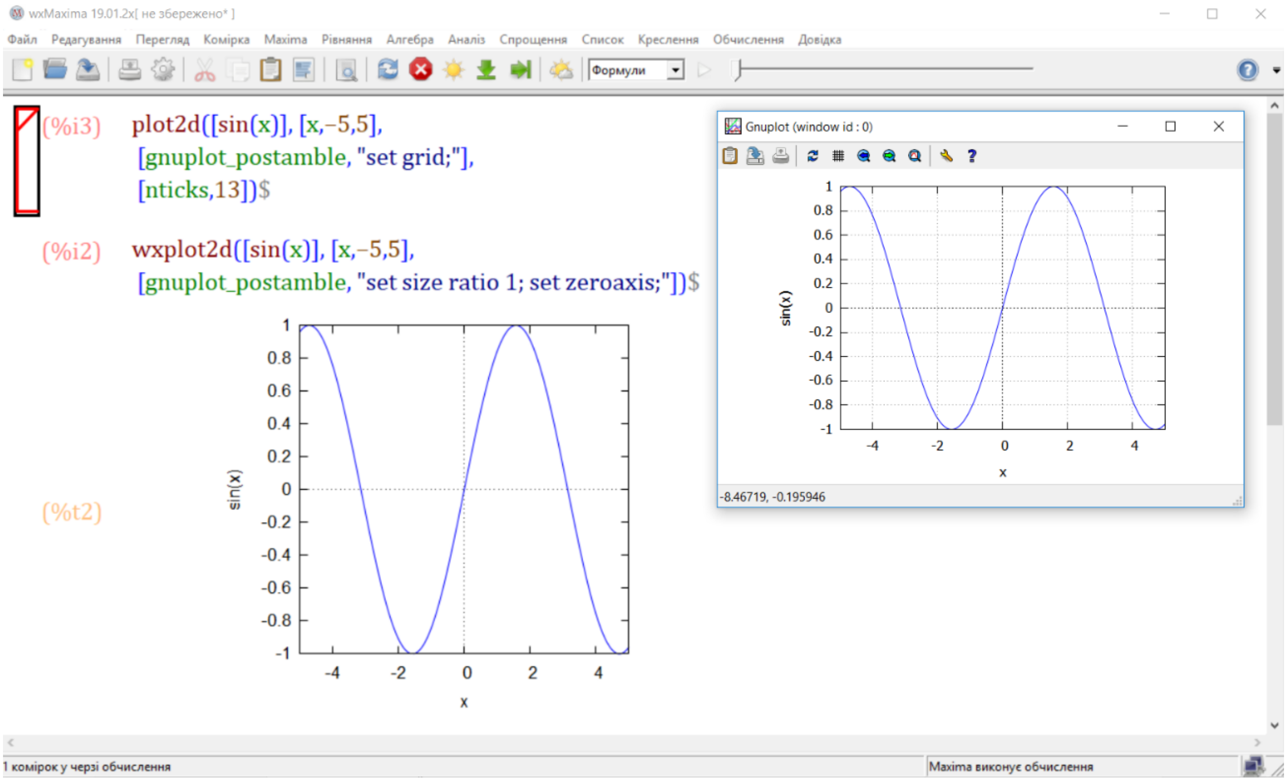


Рис. 2.2 Графік функції y=sin(x) у форматі вбудований та gnuplot.

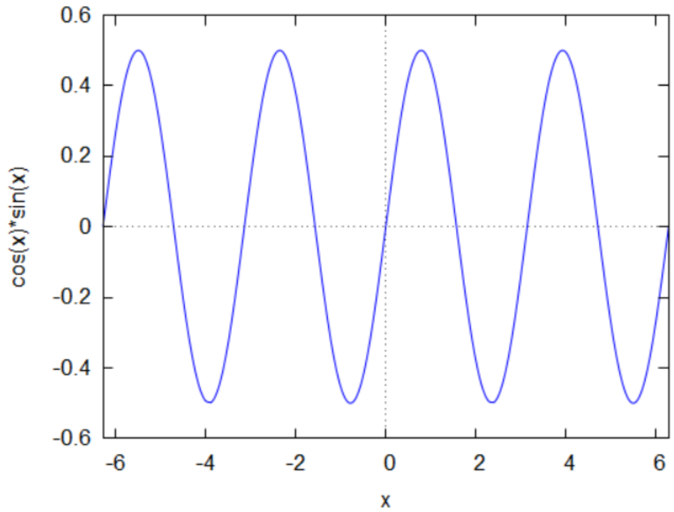
Замість виразів функцій, рівнянь ліній та списків координато точок в графічних процедурах можна вказувати їх імена, якщо вони попередньо були означені. Це значно спрощує і полегшує запис графічних процедур.

Використовуючи процедури plot2d, на одному і тому ж рисунку можна зобразити декілька графіків різного типу. Але для створення графічних зображень тривимірних об'єктів в інтерфейсі Plot пропонуються досить обмежені можливості. Зокрема за допомогою процедур plot3d можна створювати тривимірні графіки, які, по-перше, задані лише явно або параметрично, по-друге, на одному і тому ж рисунку можна зобразити не більше однієї поверхні. Розглянемо приклади використання процедур plot2d та plot3d.

Приклад 2.1. Побудувати графік функції $y = \sin(x) \cdot \cos(x)$ в межах від -2π до 2π вздовж осі Ox; інтервал вздовж осі Oy розраховується автоматично:

```
(%i1)  wxplot2d(sin(x)·cos(x), [x, −2·π,2·π]);
```

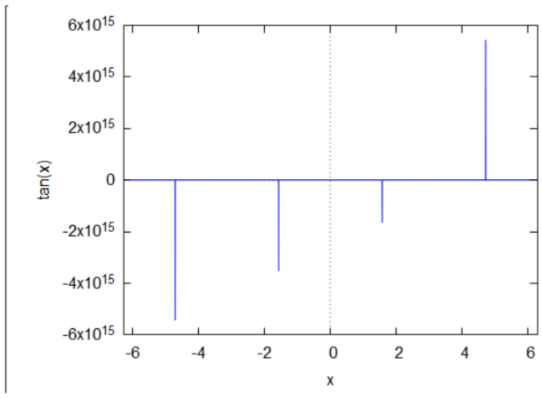
(%t1)



Праклад 2.2. Побудувати графік функції $y = tg(x)$ в межах від -2π до 2π вздовж осі Ox.

```
(%i2)  wxplot2d(tan(x), [x, −2·π,2·π]);
```

(%t2)



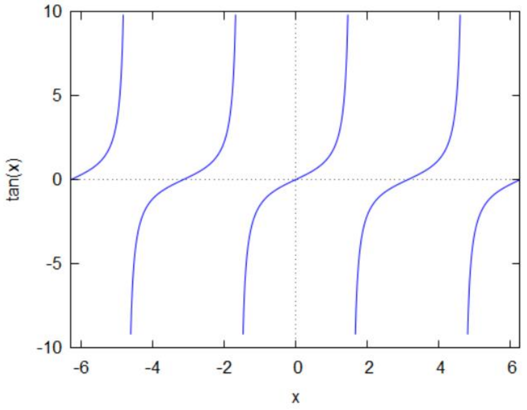
(%o2)

В результаті одержали незрозумілий графік. Це пояснюється тим, що функція тангенса невизначена в точках $x = \frac{\pi}{2} \pm \pi$. Оскільки в опціях не вказано інтервал побудови графіка вздовж осі Oy, то за замовчуванням в Maxima виконуються обчислення з точністю 16 знаків. Однак, якщо в опціях графічної процедури вказати відносно невеликий проміжок вздовж осі Oy, одержимо коректний графік функції $y = tg(\ x)$.

(%i3) wxplot2d(tan(x), [x, −2·π,2·π], [y,−10,10]);

plot2d: some values were clipped.

(%t3)

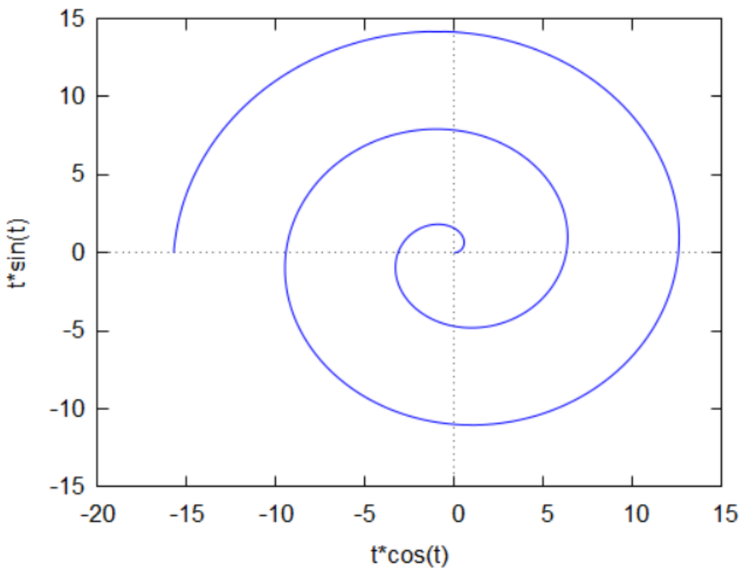


(%o3)

Приклад 2.3. Побудувати графік лінії, що задана параметрично $x(t) = t \cdot \cos(t)$, $y(t) = t \cdot \sin(t)$ з параметром $t \in [0;\ 5\pi]$; для того, щоб досягнути гладкості кривої на рисунку, слід збільшити число точок на графіку, користуючись опцією `nticks`:

(%i2) wxplot2d([parametric, t·cos(t), t·sin(t), [t,0,5·π]], [nticks,100]);

(%t2)



(%o2)

Приклад 2.4. Побудувати на одному рисунку два графіка, які задані точками.

Попередньо визначаємо список значень координат x_i іменем `xlist` та список значень координат y_i іменем `ylist`. Тоді в процедурі wxplot2d замість списків координат точок по осі *Ox* та *Oy* записуємо їх імена.

Для побудови кількох графіків на одному рисунку відповідні функції в процедурі wxplot2d описуємо в квадратних дужках через кому, як елементи списку.

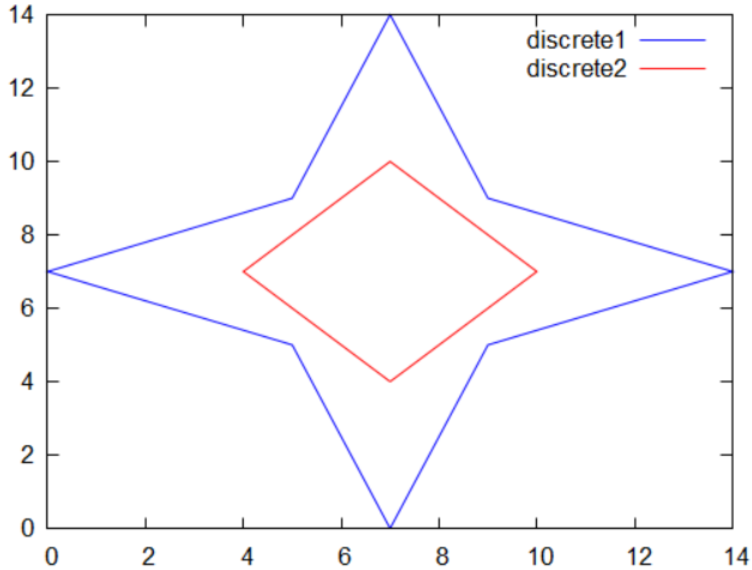
За замовчуванням точки з'єднуються відрізками.

```
(%i3)    xlist: [7,5,0,5,7,9,14,9,7];
(xlist)  [7,5,0,5,7,9,14,9,7]

(%i4)    ylist: [0,5,7,9,14,9,7,5,0];
(ylist)  [0,5,7,9,14,9,7,5,0]

(%i5)    wxplot2d([[discrete, xlist, ylist],
                   [discrete,[[7,4],[4,7],[7,10],[10,7],[7,4]]]]);
```

(%t5)



(%o5)

Приклад 2.5. Побудуємо три графіки ліній на одному рисунку, що задані явно, точками, параметрично:

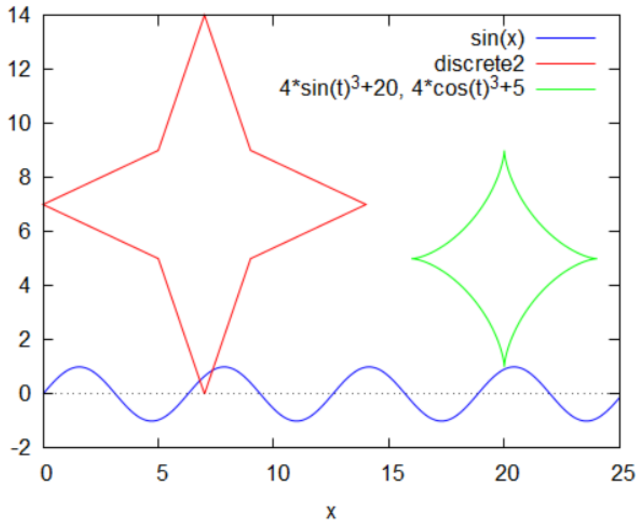
1) $y = \sin(x)$;

2) список координат xlist, ylist з прикладу 3;

3) $x(t) = 20 + 4\sin^3 t$, $y(t) = 5 + 4\cos^3 t$, $t \in [0; 2\pi]$.

Відповідні вирази задання ліній записуються в процедурі wxplot2d в квадратних дужках через кому як елементи списку:

(%i6) wxplot2d([sin(x), [discrete, xlist, ylist],
[parametric, 20+4·(sin(t))^3, 5+4·(cos(t))^3, [t,0,2·π]]],
[x,0,25], [nticks, 100]);

(%t6) 

(%o6)

Приклад 2.6. Побудуємо графік функції від двох змінних:

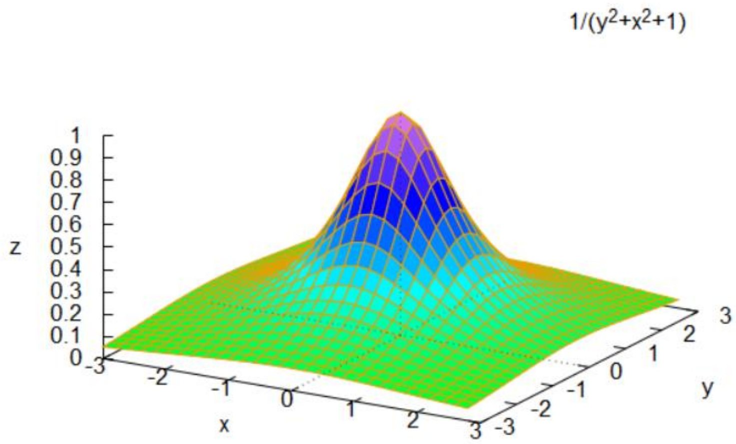
$$z = f(x, y) = \frac{1}{y^2 + x^2 + 1}$$

Попередньо позначимо через z вираз функції. Тоді в процедурі wxplot3d використовуємо замість виразу функції ідентифікатор z.

(%i7) z: 1/(x^2+y^2+1);

(z) $\frac{1}{y^2+x^2+1}$

(%i8) wxplot3d(z, [x,−3,3], [y,−3,3]);

(%t8) 

Приклад 2.7. Побудувати поверхню, що задана параметрично рівняннями:

$$x(u,v) = \sin(u),$$
$$y(u,v) = v,$$
$$z(u,v) = \cos(u) + v.$$

Попередньо присвоїмо вирази параметричних рівнянь змінним fx, fy, fz відповідно для більш простого запису процедури wxplot3d.

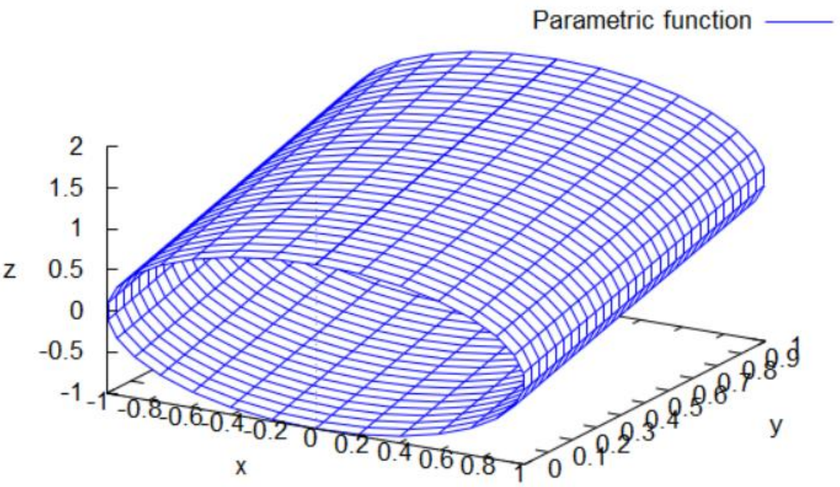
Кольорового заповнення поверхні можна позбутися, встановивши значення false для параметра palette.

В результаті одержуємо циліндричну поверхню.

```
(%i11) fx: sin(u); fy: v; fz: cos(u)+v;
(fx)   sin(u)
(fy)   v
(fz)   v+cos(u)

(%i12) wxplot3d([fx, fy, fz], [u, 0, 2·π], [v, 0, 1],
            [palette, false]);
```

(%t12)

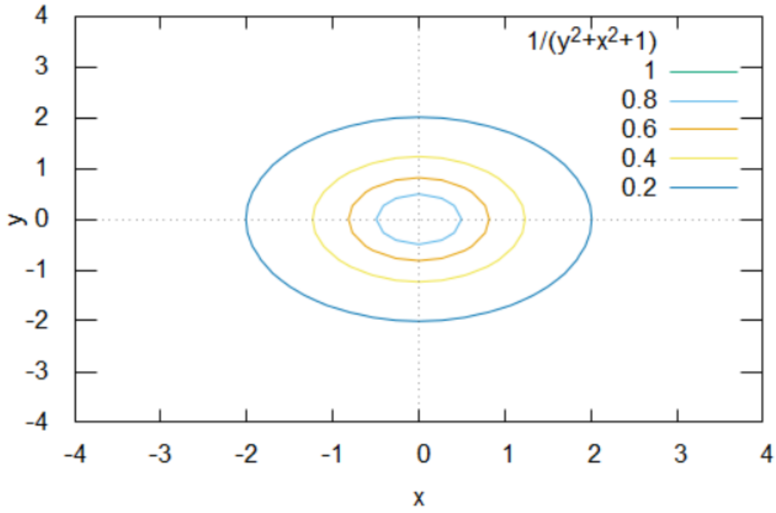


Приклад 2.8. Побудуємо контурні лінії рівнів в проекції на площину *Oxy* для функції від двох змінних $f(x,y) = \frac{1}{y^2+x^2+1}$.

Попередньо позначимо через z вираз функції. Тоді в процедурі `wxcontour_plot` використовуємо замість виразу функції ідентифікатор z.

(%i13) wxcontour_plot(z, [x, −4, 4], [y, −4, 4]);

(%t13)



Для налаштування різних властивостей зображення графічних об'єктів та графічного поля використовують спеціальні опції графічного інтерфейсу Plot.

2.1.2 Опції графічного інтерфейсу Plot

За допомогою спеціальних додаткових опцій (параметрів) можна змінювати колір, тип ліній, розміри, мітки, формати виведення графіка тощо. Опції записуються послідовно і відокремлюються комою. Як правило, опція складається з двох елементів: `[name_opt,v]`. Перший елемент – це ім'я параметра, додаткові елементи – це пов'язані з іменем значення. Наприклад, опція для визначення кольору лінії має ім'я `color`, що може набувати значень `blue, red, green` тощо. Опції можна визначати безпосередньо в процедурі побудови графіка або наперед задати командою `set_plot_option([name_opt,v1,v2,…])`, щоб користуватися ними за замовчуванням. Щоб переглянути встановлені опції, використовують команду `get_plot_option(name_opt)`:

Виклик команди `set_plot_option()` без аргументів надає перегляд всіх опцій, які встановлені за замовчуванням:

set_plot_option();

[[*color* , *blue* , *red* , *green* , *magenta* , *black* , *cyan*] , [*plot_format* , *gnuplot*] , [*grid* , 30 , 30] , Розглянемо основні опції.

`grid2d` – відображення ліній сітки на координатній площині; дана опція застосовується без прямих дужок.

`[box, false]` – графічне поле без рамки; за замовчуванням параметр box має значення true, тобто графічне поле інтерфейсу wxplot зображується всередині рамки, на якій вказуються підписи значень вздовж осей Ox та Oy.

`[axes, solid]` – координатні вісі у вигляді суцільної лінії; за замовчуванням координатні вісі є штриховою лінією.

`[yx_ratio, 1]` – вирівнювання масштабів вздовж осей координат.

`[xtics,x1,step,xn]` – підписи вздовж осі Ox від `x1` до з кроком `step`.

`[ytics,y1,step,yn]` – підписи вздовж осі Oy `y1` до `yn` з кроком `step`.

`[label, ["text_1",x1,y1], ["text_2",x2,y2],…]` – написи text_1, text_2, що розташовані в графічному полі в точках з координатами *(x1;y1)*, *(x2;y2)* відповідно.

`[title, "text"]` – заголовок `text` графічного зображення.

`[nticks, n]` – задає число, на яке буде поділений інтервал змінної. Якщо крива лінія виглядає неплавною, то необхідно збільшити значення даної опції. За замовчуванням `[nticks, 29]`.

`[style,type_1,type_2,…]` – стиль лінії графіка функції.

За допомогою параметра `style` налаштовують вигляд лінії, зокрема її тип, розмір, колір та інші особливі налаштування в залежності від типу ліній. Для визначення вигляду певної лінії опцію стилю записують у вигляді:

`[style,type_1,type_2,…]`, де `type_1`, `type_2`, … послідовно вказують на тип лінії, розмір, колір та інші особливі налаштування в залежності від типу лінії.

Для кількох ліній до стилю можна включати послідовний список стилів кожного графічного об'єкта: `[style,[style_1],[style_2],…]`.

Основні типи ліній:

`lines` – суцільна лінія, застосовується з однією або двома додатковими опціями: перша – ширина лінії (число), друга – колір (числовий код або назва).

За замовчуванням в Maxima встановлені такі коди і назви кольорів: 1 – blue, 2 – red, 3 – green, 4 – magenta, 5 – black, 6 – cyan:

```
(%i41) get_plot_option(color);
(%o41) [color, blue, red, green, magenta, black, cyan]
```

`points` – точки, застосовується з однією, двома або трьома додатковими опціями: перша – радіус точки (число), друга – колір (код або назва), третя – тип точки (код або назва). Дана опція застосовується для зображення графіка функції, що задана точками.

За замовчуванням в Maxima встановлені такі типи точок: 1 – bullet (кружечок), 2 – box (квадрат), 3 –triangles (трикутник), 4 – plus (знак +), 5 – times (знак х), 6 – asterisk (зірочка):

```
get_plot_option(point_type);
[point_type, bullet, box, triangle, plus, times, asterisk]
```

`dots` – лінія у вигляді крапок, може застосовуватися з параметром налаштування кольору.

`linespoints` – суцільна лінія разом з точками, застосовується з чотирма параметрами: ширина лінії, розмір точок, колір, тип точок.

`impulses` – вертикальні проекції точок лінії на вісь Ох (криволінійна трапеція), може застосовуватися з параметром для визначення кольору.

`[palette, false]` – забарвлення поверхні в тривимірній системі координат; за замовчування дана опція має значення `true`, тобто зображується поверхня із градієнтним забарвленням. Якщо значення `false` – поверхня зображується у вигляді ліній сітки.

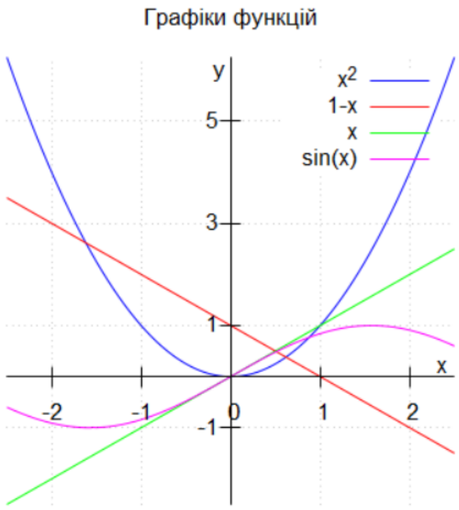
`[palette, [palette_1], [palette_2],…]` – налаштування палітри кольорів для градієнтного забарвлення поверхні.

Приклад 2.9. Побудувати графіки функцій $y = x^2$, $y = -x + 1$, $y = x$, $y = \sin x$, в одній і тій самій системі координат в межах $x \in [-2{,}5; 2{,}5]$. Осі координат зобразити суцільними лініями, що перетинаються в початку координат. Налаштувати підписи осей, сітку і заголовок графічної області.

Для виконання завдання записуємо:

```
(%i26) wxplot2d([x^2, −x+1, x, sin(x)], [x,−2.5, 2.5],
       [box, false], grid2d,
       [yx_ratio, 1], [axes, solid],
       [xtics, −2, 1, 2],  [ytics, −1, 2, 5],
       [label, ["x", 2.3, 0.2], [ "y", −0.2, 6]],
       [title, "Графіки функцій"]);
```

(%t26)

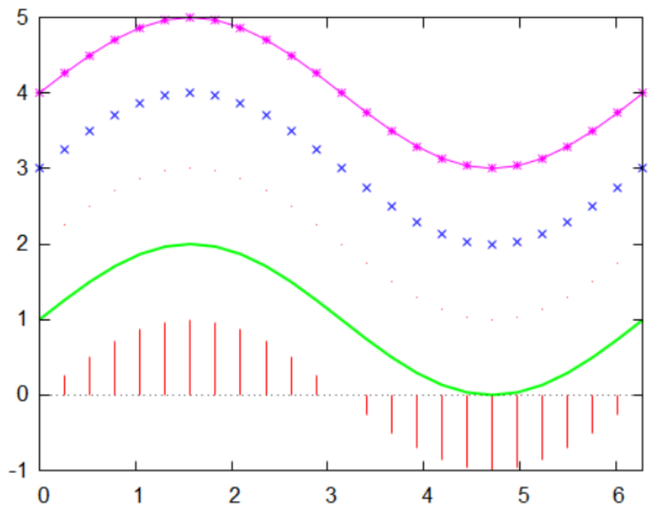


Приклад 2.10. Побудувати 5 ліній різного типу, розміру, кольору в межах одного графічного поля. Для розв'язування задачі за основу взяти графік функції $y = \sin x$.

```
(%i3) px: makelist(i, i, 0, 2·π, π/12)$
      py: makelist(sin(i), i, 0, 2·%pi, %pi/12)$

(%i4) wxplot2d(makelist([discrete, px, py+n],n,0,4),
                  [x, 0, 2·π], [y,−1,5],
                  [legend, ""],
                  [style, [impulses, 2, 3],[lines,2,3], [dots, 2],
                  [points,2,7,5], [linespoints,1,2,4,6]]
              )$
```

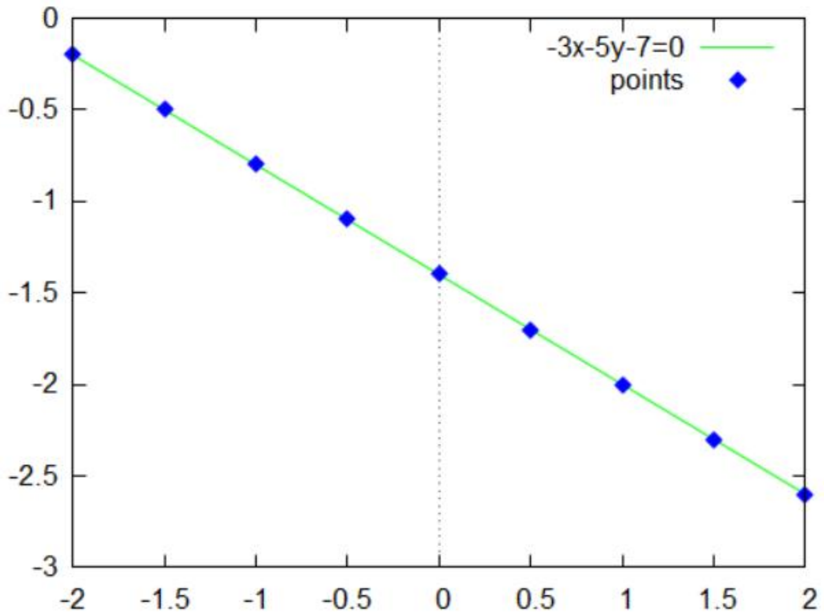
(%t4)



Приклад 2.11. Побудувати графік прямої $y = -(3x+7)/5$, зобразити суцільною лінією зеленого кольору. На прямій побудувати точки у вигляді ромбів синього кольору на проміжку $[-2; 2]$ з кроком 0,5 одиниць. На графічному полі створити напис з рівнянням прямої (додати написи до графіків).

```
(%i12) A: 3$ B: 5$ C: 7$
       l: −(A·x+C)/B$
       define(l(x),l)$
       px: makelist(i, i, −3, 3, 0.5)$
       py: makelist(subst(x=i, l),i, −3, 3, 0.5)$
       wxplot2d([l,[discrete,px,py]], [x,−2,2],
                [style, lines, points],
                [point_type, diamond],
                [color, green, blue],
                [legend, "−3x−5y−7=0", "points"]);
```

(%t12)



Задачі для самостійного розв'язування

1. Побудувати графік функції $y = y(x)$ на заданому проміжку.

1	$y = 3cos\frac{1}{2}x,\ x \in [-4\pi; 4\pi]$	6	$y = \frac{x^2+2x+2}{x+2}, x \in [-8; 5]$
2	$y = x \cdot \sin x, x \in [-8\pi; 8\pi]$	7	$y = 3^{tgx}, x \in [-5; 5]$
3	$y = \lvert x+1\rvert - \lvert x-2\rvert, x \in [-3; 6]$	8	$y = x + \cos x, x \in [-4\pi; 4\pi]$
4	$y = tg\lvert x\rvert, x \in [-4\pi; 4\pi]$	9	$y = \sqrt{1-\sin^2 x}, x \in [-3\pi; 3\pi]$
5	$y = tg2\pi x, x \in [-\pi; \pi]$	10	$y = \arccos(\sin x), x \in [-2; 2]$

2. Побудувати графік параметричної кривої. Задати однаковий масштаб вздовж координатних осей.

1	$x = cos\, 3\, t,\ y = sin\, 2\, t,\ t \in [0; 2\pi]$ (фігура Лісажу)	6	$x = t - \sin t, y = 1 - \cos t,$ $t \in [0;\ 6\pi]$ (циклоїда)
2	$x = 5\cos t,\ \ y = 5\sin t,$ $t \in [0; 2\pi]$ (коло)	7	$x = 4\cos t + 2\cos 6t,$ $y = 4\sin t + 2\sin 6t, t \in [-\pi;\ \pi]$
3	$x = 2\cos^3 t,\ \ y = 2\sin^3 t,$ $t \in [0; 2\pi]$ (астроїда)	8	$x = 2\sin^2 t, y = 2\sin^3 t,$ $t \in [-2\pi;\ 2\pi]$
4	$x = (1+\cos t)\cos t,$ $y = (1+\cos t)\sin t,\quad t \in [0; 2\pi]$ (кардіоїда)	9	$x = \cos t + t \cdot \sin t, y = \sin t + t \cdot \cos t, t \in [0;\ 10]$ (евольвента кола)
5	$x = 3\cos t,\quad y = 5\sin t,$ $t \in [0; 2\pi]$ (еліпс)	10	$x = \sin t\quad y = 2\sin 2t,\quad t \in [-2\pi;\ 2\pi]$

3. Побудувати графік функції $z = f(x, y)$ в заданих межах. На другому графіку зобразити лінії рівня поверхні.

1	$z = x\, sin\, 2\, y + y\, cos\, 3\, x,$ $(x;\ y) \in [-2\pi;\ 2\pi]$	6	$z = x^2 - y^2, (x;\ y) \in [-10;\ 10]$
2	$z = \sin 2y + \cos 2x,$ $(x;\ y) \in [-\pi;\ \pi]$	7	$z = 2^{xy+y}, x \in [-5;\ 5],$ $y \in [-1;\ 1]$

3	$z = 2x^2 + y^2$, $(x;\ y) \in [-5;\ 5]$	8	$z = \frac{y}{x}$, $(x;\ y) \in [-5;\ 5]$
4	$z = xy$, $(x;\ y) \in [-5;\ 5]$	9	$z = \frac{2}{x^2y^2+5}$, $(x;\ y) \in [-5;\ 5]$
5	$z = \sqrt{x^2 + y^2}$, $(x;\ y) \in [-8;\ 8]$	10	$z = \frac{x}{x^2+x^2y^2+1}$, $(x;\ y) \in [-5;\ 5]$

4. Зобразити на одному графічному полі послідовність побудови графіка функції $y = y(x)$ за допомогою геометричних перетворень. Основний графік зобразити потовщеною лінією червоного кольору.

1	$y = \sin\left\lvert x + \frac{\pi}{4}\right\rvert$	6	$y = \lvert x^2 + \lvert x\rvert - 6\rvert$
2	$y = 2cos\left\lvert x - \frac{\pi}{3}\right\rvert$	7	$y = 2\ cos(x - \frac{\pi}{3}) - 1$
3	$y = 2\sin\left\lvert x + \frac{\pi}{6}\right\rvert$	8	$y = 3\sin(x + 4) - 2$
4	$y = -\cos\left\lvert x - \frac{\pi}{4}\right\rvert$	9	$y = 2cos(x - \frac{\pi}{6}) + 2$
5	$y = \lvert 4x^2 - 4\lvert x\rvert + 3\rvert$	10	$y = 2\mathrm{tg}\left(x + \frac{\pi}{4}\right) - 1$

5. Побудувати дискретний графік, що заданий послідовністю точок значень функції $y = \sin 2x$ на проміжку $[-\pi; \pi]$ з кроком $\frac{\pi}{10}$.

6. Побудувати фігуру довільної форми, використовуючи дискретний графік.

7. Побудувати графік функції $y = 3x^4 - 5x^2 + 4$ та дотичну ($y = -2x$) до неї в точці $(-1; 2)$. Зобразити: лінія графіка – синього кольору, потовщена; дотична лінія – рожева, тонка; точка – червона, кругла. Встановити відображення ліній сітки.

8. Побудувати поверхню, задану параметрично: $x = \sin v \cos u$, $y = \sin v \sin u$, $z = \cos v$, $v \in [0; \pi]$, $u \in [0; 2\pi]$.

Контрольні питання

1. Яка з записаних функцій призначена для побудови графіка лінії, що задана параметрично?

a) plot2d(f(x), [x, -5, 5]);

b) plot2d(discrete, [1, 2, 3], [1, 2, 3]);

c) plot2d(parametric, x(t), y(t), [t, -10,10]);

d) plot2d([x(t), y(t)], [t, -10,10]);

2. Запис якої з поданих команд є правильним?

a) wxplot2d(tan(x), [x, -2*pi,2*%pi], [y, -10,10]);

b) plot2d(tan(x), [x, y, -2*pi,2*%pi]);

c) plot2d(tg(x), [x, -2*pi,2*%pi], [y, -10,10]);

d) wxplot2d(tan(x), [x, -2*pi,2*%pi]);

3. Яке призначення функції implicit_plot()?

a) побудова контурних ліній явно заданої функції;

b) побудова контурних ліній неявно заданої функції

c) побудова неявно заданої функції;

4. Як переглянути значення опцій у функціях Plot, які встановлені за замовчуванням?

a) get_plot_option();

b) get_plot_option(default);

c) set_plot_option();

d) set_plot_option(default);

5. Встановіть відповідність між опцією (a-d) та її призначенням (1-5):

a) grid2d	1) заголовок графіка;
b) [box, false]	2) підписи координатних осей;
c) [axes, solid]	3) графічне поле без рамки;
d) [title, “text”]	4) координатні осі – суцільні лінії;
	5) лінії сітки на графічному полі.

2.2 Графічний інтерфейс Draw

Для використання графічних процедур інтерфейсу Draw спочатку необхідно завантажити відповідний пакет: `load(draw)$`.

```
(%i1) load(draw);
(%o1) C:\maxima-5.42.2\share\maxima\5.42.2\share\draw\draw.lisp
```

2.2.1 Команди створення графіків Draw та їх основні опції

Пакет розширення графічних процедур Draw містить такі команди:

`draw(scene1,scene2,…,opts,…)` – побудова графічних об'єктів `scene1,scene2,…`, встановлення глобальних опцій `opts`.

`gr2d(opts,graph_obj1,graph_obj2,…)` – визначення двовимірного зображення як набору довільних двовимірних графічних об'єктів `graph_obj1,graph_obj2,…` з параметрами `opts`.

`gr3d(opts,graph_obj1,graph_obj2,…)` – визначення тривимірного зображення як набору довільних тривимірних графічних об'єктів `graph_obj1,graph_obj2,…` з параметрами `opts`.

`draw2d(opts,graph_obj1,graph_obj2,…)` – побудова двовимірного графіка як набору довільних двовимірних графічних об'єктів `graph_obj1,graph_obj2,…` з опціями `opts`.

`draw3d(opts,graph_obj,…)` – побудова тривимірного графіка як набору довільних тривимірних графічних об'єктів `graph_obj` з опціями `opts`.

Команди `draw2d(…)`, `draw3d(…)` та `draw(gr2d(…))`, `draw(gr3d(…))` – подібні, але мають деякі відмінності в написанні опцій та їх застосуванні. Зокрема за допомогою процедур `draw(gr2d(…))`, `draw(gr3d(…))` можна зобразити кілька окремих графіків в одному графічному полі.

На відміну від графічних процедур Plot, в процедурах Draw опції, які застосовуються до певного графічного об'єкта, повинні йому передувати. Позиція глобальних опцій, наприклад для визначення формату виведення графічної області, є довільною.

Записуються опції у вигляді рівності *name=value*, ліва частина якої – ім'я опції, права – її значення. Значення також можуть подаватися у вигляді списку, наприклад, для визначення проміжку значень змінної.

Основні опції налаштування графічного зображення в інтерфейсі Draw такі:

`user_preamble="text"` – задання вигляду системи координат (`set size ratio –1, set grid polar, set view map`)

`dimensions=[width,height]` – задання розмірів графічної області (ширина, висота).

`proportional_axes=xy` – задання однакових масштабів на координатних вісях.

`x(yz)range=[min,max]` – задання меж графічного поля по осі *x(yz)*.

`grid=true/false` – задання зображення ліній сітки, якщо `true`; за замовчуванням значення `false`.

`x(yz)tics_axis=true/false` – якщо `true`, то позначення шкали подається безпосередньо вздовж осі, якщо `false` – шкала на межі графічного поля.

`x(yz)tics=value` – задання шкали вздовж осі. За замовчуванням `xtics=auto`.

Також можливі такі значення опції `xtics` (на прикладі осі Ox):

`xtics=none` – шкала відсутня;

`xtics=[start,incr,end]` – поділки на осі починаються з `start` і закінчуються значенням `end` з інтервалом довжиною `incr`;

`xtics={n1,n2,…}` – шкала на осі в точках *n1,n2,…*;

`xtics={["label1",n1],["label2",n2],…}` – шкала на осі в точках *n1,n2,…* з написами *label1, label2*, …

`x(yz)tics_rotate=true/false` – поворот підписів шкали на 90º.

`title="text"` – заголовок зображення, за замовчуванням: порожній рядок.

`key="text"` – назва графіка в написах графічного поля, за замовчуванням: порожній рядок.

`x(yz)label="text"` – підпис осі x(yz).

`x(yz)axis=true/false` – зображення осей на графіку.

`x(yz)axis_width=width` – товщина лінії відповідної осі координат.

`x(yz)axis_color=colorname` – колір відповідної осі.

`x(yz)axis_type=solid/dots` – тип лінії відповідної осі: суцільна лінія (solid) або пунктирна (dots).

`columns=n` – число колонок для кількох графіків на одному рисунку.

`background_color=name` – колір фону графічної області.

`color=colorname` – колір графічного об'єкта, задається назвою (Додаток 1) або 16-ковим кодом RGB.

`fill_color=colorname` – колір заливання графічного об'єкта.

`line_width=width` – ширина лінії (число).

`line_type=solid/dots` – тип лінії (суцільна, пунктирна).

`point_size=size` – розмір точки, за замовчуванням point_size=1.

`point_type=n` – тип точки, можливі такі значення: -1(none), 0(dot), 1(plus), 2(multiply), 3(asterisk), 4(square), 5(filled_square), 6(circle), 7(filled_circle), 8(up_triangle), 9(filled_up_triangle), 10(down_triangle), 11(filled_down_triangle), 12(diamant), 13(filled_diamant).

`point_joined=true/false/impulses` – з'єднання точок лінійними відрізками, якщо `true`; точки не з'єднані, якщо `false`; від точок проведені вертикальні прямі до осі *Ox* (або площини *Oxy* в просторі), якщо `impulses`.

`nticks=n` – задання частоти дискретизації точок лінії для явно, в полярних координатах та параметрично заданих ліній; за замовчуванням `nticks=29`, для того щоб лінія була більш плавною, дану опцію необхідно збільшити.

`ip_grid=[nx,ny]` – задання частоти дискретизації точок лінії для неявно визначених функцій і кривих ліній; за замовчуванням `ip_grid=[50,50]`.

`transform=[list]` – перетворення координат графічного об'єкта згідно формул `list`. За замовчуванням `transform=none` – координатний простір не перетворюється.

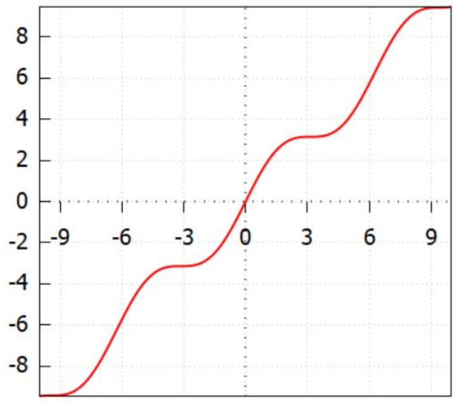
Для того, щоб не повторювати запис налаштування опцій в кожній новій процедурі, в пакеті `draw` є спеціальна команда визначення користувацьких опцій. Встановлення опцій користувача за замовчуванням виконується за процедурою `set_draw_defaults(opt1,opt2,…)`.

Приклад 2.12. Встановити за замовчуванням опції користувача: відображення координатних осей, шкали, пропорційний масштаб осей, лінії сітки, колір, товщину лінії графіка. Побудувати графік явно заданої функції $y = sinx + x$.

```
(%i2) set_draw_defaults(
        xaxis=true, yaxis=true,  /·показати координатні вісі·/
        xaxis_type=dots, yaxis_type=dots, /· координатні осі пунктиром·/
        xaxis_width=2, yaxis_width=2, /· ширина осей координат·/
        xtics=3,  /· підписи на осі Ox, кратні трьом·/
        xtics_axis=true, /· підписи розташувати на осі Ox·/
        proportional_axes=xy, /· дотримуватися пропорціності вздовж осей·/
        grid=true, /· лінії сітки координатної площини·/
        color=red, /· колір лінії графіка ·/
        line_width=2)$ /· товщина лінії графіка ·/

(%i3) wxdraw2d(explicit(sin(x)+x, x,-10,10));
```

(%t3)

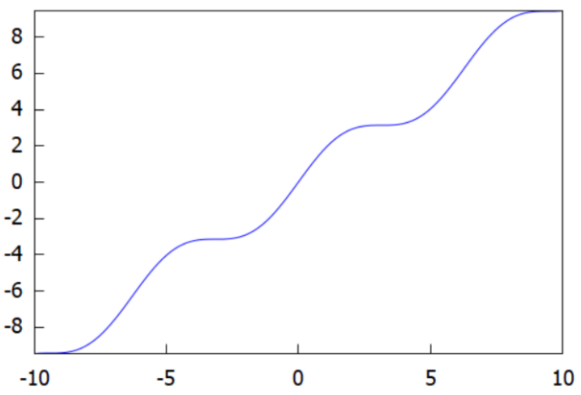


(%o3)

За викликом процедури без аргументів `set_draw_defaults()` скасовуються користувацькі налаштування:

(%i4) set_draw_defaults()$

(%i5) wxdraw2d(explicit(sin(x)+x, x,−10,10));

(%t5) 

(%o5)

Одним із аргументів процедури `wxdraw2d` в розглянутому вище прикладі є запис `explicit(sin(x)+x,x,-10,10)`. Цим записом задається графічний об'єкт побудови, а саме – графік явно заданої функції. Розглянемо ряд процедур для зображення двовимірних та тривимірних графічних об'єктів, які є в пакеті *draw*.

2.2.2 Двовимірні графічні об'єкти

За допомогою процедур пакету *draw* можна зобразити такі графічні об'єкти:

`explicit(f(x),x,x1,x2)` – графік явно заданої функції $f(x)$, x змінюється в межах від *x1* до *x2*.

`implicit(equation,x,x1,x2,y,y1,y2)` – графік неявно заданої функції або кривої лінії, що задана рівнянням `equation`; змінні *x* та *y* змінюються в межах від *x1* до *x2* та від *y1* до *y2* відповідно.

`parametric(x(t),y(t),t,t1,t2)` – побудова кривої, що задана параметрично, з параметром $t_1 \le t \le t_2$.

`polar(r(φ),φ,φ1,φ2)` – графік кривої `r(φ)`, що задана в полярних координатах; полярний кут φ змінюється в межах від φ1 до φ2.

`points([x1,x2,…],[y1,y2,…])` – точки з координатами $(x_i; y_i)$, *i=1,2,…*, що задані списком значень абсцис та списком значень ординат.

`points([[x1,y1],[x2,y2],…])` – точки з координатами $(x_i;y_i)$, *i=1,2,…,* що задані списком пар координат.

`polygon([x1,x2,…],[y1,y2,…]))` – многокутник, з вершинами в точках $(x_i;y_i)$, *i=1,2,…*, що задані списками значень абсцис та ординат.

`polygon([[x1,y1],[x2,y2],…])` – многокутник, з вершинами в точках $(x_i;y_i)$, *i=1,2,…*, що задані списком пар координат.

`rectangle([x1,y1],[x2,y2])` – прямокутник, в якому $(x_1;y_1)$ – координати верхньої лівої вершини та $(x_2;y_2)$ – координати нижньої правої вершини.

`label([“text”,x,y],…)` – напис в точці з координатами $(x;y)$, вирівнювання, орієнтація задається опціями.

`vector([x,y],[dx,dy])` – вектор з початком в точці $(x;y)$ та координатами $(dx;dy)$

`region(expr,x,x1,x2,y,y1,y2)` – область в декартовій системі координат, що задана відношенням *expr* і яка визначається умовою *expr=true*

`ellipse(x0,y0,a,b,w1,w2)` – еліпс з центром в точці $(x_0;y_0)$ та півосями *a* і *b*, `w1`, `w2` – значення початкового і кінцевого кута, в секторі якого будується еліпс.

Отже, в Maxima достатнє число геометричних форм та налаштувань їх опцій, щоб створювати довільні графічні композиції.

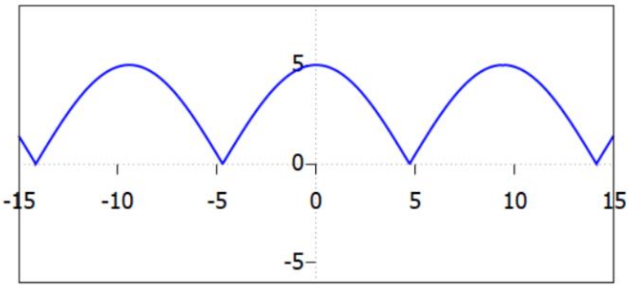
Розглянемо приклади побудови графіків та створення графічних зображень за допомогою засобів інтерфейсу Draw.

Приклад 2.13. Побудувати графік функції $y = \left|5cos\,\frac{x}{3}\right|$.

Дана функція задана явно. Тому скористаємося графічним об’єктом `explicit()`.

```
(%i3) wxdraw2d(proportional_axes=xy, /· пропорціний масштаб ·/
              xaxis=true, yaxis=true, /·показати координатні вісі·/
              ytics_axis=true, /·показати підписи осі Oy·/
              ytics=5, /· підписи осі Oy, кратні 5·/
              xtics_axis=true, /·показати підписи осі Ox·/
              yrange=[−6,8], /·межі по осі Oy·/
              line_width=2, /· товщина лінії графіка ·/
              explicit(abs(5·cos(x/3)), x,−15,15)
              );
```

(%t3)



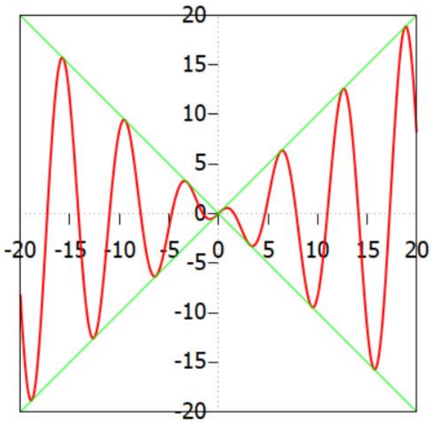
(%o3)

В графіках, які складаються з кількох графічних об'єктів, список аргументів команди draw2d(3d) може бути занадто довгим і важким для наочного сприймання. Щоб цьому запобігти, запис процедури можна структурувати, користуючись довільними лінійними проміжками, розривами, відступами між іменами змінних, опціями. Також можна визначати графічні об'єкти не безпосередньо в записі графічної команди, а попередньо призначити графічному об'єкту ім'я у вигляді змінної і надалі використовувати це ім'я в списку аргументів команди draw2d(3d).

Приклад 2.14. Побудувати на одному графічному полі графіки функцій $f1(x) = xcosx$, $f2(x) = x$, $f3(x) = -x$.

Змінній `f1` присвоїмо значення аналітичного виразу відповідної функції, використовуючи графічний об'єкт `explicit()`. Тоді в процедурі `draw2d()` будемо використовувати ім'я змінної `f1` для запису графічного об'єкта.

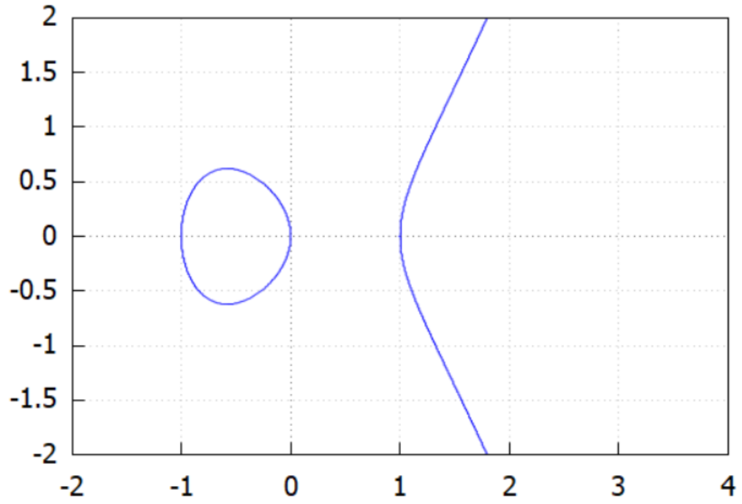
```
f1: explicit(x·cos(x), x,−20,20)$
wxdraw2d(proportional_axes=xy,
            xaxis=true, yaxis=true,
            ytics_axis=true, ytics=5,
            xtics_axis=true,
            line_width=2, /· товщина лінії графіка  f1·/
            color=red,  /· колір лінії графіка  f1·/
            f1,
            color=green, /· колір лінії графіка  y=x·/
            line_width=1,  /· товщина лінії графіка  y=x·/
            explicit(x, x,−20,20),
            explicit(−x,x,−20,20));
```



Приклад 2.15. Побудувати графік кривої, що задана рівнянням $y^2 = x^3 - x$.

Для побудови графіка скористаємося графічним об'єктом `implicite()`.

```
wxdraw2d(proportional_axes=xy, grid=true,
            xaxis=true, yaxis=true,
            ip_grid=[100,100], /·гладкість  лінії·/
            implicit(y^2=x^3−x, x,−2,4,y,−2,2));
```

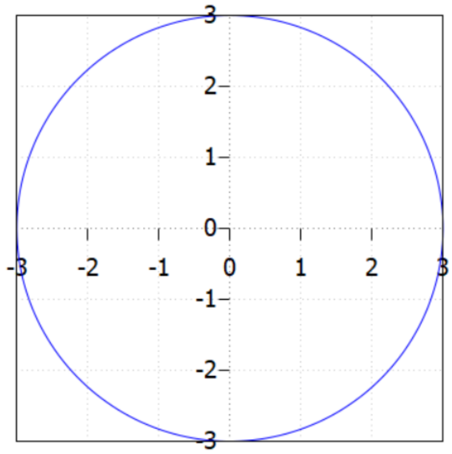


Для збільшення гладкості кривої лінії на малюнку в процедурі `draw2d()` використовується опція задання частоти точок кривої лінії `ip_grid`. Ця опція використовується для графічних об'єктів неявно заданих функцій або кривих, заданих рівнянням $F(x; y) = 0$ і є аналогом опції `ntics`, що використовується для явно, параметричних кривих та в полярних координатах.

Приклад 2.16. В площині *Oxy* побудувати криву, що задана параметричними рівняннями $x(t) = 3\sin t$, $\mathrm{y(t)} = 3\cos t$, $0 \leq \mathrm{t} \leq 2\pi$ (коло з центром у початку координат і радіусом *R*=3).

```
(%i15) wxdraw2d(proportional_axes=xy, grid=true,
                xaxis=true, yaxis=true,
                ytics_axis=true,  xtics_axis=true,
                nticks=200,
                parametric(3·sin(t), 3·cos(t), t,0,2·π));
```

(%t15)



Приклад 2.17. Зобразити графічну ілюстрацію розв'язування рівняння $3^x = 4 - x$.

В одній системі координат будуємо графіки функцій, записані в лівій і правій частині рівняння. Для побудови графіків функцій $y1 = 3^x$, $y2 = 4 - x$ використовуємо процедуру `explicit()`.

Знаходимо точку перетину графіків. За графіком визначаємо її координати М(1;3). Абсциса знайденої точки є розв'язком рівняння. Виконавши перевірку, переконуємося, що знайдена точка задовольняє обидва рівняння $y1 = 3^x$, $y2 = 4 - x$. Оскільки функція $y1 = 3^x$ є зростаючою, а функція $y2 = 4 - x$ – спадною для всіх $x \in R$, то корінь $x = 1$ – єдиний корінь даного рівняння.

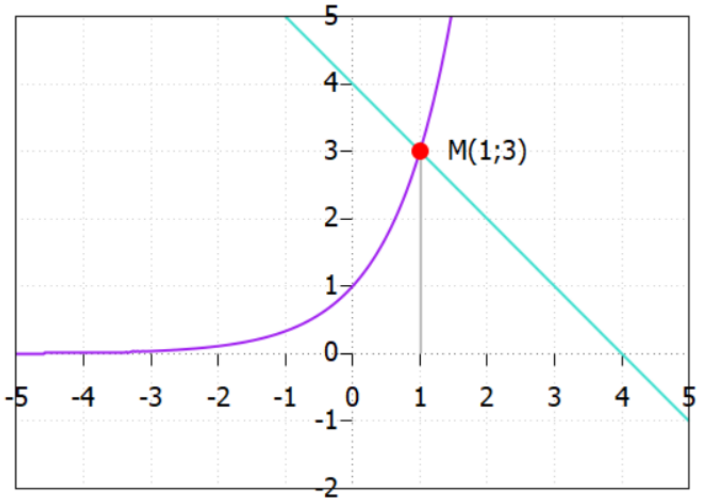
Для зображення точки М(1;3) задаємо її колір та тип за допомогою опцій і використовуємо процедуру `point()`, в якій вказуємо координати точки.

Для її підпису використовуємо процедуру `label()` із вказівкою координат місця її підпису.

Щоб зобразити проєкцію точки М(1;3) на вісь абсцис, побудуємо відрізок. Для цього скористаємося процедурою `implicit()`, за допомогою якої зображуємо рівняння прямої $x = 1$ для довільних значень x і для $y \in [0; 3]$.

```
y1: explicit(3^x, x,–5,5)$  y2: explicit(4–x, x,–5,5)$
wxdraw2d(proportional_axes=xy, xaxis=true, yaxis=true, grid=true,
            ytics_axis=true, xtics_axis=true, /·шкала на осях·/
            yrange=[–2,5], xtics=[–5,1,5], /·підписи на осях·/
            line_width=2, /· товщина лінії графіка  y1·/
            color=purple,  /· колір лінії графіка  y1·/
            y1, /·графік  y1·/
            color=turquoise, /· колір лінії графіка  y2·/
            y2, /·графік  y2·/
            /·відрізок для визначення проєкції точки·на вісь x·/
            color=gray, /· колір відрізка прямої x=1·/
            implicit(x=1, x,–2,4,y,0,3), /·  відрізок прямої x=1·/

            color=red,  point_size=2, /· колір та  розмір точки·/
            point_type=filled_circle, /· тип точки, кругла·/
            points([[1,3]]), /· точка·/
            color=black, /· колір підпису точки·/
            label(["M(1;3)",2,3]) );/·підпис точки·/
```

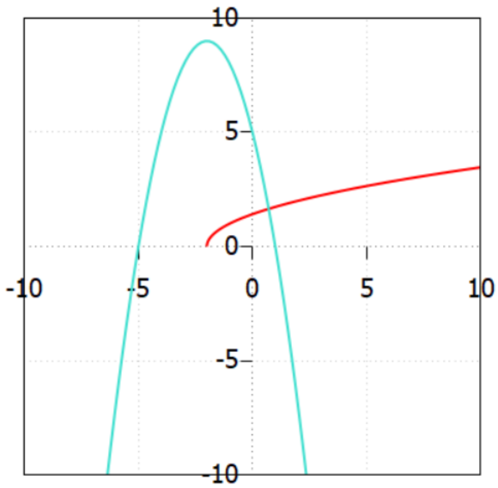


Для деяких рівнянь неможливо знайти точний розв'язок. Наприклад, рівняння $\sqrt{x+2} = -x^2 - 4x + 5$ має єдиний дійсний корінь, який можна подати у вигляді нескінченного десяткового дробу, тобто знайти розв'язок з певною точністю. Для розв'язування такого виду рівнянь в Maxima є функція `find_root(expr, x, x1, x2)`, де `expr` – вираз рівняння, `x` – змінна, відносно якої потрібно знайти роз'язок. Щоб застосувати цю функцію, потрібно вказати для змінної `x` межі `x1, x2`, такі, що $x \in (x1; x2)$, тобто відокремити шуканий корінь. Досить легко знайти інтервал, де знаходиться корінь рівняння, якщо зобразити графіки лівої і правої частини даного рівняння та подивитися на абсциси точок їх перетину. Також, застосовуючи графічний спосіб, можна знайти і саме значення кореня з деякою точністю.

Приклад 2.18. Знайти розв'язок рівняння $\sqrt{x+2} = -x^2 - 4x + 5$ графічним способом з точністю до сотих.

В одній системі координат будуємо графіки функцій лівої і правої частини даного рівняння за допомогою процедури `explicit`(), враховуючи область допустимих значень виразу $\sqrt{x+2}$, тобто $x \in [-2; +\infty)$

```
f1: explicit(sqrt(x+2), x,−2,10)$
f2: explicit(−(x^2)−4·x+5, x,−10,10)$
wxdraw2d(proportional_axes=xy, grid=true,
            xaxis=true, yaxis=true,
            ytics_axis=true,   xtics_axis=true,
            yrange=[−10,10],
            line_width=2, /· товщина лінії графіка  f1·/
            color=red,  /· колір лінії графіка  f1·/
            f1,
            color=turquoise, /· колір лінії графіка  y=x·/
            f2);
```



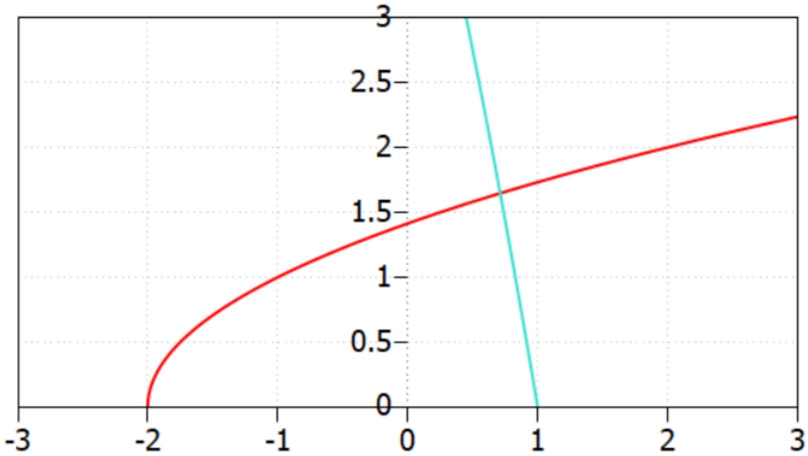
За одержаним графіком важко визначити числове значення кореня з точністю до сотих, але напевне можемо сказати, що корінь знаходиться в межах інтервалу $(0;5)$. Тепер для визначення кореня скористаємося процедурою `find_root()`.

```
find_root(sqrt(x+2)=−(x^2)−4·x+5, x, 0, 5);
0.7116928565730021
```

Отже, коренем даного рівняння з точністю до сотих є $x = 0{,}71$.

Перевіримо знайдене значення за допомогою графічного методу. Скопіюємо код, написаний для побудови графіка лівої і правої частини рівняння. Але перш, ніж його запускати на виконання звузимо межі побудови графіка вздовж осей *Ox, Oy*. Тобто змінюємо межі змінної `x` в процедурах `explicite()` та межі опції `yrange=[]`.

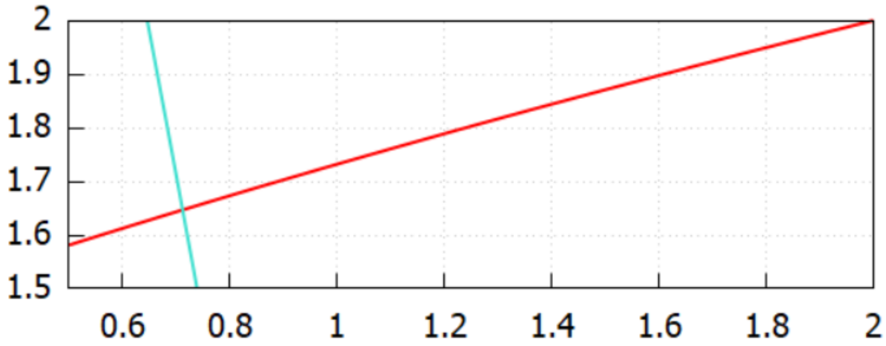
```
f1: explicit(sqrt(x+2), x,−2,3)$
f2: explicit(−(x^2)−4·x+5, x,−3,3)$
wxdraw2d(proportional_axes=xy, grid=true,
                xaxis=true, yaxis=true,
                ytics_axis=true, xtics_axis=true,
                yrange=[0,3],
                line_width=2, /· товщина лінії графіка  f1·/
                color=red,  /· колір лінії графіка  f1·/
                f1,
                color=turquoise, /· колір лінії графіка  y=x·/
                f2);
```



Тепер за графічною ілюстрацією встановлюємо, що точка перетину графіків, абсциса якої є коренем даного рівняння, знаходиться в межах $x \in (0.5; 1)$, $y \in (1.5; 2)$.

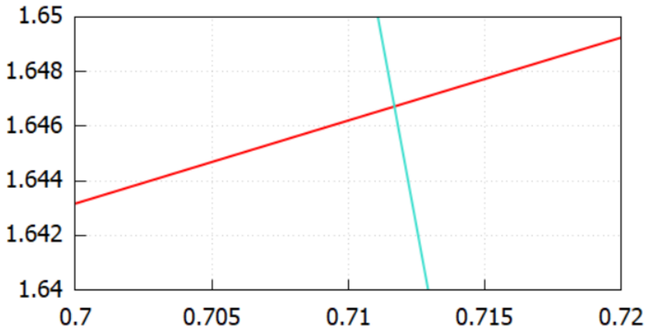
Знову копіюємо код побудови графічної ілюстрації розв'язування даного рівняння і змінюємо межі побудови графіків.

```
f1: explicit(sqrt(x+2), x,0.5,2)$
f2: explicit(−(x^2)−4·x+5, x,0.5,2)$
wxdraw2d(proportional_axes=xy, grid=true,
            xaxis=true, yaxis=true,
            ytics_axis=true, xtics_axis=true,
            yrange=[1.5,2],
            line_width=2, /· товщина лінії графіка  f1·/
            color=red,  /· колір лінії графіка  f1·/
            f1,
            color=turquoise, /· колір лінії графіка  y=x·/
            f2);
```



За побудованим графіком більш точно вказуємо межі координат точки перетину графіків. Для більшої наочності графічного зображення в процедурі `draw2d()` допишемо опцію налаштування підписів вздовж осі x так, щоб підписи були від 0,7 до 0,72 з кроком 0,005.

```
f1: explicit(sqrt(x+2), x,0.7,0.72)$
f2: explicit(−(x^2)−4·x+5, x,0.7,0.72)$
wxdraw2d(proportional_axes=xy, grid=true,
            xaxis=true, yaxis=true,
            ytics_axis=true, xtics_axis=true,
            yrange=[1.64,1.65], xtics=[0.7,0.005,0.72],
            line_width=2, /· товщина лінії графіка  f1·/
            color=red,  /· колір лінії графіка  f1·/
            f1,
            color=turquoise, /· колір лінії графіка  y=x·/
            f2);
```



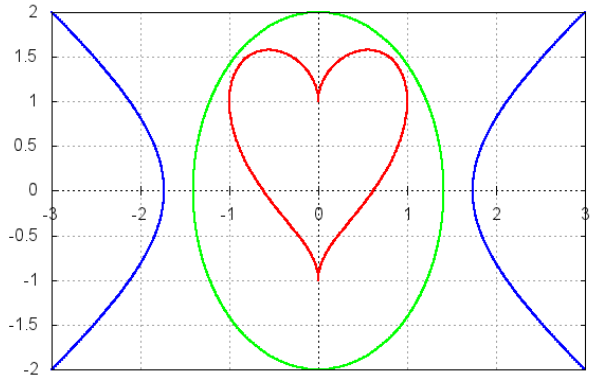
За одержаним зображенням визначаємо, що абсциса точки перетину графіків, тобто корінь даного рівняння, з точністю до сотих дорівнює 0,71. Це значення співпадає з одержаним раніше за допомогою функції `find_root()`.

Приклад 2.19. Побудувати графіки кривих ліній, що задані рівняннями: $x^2+(y-\sqrt{|x|})^2=1$, $\frac{x^2}{3}-\frac{y^2}{2}=1$, $\frac{x^2}{2}+\frac{y^2}{4}=1$ на одному малюнку.

Лінії задані у неявному вигляді, тому для їх побудови скористаємося графічним об’єктом `implicit()`.

Спочатку призначимо графічним об’єктам даних ліній імена, які далі будемо застосовувати в процедурі створення графіка `wxdraw2d()`.

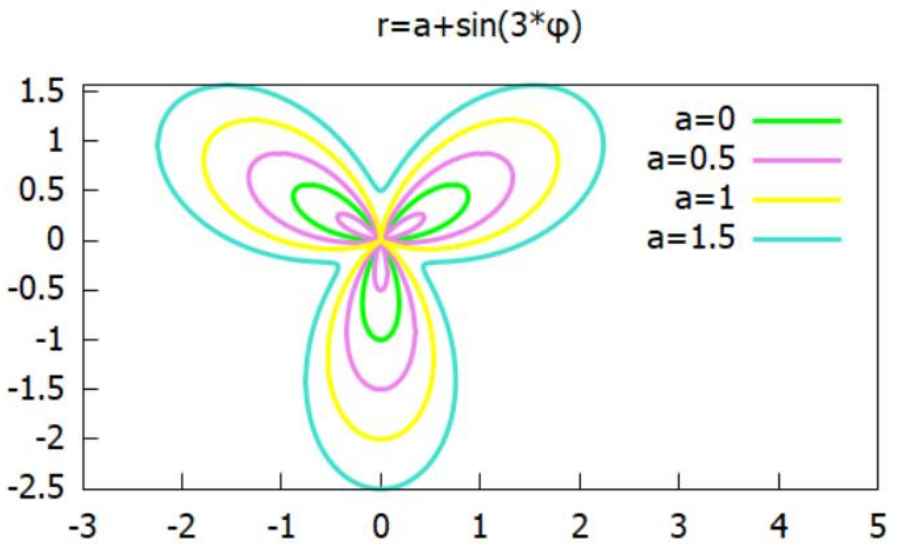
```
f1: implicit(x^2+(y–sqrt(abs(x)))^2=1,x,–2,2,y,–2,2)$
f2: implicit(x^2/3–y^2/2=1,x,–3,3,y,–2,2)$
f3: implicit(x^2/2+y^2/4=1,x,–3,3,y,–2,2)$
wxdraw2d(dimensions=[600,400], /· розмір графічної області·/
    xaxis=true, yaxis=true,  /· зображення координатних осей·/
    xtics_axis=true, /· зображення підписів на осі Ox·/
    xtics=[–3,1,3], /·шкала на осі Ox від –3 до 3 з кроком 1·/
    ip_grid=[80,80], grid=true,
    color=red, f1, /·графік f1 червоного кольору·/
    color=blue, f2, /·графік f2 синього кольору·/
    color=green, f3); /·графік f3 зеленого кольору·/
```



Приклад 2.20. В полярних координатах побудувати лінії $r = a + \sin 3\varphi$. Дослідити вплив значень параметра a.

Розглянемо випадки, коли $a = 0$, $0 < a < 1$, $a = 1$, $a > 1$. Для кожного випадку побудуємо лінії різного кольору. Нехай a набуває значень: 0; 0,5; 1; 1,5. В написах до графіка відобразимо значення параметра, використовуючи опцію `key`. Опція `user_preamble="set grid polar"` є обов'язковою для побудови графіка в полярних координатах. Записуємо:

```
wxdraw2d(
user_preamble="set grid polar", /·перехід до полярної системи координат·/
xrange=[−3,5], proportional_axes=xy,
nticks=500, title="r=a+sin(3·φ)", line_width=3,
color=green, key="a=0",  polar(sin(3·φ),φ,0,2·π),
color=violet, key="a=0.5",  polar(0.5+sin(3·φ),φ,0,2·π),
color=yellow, key="a=1",  polar(1+sin(3·φ),φ,0,2·π),
color=turquoise, key="a=1.5", polar(1.5+sin(3·φ),φ,0,2·π))$
```



Як видно з одержаного графіка, дане рівняння визначає лінію трилисника. Для $a = 1{,}5$ пелюстки кривої мають незакінчений вигляд, їх довжина складає 2,5 одиниці. Для $a = 0$ і $a = 1$ маємо трипелюсткові троянди, довжина пелюсток яких рівна 1 і 2 відповідно. Для $a = 0{,}5$ отримали шестипелюсткову троянду, в якої три великі та три малі пелюстки мають довжину 1,5 та 0,5 одиниць відповідно.

Приклад 2.21. Параметричні рівняння епіциклоїди мають вигляд:

$x=(a+1)\left(\cos t-\dfrac{\cos((a+t)t)}{a+1}\right)$, $y=(a+1)\left(\sin t-\dfrac{\sin((a+t)t)}{a+1}\right)$. Побудувати графіки епіциклоїди, якщо $a=1,2,3,4$.

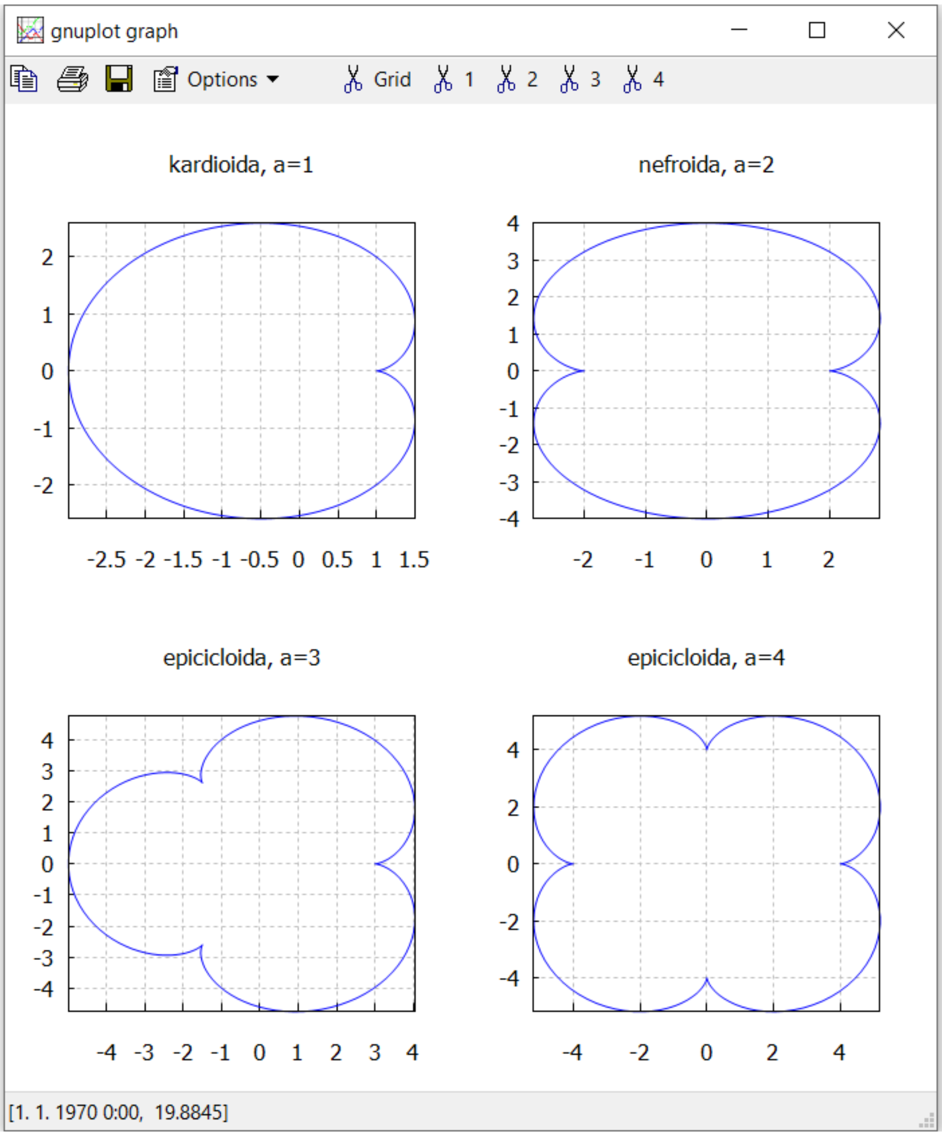
Використовуючи процедуру `makelist`, створимо список параметричних рівнянь епіциклоїди для значень параметра $a=1,2,3,4$, призначаємо йому ім'я list.

Цікаво одночасно спостерігати графіки для різних значень параметра. Тому створимо малюнок, на якому розмістимо графіки для всіх значень параметра *a*. Для цього слід скористатися процедурою `wxdraw(gr2d(…))`.

Щоб на одному зображенні компактно розмістити 4 різних графіки, застосуємо опцією `columns=2`, за якою графічна область ділиться на дві колонки.

Слід зазначити, що опція `columns` характерна лише для процедури `wxdraw(gr2d(…))`. За допомогою опції `title` в заголовках графіків виведемо назву епіциклоїди для кожного значення *a*.

```
list: makelist(parametric((a+1)·(cos(t)−(cos((a+1)·t))/(a+1)),
                          (a+1)·(sin(t)−(sin((a+1)·t))/(a+1)),
                          t,0,2·π),a,1,4)$
draw(gr2d( title="kardioida, a=1", nticks=200, grid=true, list[1]),
     gr2d(title="nefroida, a=2" , nticks=200, grid=true, list[2]),
     gr2d(title="epicicloida, a=3",nticks=200, grid=true, list[3]),
     gr2d(title="epicicloida, a=4",nticks=200, grid=true, list[4]),
     columns=2)$
```



Приклад 2.22. Побудувати пряму, яка задана точкою M(2;1) та напрямним вектором $\overrightarrow{AB}$ з координатами A(-2;1), B(1;3). Зобразити точку M круглою червоною точкою на прямій, вивести її позначення. Зеленим кольором зобразити стрілку, яка позначатиме напрямний вектор прямої. Вивести позначення осей та заголовок графіка, лінії сітки. Заголовком є рівняння лінії, що проходить через дану точку в заданому напрямі.

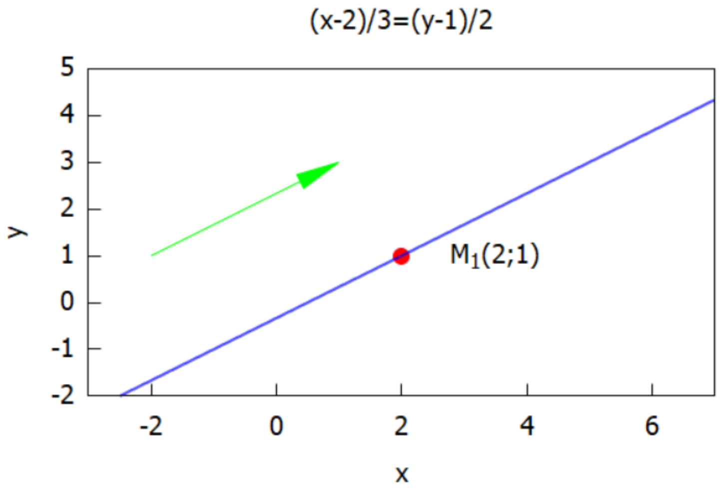
Якщо задано вектор $\vec{p}(m;n)$ паралельний прямій і точку $M_0(x_0;y_0)$ на цій прямій, то рівняння прямої можна записати у вигляді $\frac{x-x_0}{m} = \frac{y-y_0}{n}$. Вектор $\vec{p}(m;n)$ називається напрямним вектором прямої.

Знаходимо координати вектора $\overrightarrow{AB} = (1+2;3-1) = (3;2)$. Отже, рівняння шуканої прямої матиме вигляд $\frac{x-2}{3} = \frac{y-1}{2}$.

```
A:[−2,1]$ B:[1,3]$ M:[2,1]$ /· координати даних точок·/
AB:[3,2]$ /· координати вектора·/
wxdraw2d(
  xlabel="x", ylabel="y",/·написи на осях·/
  title="(x−2)/3=(y−1)/2", /·заголовок, рівняння прямої·/
  ip_grid=[200,200],/·частота дискретизації·/
  color=green, /· колір вектора·/
  head_angle=10, /· кут стрілки вектора·/
  head_length=0.7, /· довжина стрілки вектора·/
  vector(A,AB), /· вектор·/
  color=red, /· колір точки·/
  point_type=7, /· тип точки, кругла·/
  point_size=2, /· розмір точки·/
  points([M]), /· точка·/
  color=black, /· колір підпису точки·/
  label(["M_{1}(2;1)",3.5,1]),/·підпис точки·/
```


```
  label(["M_{1}(2;1)",3.5,1]),/·підпис точки·/
```


```
  line_type=dots, color=blue,/·тип лінії та колір прямої·/
  line_width=2,/·ширина лінії прямої·/
/· пряма, рівняння прямої подається як неявно задана функція: ·/
  implicit((x−2)/3=(y−1)/2,x,−3,7,y,−2,5));
```



В даному прикладі для зображення вектора використовується графічний об’єкт `vector()`, перший аргумент якої – координати точки початку вектора, другий аргумент – координати вектора. Дана функція має характерні графічні опції:

`head_both=true/false` – стрілка з обох кінців;

`head_length=n` – довжина стрілки вектора, n – ціле додатне число;

`head_angle=n` – кут стрілки (за замовчуванням 45);

`head_type` – тип стрілки; можливі значення: `'filled` (за замовчуванням), `'empty`, `'nofilled`.

Для створення підпису точки використовувалась функція `label()`, аргументами якої є текст підпису та координати місця напису. Дана функція має власні графічні опції:

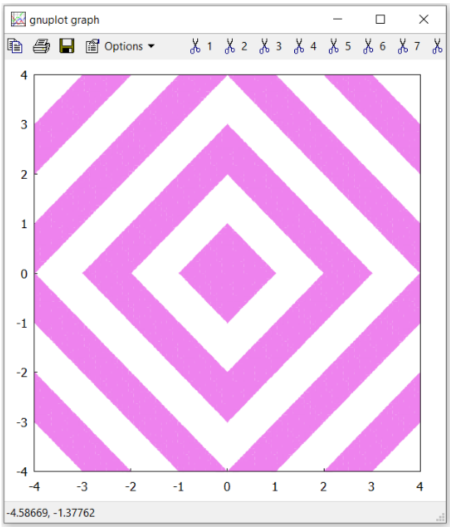
`label_alignment` – вирівнювання напису відносно його координат, можливі значення: `center` – по центру (за замовчуванням), `left` – по лівому краю, `right` – по правому краю.

`label_orientation` – орієнтація напису, можливі значення: `horizontal, vertical`.

Приклад 2.23. Зобразити геометричне місце точок, які задовольняють нерівність $sin(\pi(|x|+|y|))\geq 0$.

Геометричне місце точок, що задане нерівністю від двох змінних легко зобразити, використовуючи функцію `region()`, що визначає область в декартовій системі координат за умовою *expr(x,y)=true*.

```
draw2d(dimensions=[600,600],
        x_voxel=70, y_voxel=70,
        fill_color=violet,
        region(sin(π·(abs(x)+abs(y)))>=0, x,−4,4,y,−4,4))$
```



Слід зауважити, що частота дискретизації для об’єкта region() задається опціями x_voxel, y_voxel (за замовчуванням мають значення 10). Колір заповнення області, що відповідає геометричному місцю точок визначається опцією fill_color.

Приклад 2.24. В декартовій системі координат створити малюнок, використовуючи процедури для побудови графічних об’єктів (трикутники, прямокутник, многокутник, коло, еліпс та ін.) та засоби налаштування їх параметрів.

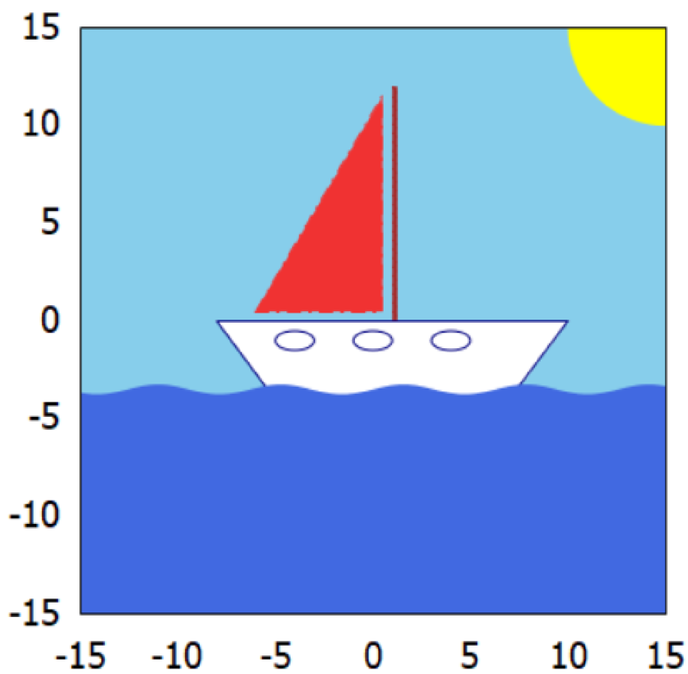
Перш ніж зображувати графічні елементи, налаштуємо однаковий масштаб вздовж координатних осей та визначимо розміри графічного поля, в межах якого будуть всі фігури малюнка. Спочатку записуємо властивості графічні об’єктів і самі об’єкти, які є фоном, зокрема небо. Останніми записуємо об’єкти, що знаходяться на передньому плані малюнка, в цьому прикладі – море.

Якщо використовуємо опції, то слід враховувати, що задана опція відноситься до всіх записаних після неї графічних об’єктів, поки не зміниться її значення. Наприклад опцією border задається контур фігури. За замовчуванням вона має значення true, тобто контур фігури малюється. Якщо записати border=false, то контур графічних об’єктів малюватися не буде. В цьому прикладі без контуру зображуємо небо, сонце, щоглу і парус. В такому разі для цих графічних елементів достатньо визначити колір заливки фігури за допомогою опції fill_color.

Параметр transparent означає прозорість фігури і за замовчуванням має значення false. Для того, щоб повернути малювання контуру многокутника для зображення корпусу човника, записуємо border=true. В такому разі потрібно вказати за допомогою параметра color колір контуру.

Для зображення моря у вигляді синусоїди використаємо процедуру explicit() разом з опцією filed_func=true, за якою відбувається заповнення фігури кольором, що обмежена зверху функцією, а з боків і знизу – межами графічної області. Для застосування опції filed_func попередньо потрібно задати колір заливки за допомогою опції fill_color.

```
wxdraw2d(proportional_axes=xy,
        xrange=[-15,15], yrange=[-15,15], /·межі графічного поля·/
        border=false, fill_color =skyblue,
        rectangle([-15,15],[15,-5]), /·небо·/
        fill_color = yellow,
        ellipse(15,15,5,5,270,-90), /·сонце·/
        fill_color = brown,
        rectangle([1,12],[1.25,0]), /·щогла·/
        fill_color =light_red,
        triangle([-6,0.5],[0.5,0.5],[0.5,11.5]), /·парус·/
        border=true, color=navy,
        fill_color =white,
        polygon([[-8,0],[10,0],[7,-4],[-5,-4]]), /·корпус·/
        transparent=true,
        ellipse(-4,-1,1,1/2,0,360), /·ілюмінатор·/
        ellipse(0,-1,1,1/2,0,360),  /·ілюмінатор·/
        ellipse(4,-1,1,1/2,0,360),  /·ілюмінатор·/
        fill_color = royalblue,
        filled_func = true,
        explicit(0.25·sin(x)-3.5,x,-15,15)); /·море·/
```



Опція `filed_func` за замовчуванням має значення `false`. Окрім значень `true` і `false`, їй можна надати значення виразу явно заданої функції. Тоді буде заповнюватися кольором фігура, що обмежена функцією, заданою `filed_func` та функцією з процедури `explicit()`.

Задачі для самостійного розв'язування

1. Дано функції $f(x)$ та $g(x)$. Побудувати графіки функцій: а) $f(g(x))$; б) $f(x)+g(x)$; в) $f(x)g(x)$, г) $|g(x)|$.

1) $f(x) = sin\frac{1}{x}$, $g(x) = x^2$;

2) $f(x) = x$, $g(x) = \frac{1}{x}$;

3) $f(x) = x$, $g(x) = \frac{1}{x^2}$;

4) $f(x) = x^2$, $g(x) = \frac{1}{x}$;

5) $f(x) = \sin x$, $g(x) = x^2 - 1$;

6) $f(x) = \cos x$, $g(x) = -x^2 + 2$;

7) $f(x) = x^4$, $g(x) = 5x + 2$;

8) $f(x) = \sqrt{x}$, $g(x) = \frac{x}{x-1}$;

9) $f(x) = \frac{1}{x}$, $g(x) = 2x^2 - 3x + 1$;

10) $f(x) = x^2$, $g(x) = \cos x$.

2. Побудувати зображення кривої, що задана рівнянням у декартовій системі координат. Встановити відображення осей, підписи, лінії сітки, відображення виразу рівняння на графічному полі.

1) $x^{\frac{2}{3}} + y^{\frac{2}{3}} = 1$

2) $x^3 + y^3 = 8xy$

3) $(x^2 + y^2 - 1)^3 = x^2y^3$

4) $(x^2 + y^2)^2 + 2x(x^2 + y^2) = y^2$

5) $(x^2 + y^2)^2 = 2(x^2 - y^2)$

6) $9x^2 + 25y^2 = 225$

7) $x^2 - 4y^2 = 8$

8) $y^2 = -2x$

9) $x^2 + 4y^2 + 10x + 8y + 28 = 0$

10) $x^2 + y^2 + 4x - 6y - 3 = 0$

3. Побудувати криві, задані параметрично. Дослідити вплив числових значень a та b на вигляд кривої.

1) $x(t) = 2a\,cos\,t + a\,cos\,3a\,t,$

$y(t) = 2a\,sin\,t + a\,sin\,3a\,t,\ \ t \in [0;\ 2\pi]$ (гіпоциклоїда);

2) $x(t) = a\,sin\,t, y(t) = b\,cos\,t, t \in [0;\ 2\pi]$ (еліпс);

3) $x(t) = t\,sin\,a\,t, y(t) = t\,cos\,a\,t, t \in [0;\ 5\pi]$ (спіраль);

4) $x(t) = sin\,a\,t, y(t) = cos\,b\,t, t \in [0;\ 2\pi]$ (фігури Лісажу);

5) $x(t) = 2a\cos t\,(1 + \cos t), y(t) = 2a\sin t\,(1 + \cos t), t \in [0;\ 2\pi]$

6) $x(t) = \frac{3at}{1+t^3}, y(t) = \frac{3at^2}{1+t^3}, t \in [-5\pi;\ 5\pi]$ (Декартів лист)

7) $x(t) = 2a\cos t, y(t) = a(1 - \cos 2t), t \in [0;\ 2\pi]$ (локон Аньєзі)

8) $x(t) = a\,sin^3\,t, y(t) = a\,cos^3\,t, t \in [0;\ 2\pi]$ (астроїда);

9) $x(t) = a\,(t - \sin t), y(t) = a\,(1 - \cos t), t \in [0;\ 6\pi]$ (циклоїда);

10) $x(t) = a\sqrt{2}\frac{t+t^3}{1+t^4},$

$y(t) = a\sqrt{2}\frac{t-t^3}{1+t^4}, t \in [-2\pi;\ 2\pi]$ (лемніската Бернуллі).

4. У полярних координатах побудувати лінії. Дослідити вигляд ліній в залежності від значень параметра a.

1) $r = a + sin\frac{5\phi}{3};$

2) $r(\phi)= a\phi, \phi \in [0;\ 5\pi]$ (спіраль Архімеда);

3) $r(\phi)= e^{\phi}$ (логарифмічна спіраль);

4) $r(\phi)= a/\phi$ (гіперболічна спіраль);

5) $r(\phi)= a(1 - \cos\phi), \phi \in [0;\ 2\pi]$ (спіраль Архімеда);

6) $r(\phi)= \sqrt{2a^2\cos 2\phi}, \phi \in [0;\ 2\pi]$ (лемніската Бернуллі);

7) $r(\phi)= \frac{3a\cos\phi\sin\phi}{\cos^3\phi+\sin^3\phi}, \phi \in [0;\ 2\pi]$ (Декартів лист);

8) $r(\phi)= \frac{2a\sin^2\phi}{\cos\phi}, \phi \in [0;\ 2\pi]$ (цисоїда);

9) $r(\phi)= -\frac{a\ \cos 2\varphi}{\cos\phi}, \phi \in [0;\ 2\pi]$ (строфоїда);

10) $r(\phi)= a\sqrt{cos\,2\,\phi}\ ,\ \ \phi \in [0;\ 2\pi].$

5. Зобразити графічну ілюстрацію розв'язування рівняння.

1) $5^x = 6 - x$;

2) $\left(\frac{1}{3}\right)^x = 3$;

3) $\sin x = x - \pi$;

4) $\cos x = x^2 + 1$;

5) $(x - 3)^3 = 5 - \sqrt{x + 1}$;

6) $\sqrt{x + 2} = -x^2 - 4x - 5$;

7) $(x - 2)^2 = \sqrt{x} + 2$;

8) $\frac{3}{x} = 3x$;

9) $2x - 6 = \sqrt{x}$;

10) $\sin\frac{\pi x}{2} = x^2 - 2x + 2$.

6. Знайти корінь рівняння графічним способом з точністю до тисячних. Перевірити знайдене значення за допомогою процедури `find_root()`.

1) $ln^2(x - 1) = 3\,cos\,2\,x + 1$;

2) $\frac{10}{1+x^2} = 2\,sin\,2\,x + x$;

3) $\frac{10x-2}{3+x^2} = 2\,cos\,2\,x + \sqrt[4]{x}$;

4) $\frac{10}{1+x^2} = 2\,cos\,2\,x + x$;

5) $sin\,x \cdot \sqrt{81 - x^2} = 5arctg(x)$;

6) $\sqrt{25 - x^2} = arctg(2x)$;

7) $arctg(2x) - \frac{(x-4)^4}{5} + sin^2\,5\,x = 0$;

8) $sin\,x^2 \cdot \sqrt{81 - x^2} = 5e^{-x^2}$;

9) $\frac{x^2-9}{x^2+4} = \sqrt{x^2 + 1} \cdot e^{x\,cos\,x}$;

10) $\frac{10}{1+x^2} = 3\,cos\,2\,x + 2x$.

7. Побудувати пряму, що проходить через точки M_1 та M_2. Позначити дані точки на прямій, зобразити напрямний та нормальний її вектори. В заголовку графіка написати рівняння прямої M_1M_2.

1) $M_1(0;\ 2)$, $M_2(3;\ 4)$

2) $M_1(2;\ 2)$, $M_2(4;\ 3)$

3) $M_1(1;\ 2)$, $M_2(4;\ 3)$

4) $M_1(2;\ 2)$, $M_2(1;\ 3)$

5) $M_1(0; 1)$, $M_2(3;\ 5)$

6) $M_1(2;\ 1)$, $M_2(4;\ 3)$

7) $M_1(2;\ 2)$, $M_2(1;\ 5)$

8) $M_1(-2;\ -1)$, $M_2(4;\ 3)$

9) $M_1(-1;\ -2)$, $M_2(2; 2)$

10) $M_1(-2;\ -1)$, $M_2(4;\ 5)$.

8. Зобразити множину точок, які задовольняють нерівність:

1) $|y| - \cos x \geq 0$;

2) $2x + 3y - 6 \leq 0$ і $2x + 3y + 6 \geq 0$

3) $|y| - \cos x \leq 0$;

4) $|x| + |y| < 3$;

5) $|x| + |y| \geq 5$;

6) $x^2 + y^2 \geq 5$;

7) $x^2 + y^2 < 2$;

8) $\sin \pi x - \sin \pi y \geq 0$;

9) $(y^2 - \arcsin^2(\sin x))(y^2 - \arcsin^2(\sin(x + \dfrac{\pi}{6}))) < 0$;

10) $(y - sin\, x)(y + sin\, x) < 0$.

9. Виконайте дослідження. Підготуйте в Maxima довідковий матеріал з ілюстраціями про властивості цієї лінії.

9.1. Знайдіть відомості про криву з назвою «Декартів лист». Створіть графік кривої та ілюстрацію значення параметра. Оформлення графіка обрати самостійно.

9.2. Знайдіть відомості про ланцюгову лінію (катенарію), побудуйте її графік. Виконайте дослідження вигляду кривої в залежності від параметра.

9.3. Рівняння $r = R\sin k\varphi$ визначає k-пелюсткову троянду. Дослідити вигляд кривої для різних значень k. Розглянути випадки, коли k – ціле парне, ціле непарне число, раціональне число, ірраціональне число. Що означає параметр R?

10. Задана функція $f(x) = \sin x + \sin 2x$ Визначте дискретний аргумент в діапазоні від 0 до 2π з кроком 0,1. Створіть список значень функції $f(x)$ для вказаного дискретного аргументу. Побудуйте точковий графік функції $f(x)$ для вказаного дискретного аргументу.

11. В декартовій системі координат створити малюнок, використовуючи процедури для побудови графічних об'єктів (трикутник, прямокутник, многокутник, коло, еліпс та ін.) та засоби налаштування їх параметрів.

Контрольні питання

1. Який запис функції draw2d() є правильним?

a) draw2d(graph_obj1, graph_obj2, opts);

b) draw2d(opts1, graph_obj1, opts2, graph_obj2);

c) draw2d(opts1, opts2, graph_obj1, graph_obj2);

d) draw2d(graph_obj1, graph_obj2, opts1, opts2);

2. Яка команда в Maxima призначена для побудови графіка функції $f(x) = x^2$ на відрізку [-2;2]?

a) draw(explicit(x^2, x, -2, 2));

b) explicit(x^2, x, -2, 2);

c) draw2d(explicit(x, x^2, -2, 2));

d) draw2d(explicit(x^2, x, -2, 2));

3. Встановіть відповідність між опцією функції draw2d() (1-4) та її призначенням (a-e):

1) proportional_axes=xy	a) заголовок графічного поля;
2) grid=true	b) назва графічного об'єкта;
3) title= "text"	c) напис на графічному полі;
4) key= "text"	d) однаковий масштаб по осях;
	e) лінії сітки в графічному полі.

4. Встановіть відповідність між опцією функції draw2d() (1-4) та її призначенням (a-e):

1) xrange=[min, max]	a) назва осі Ox;
2) xtics_axis=true	b) підписи на шкалі осі Ox за замовчуванням;

3) xtics=auto c) межі графічного поля вздовж осі Ox;

4) xlabel= “text” d) наявність шкали з підписами на осі Ox;

e) зображення осі Ox на графічному полі.

5. Встановіть відповідність між опцією функції draw2d() (1-4) та її призначенням (a-e):

1) yaxis=true a) пунктирна лінія осі Oy;

2) yaxis_width=2 b) товщина лінії осі Oy;

3) yaxis_type=dots c) колір осі Oy;

4) yaxis_color=red d) колір шкали по осі Oy;

e) зображення осі Oy на графічному полі.

6. Яка опція використовується у функції draw2d() для налаштування гладкості графіка неявно заданої функції?

a) ntics; b) ip_grid; c) x_voxel; d) y_voxel;

7. Яка задати у функції draw2d() напис точки A_1 у місці графічному поля з координатами (3; 4)?

a) key(“A1(3;4)”);

b) label([“A_{1}”, 3, 4]);

c) key([“A_{1}”, 3, 4]);

d) label(“A1(3;4)”);

8. Як скасувати налаштування опцій користувача?

a) get_draw_defaults();

b) set_draw_cancel();

c) set_draw_defaults();

d) get_draw_cancel();

9. Вкажіть графічний об’єкт для зображення вектора:

a) points(); b) vector(); c) region(); d) polygon;

10. Вкажіть графічний об’єкт для зображення многокутника:

a) points(); b) rectangle(); c) region(); d) polygon;

2.2.3 Тривимірні графічні об'єкти

За допомогою процедур пакету draw будують тривимірні графічні об'єкти, подані в таблиці. Процедури побудови тривимірних графічних об'єктів подібні до процедур у двовимірній системі координат. Відмінність полягає у використанні в процедурах третьої координати точок простору вздовж осі аплікат. У просторі немає полярної системи координат, але є циліндрична та сферична система координат, в яких рівняння поверхонь або кривих може мати простіший вигляд, ніж у прямокутній системі координат.

`explicit(f(x,y),x,x1,x2,y,y1,y2)` – графік явно заданої функції $f(x,y)$ на проміжках $x \in [x_1; x_2]$ та $y \in [y_1; y_2]$. draw_realpart, xu_grid, yv_grid, line_type, line_width, key, wired_surface, enhanced3d and color.

`implicit(equation,x,x1,x2,y,y1,y2,z,z1,z2)` – графік неявно заданої функції від двох змінних або рівняння поверхні, що задана рівністю `equation`, на проміжках $x \in [x_1; x_2]$, $y \in [y_1; y_2]$, $z \in [z_1; z_2]$.

`parametric(x(t),y(t),z(t),t,t1,t2)` – лінія, задана параметрично, з параметром $t \in [t_1; t_2]$.

`parametric_surface(x(u,v),y(u,v),z(u,v),u,u1,u2,v,v1,v2)` – поверхня, задана параметрично, з параметрами $u \in [u_1; u_2]$ та $v \in [v_1; v_2]$.

`cylindrical(r(z,φ),z,z1,z2,φ,φ1,φ2,v,v1,v2)` – поверхня, задана рівнянням `r(z,φ)`, у циліндричних координатах, з азимутом $\varphi \in [\varphi_1; \varphi_2]$. xu_grid, yv_grid, line_type, key, wired_surface, enhanced3d and color

`spherical(r(φ,v),φ,φ1,φ2)` – поверхня, задана рівнянням `r(φ,v)` у сферичних координатах.

`points([x1,x2,…],[y1,y2,…],[z1,z2,…])` – точки з координатами $(x_i;y_i;z_i)$, *i=1,2,…*, що задані списком значень абсцис, списком значень ординат та списком значень аплікат.

`points(([[x1,y1,z1],[x2,y2,z2],…])` – точки з координатами $(x_i;y_i;z_i)$, *i=1,2,…*, що задані списком трійок відповідних координат.

`label([“text”,x,y,z],…)` – напис в позиції точки з координатами $(x;y;z)$, вирівнювання, орієнтація задається опціями.

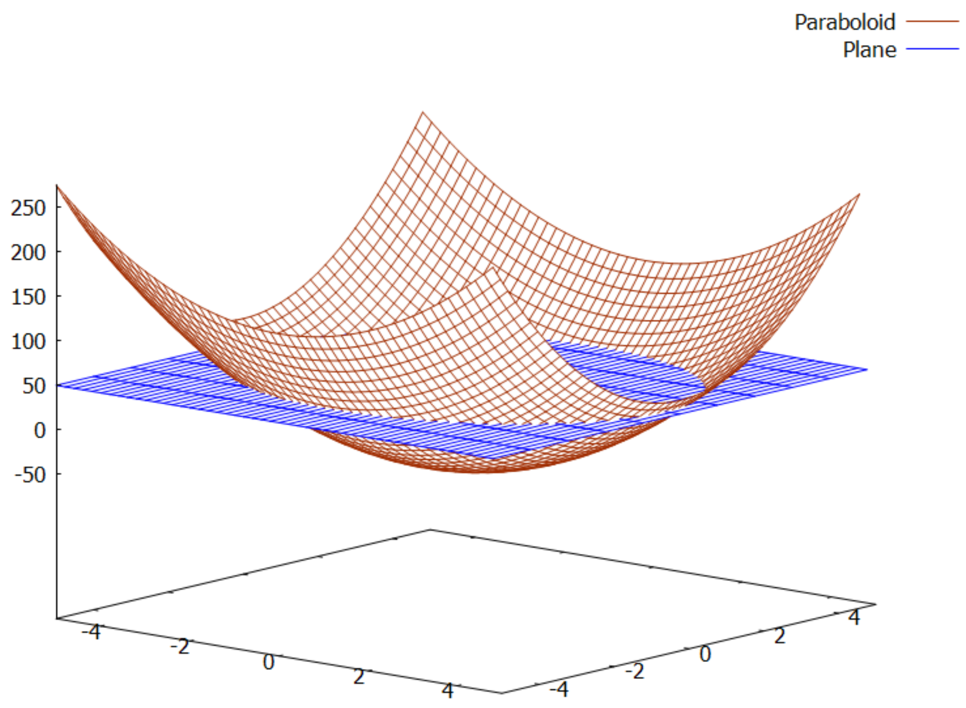
`vector([x,y,z],[dx,dy,dz])` – вектор з початком в точці $(x;y;z)$ та координатами $(dx\ ;dy;\ dz)$.

`mesh([x1,y1,z1],[x2,y2,z2],…)` – просторова сітка з вузлами в точках з координатами (x1,y1,z1), (x2,y2,z2),…

Приклад 2.25. Зобразити перетин еліптичного параболоїда $z = 5x^2 + 8y^2 - 50$ і площини, яка претинає вісь аплікат в $\mathrm{z} = 50$ і паралельна координатній площині *Oxy*.

Для побудови поверхні еліптичного параболоїда, що заданий як явно виражена функція від двох змінних, використовуємо процедуру `explicit()`. Оскільки дана площина паралельна координатній площині *xOy* і претинає вісь аплікат в $\mathrm{z} = 50$, то рівняння площини задається рівністю $\mathrm{z} = 50$, а отже для її побудови використовуємо процедуру `explicit()`.

```
draw3d(key = "Paraboloid", /·назва графіка ·/
        color = "#a02c00", /·колір параболоїда·/
        /·еліптичний параболоїд:·/
        explicit(5·(x^2)+8·(y^2)−50, x,−5,5,y,−5,5),
        yv_grid = 10, /·число ліній сітки вздовж осі y ·/
        color = blue, /·колір площини·/
        key = "Plane", /·назва графіка ·/
        explicit(50, x,−5,5,y,−5,5),  /·площина z=50·/
        surface_hide=true)$  /·приховати невидиму частину поверхні ·/
```



В цьому прикладі використовуються опції `color` і `key` для задання кольору і підпису графіків. Ці опції загальні для всіх функцій пакету `draw`.

Крім загальних опцій до тривимірних графіків застосовуються деякі спеціальні опції. Зокрема для налаштування гарного вигляду поверхонь в прикладі використовувалися опції `yv_grid` для встановлення числа ліній сітки по осі *y* та опція `surface_hide=true`, щоб поверхні не зливалися одна з одною, а невидима їх частина була прихованою.

2.2.4 Опції для тривимірної графіки

`xu_voxel,yv_voxel` – частота дискретизації (явно задані поверхні, за замовчуванням 10)

`x(yz)_voxel` – частота дискретизації (графічний об'єкт region, неявно задана поверхня, за замовчуванням 10)

`xu_grid=n,yv_grid=n` – число ліній сітки вздовж відповідної осі (явна або параметрична, за замовчуванням 30)

`proportional_axes=xyz` – однаковий масштаб вздовж осей

`wired_surface=true/false` – використовується, щоб відобразити або приховати сітку на поверхні, якщо задане забарвлення поверхні `enchanced3d=true`.

`enhanced3d=true/false` – градієнтне забарвлення поверхні

`palette=[r,g,b]` – кольори градієнтного забарвлення поверхні, де `r,g,b` – назви або коди кольорів градації. Якщо `palette=gray` – поверхня в градаціях сірого кольору. За замовчуванням `palette=color`.

`surface_hide=true/false` – приховати невидиму частину поверхні

`contour=value` – зображення контурних ліній. За замовчуванням `contour=none`, тобто лінії не будуються. Також можливі значення: `base` – контурні лінії проектуються на площину Oxy; `surface` – контурні лінії будуються на поверхні; `both` – контурні лінії зображуються і на поверхні і в проекції на площину Oxy; `map` – проектуються на площину Oxy, вигляд зверху.

`contour_levels=n` – число контурних ліній, які будуються через рівні проміжки (за замовчуванням 5). Якщо `contour_levels=[n1,k,n2]`, то контурні лінії будуються від значення n1 до n2 з кроком k; якщо `contour_levels={n1,n2,…}`, контурні лінії будуються для значень n1,n2,…

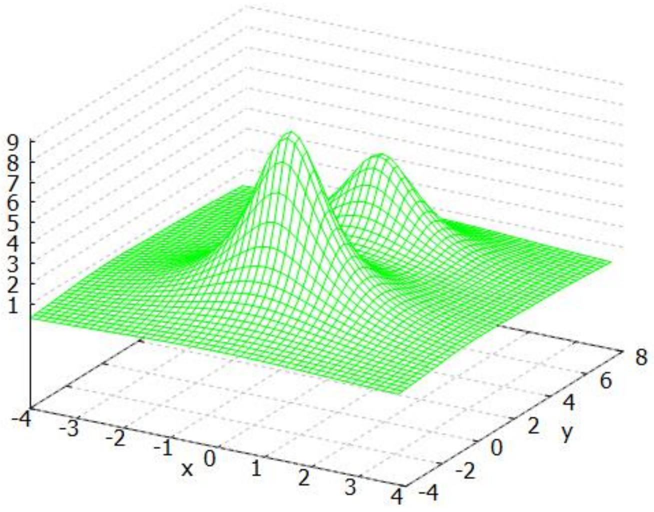
`view=[φ,θ]` – пара кутів (в градусах), за якою задають орієнтацію координатних осей Oxyz у паралельній проєкції просторового зображення на графічному полі.

Розглянемо більш детально опції для тривімірної графіки, особливості їх налаштування та приклади використання в графічних процедурах.

Для зображення тривимірних об’єктів використовувати wx-аналог графічних процедур не дуже зручно, оскільки зображення буде статичне. Якщо застосовувати процедуру `draw3d` (без wx), графік поверхні з’являється в окремому вікні, де можна його повертати та спостерігати з різних сторін, тобто змінювати напрям проєкції тривимірного зображення на площину.

Приклад 2.26. Побудувати поверхню $z = \frac{5}{(y-5)^2+x^2+1} + \frac{9}{y^2+x^2+1}$. Зобразити лінії рівня $z(x,y)=c$, де c=1,2,…9, встановити градієнтне забарвлення поверхні. Показати вигляд поверхні зверху та у фронтальній проєкції.

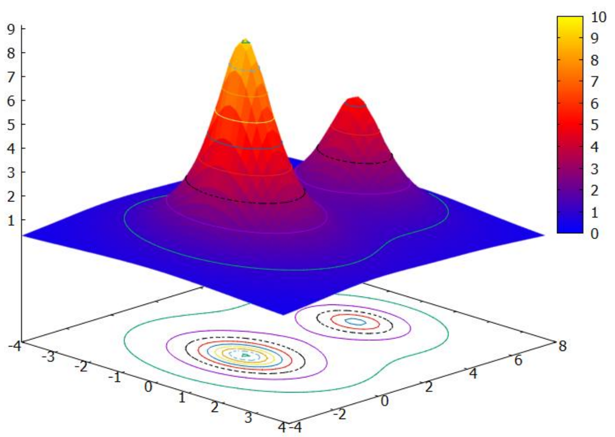
```
load(draw)$ /* завантажуємо графічний пакет */
/* Позначимо вираз функції через f: */
f: 5/((y-5)^2+x^2+1)+9/(y^2+x^2+1)$
/*Створюємо графічний об’єкт явно заданої функції: */
g: explicit(f,x,-4,4,y,-4,8)$
/*Описуємо графічну процедуру: */
draw3d(surface_hide=true,/*невидима частина поверхні прихована*/
        xu_grid=50,yv_grid=50,/*збільшуємо частоту дискретизації*/
        xlabel="x",ylabel="y",/*підписи осей*/
        grid=true, /*координатна сітка площини xOy*/
        color=green, /*колір поверхні*/
        g)$ /* графічний об’єкт поверхні*/
```



За замовчуванням (коли `enhanced3d=false`) поверхня зображується у вигляді координатної сітки, колір якої визначається опцією `color`. Для забарвлення поверхні встановлюємо опцію `enhanced3d=true`. Колір забарвлення поверхні задається опцією palette. За замовчуванням `palette=color` і забарвлення відбувається в градації кольорів спектру. Якщо записати `palette=[blue,red,yellow]`, то колір поверхні буде градієнтом відтінків синього, червоного, жовтого. Також зобразимо на графіку і в проекції на площину Oxy лінії рівня.

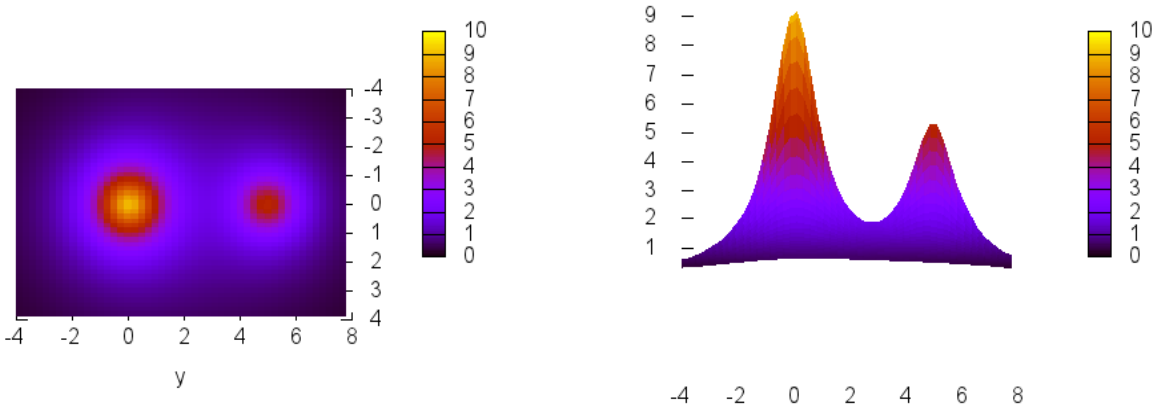
Записуємо програмний код у вигляді:

```
draw3d(surface_hide=true,xu_grid=50,yv_grid=50,
    enhanced3d=true,palette=[blue,red,yellow],
    contour_levels=[1,1,9],contour=both,g)$
```



Дану поверхню можна оглядати зверху та у фронтальній проєкції. Для цього скористаємося опцією `view[φ,υ]`. Щоб зобразити обидва графіки на одному графічному полі, скористаємося процедурою `gr3d()`.

```
s1:gr3d(axis_3d=false,
     proportional_axes=xyz, ylabel="y",
     xu_grid=50,yv_grid=50,
     enhanced3d=true,
     view=[0,90],g) $ /·проєкція вигляд зверху·/
s2: gr3d(axis_3d=false,
     proportional_axes=xyz,
     ylabel="y", zlabel="z",
     xu_grid=50,yv_grid=50,
     enhanced3d=true,
     view=[90,90],g)$ /·фронтальна проєкція·/
draw(s1,s2,columns=2, dimensions=[1000,800]);
```



Приклад 2.27. Задано дві точки $M_1(3;-1;2)$ і $M_2(4;-2;-1)$. Скласти рівняння площини, яка проходить через точку M_1 перпендикулярно вектору $\overrightarrow{M_1M_2}$. Побудувати площину, вектор і точки, позначити точки на малюнку, записати їх координати.

Загальне рівняння площини, яка проходить через точку $M(x_0;y_0;z_0)$ має вигляд $A(x-x_0)+B(y-y_0)+C(z-z_0)=0$, причому $\vec{n}(A;B;C)$ – нормальний вектор цієї площини (вектор, перпендикулярний до площини).

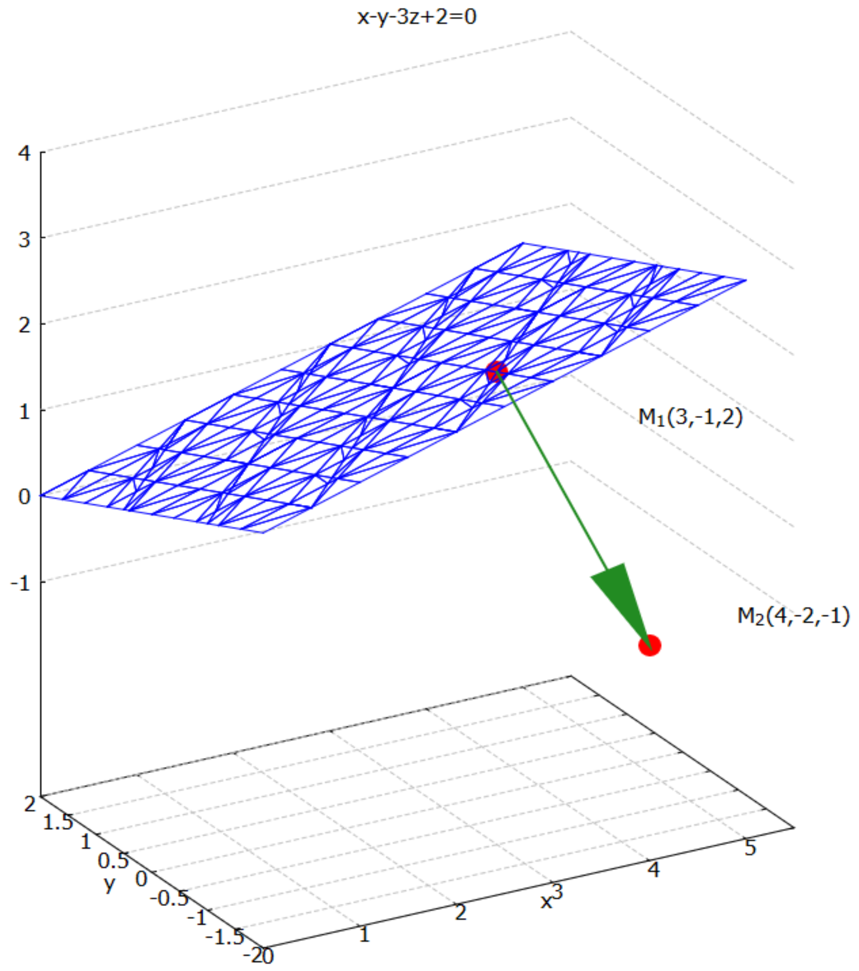
Знаходимо координати нормального вектора:

$\overrightarrow{M_1M_2}=(4-3;\ -2+1;\ -1-2)=(1;\ -1;\ -3).$

Підставляємо координати вектора $\overrightarrow{M_1 M_2}$ та координати точки M_1 в загальне рівняння площини: $1 \cdot (x-3) - 1 \cdot (y+1) - 3 \cdot (z-2) = 0$. Маємо $x - y - 3z + 2 = 0$ – загальне рівняння даної площини. Оскільки площина задана загальним рівнянням, то для побудови її графічного зображення скористаємося процедурою `implicit()`. Спочатку позначимо змінними процедури побудови заданих графічних об'єктів. Далі в процедурі draw3d() будемо використовувати імена цих змінних.

```
M: points([[3,-1,2],[4,-2,-1]]); /·задані  точки ·/
plan: implicit(x-y-3·z+2=0,x,0,5,y,-2,2,z,-1,4); /·рівняння площини·/
normal: vector([3,-1,2],[1,-1,-3]); /·вектор·/
text: label(["M_{1}(3,-1,2)",5,-1,1],
        ["M_{2}(4,-2,-1)",5.5,-2,-1]); /·написи біля точок·/

draw3d(proportional_axes=xyz,
      title="x-y-3z+2=0", /·заголовок графічного поля·/
      xlabel="x",ylabel="y",  grid=true, /·позначення осей, сітка·/
      color=red, point_size=2, point_type=7, /·властивості точок·/
      M, /·точки·/
      color=black, /·колір написів·/
      text, /·написи біля точок·/
      color=blue, /·колір площини·/
      plan, /·площина·/
      color=forest_green, /·колір вектора·/
      head_angle=10, /·кут стрілки вектора·/
      head_length=0.5, /·величина стрілки вектора·/
      line_width=2, /·ширина лінії вектора·/
      normal)$ /·вектор·/
```

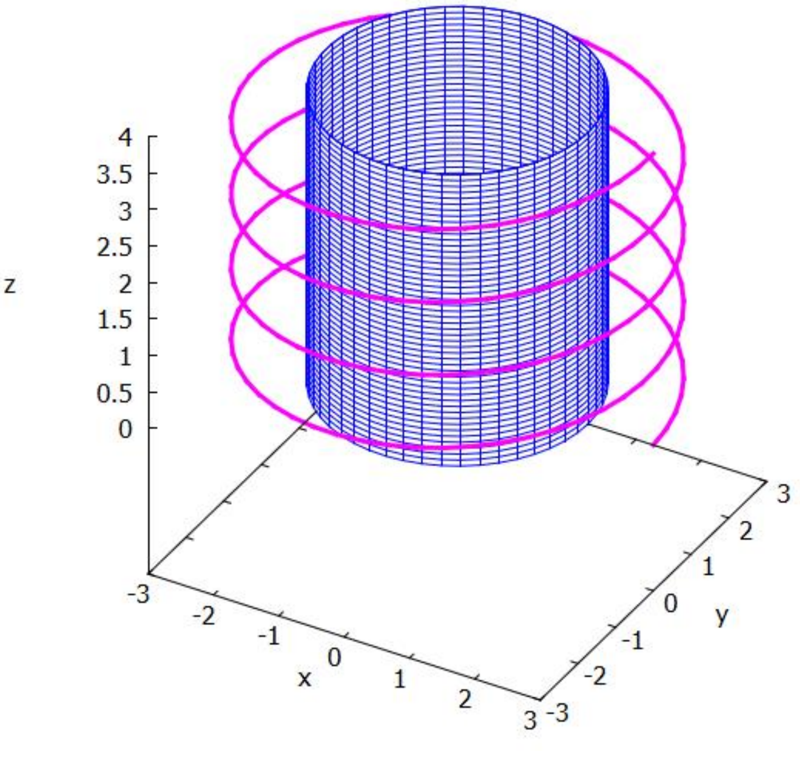


Приклад 2.28. Побудувати циліндр та циліндричну гвинтову лінію, що задані параметричними рівняннями.

Параметричні рівняння циліндра:	*Параметричні рівняння гвинтової лінії:*
$x(t;v) = 2\cos t,$	$x(t) = 3\cos t,$
$y(t;v) = 2\sin t,$	$y(t) = 3\sin t,$
$z(t;v) = v,$	$z(t) = \frac{t}{2\pi}, t \in [0;8\pi],$
$t \in [0;2\pi], v \in [0;4]$	

Позначимо змінними процедури для побудови циліндра та гвинтової лінії, що задані параметричними рівняннями поверхні та кривої лінії у просторі. Тоді в процедурі `draw3d()` використовуємо тільки імена цих змінних.

```
/*змінна для позначення процедури побудови циліндра:*/
cilind: parametric_surface(2*cos(t), 2*sin(t), v, t, 0, 2*%pi, v, 0, 4)$
/*змінна для позначення процедури побудови гвинт. лінії*/
spiral: parametric(3*cos(t), 3*sin(t), t/2/%pi, t, 0, 8*%pi)$
draw3d(nticks=200, surface_hide=true, /*плавність ліній, поверхня у вигляді сітки*/
    proportional_axes=xyz, /*однаковий масштаб вздовж осей*/
    xlabel="x",ylabel="y", zlabel="z", /*підписи координатних осей*/
    cilind, /*циліндр*/
    color=magenta, line_width=3, /*колір і ширина гвинтової лінії*/
    spiral); /*гвинтова лінія*/
```



Розглянемо завдання і побудову поверхонь в циліндричній та сферичній системі координат.

Циліндрична система координат

Циліндрична система координат – тривимірна система координат, кожна точка якої задається двома полярними координатами на перпендикулярній проєкції деякої фіксованої площини та відстанню (зі знаком) від цієї площини. Тобто циліндрична система координат є розширенням полярної системи координат шляхом додавання третьої координати (як правило z), якою задається висота точки над площиною (рис. 2.3).

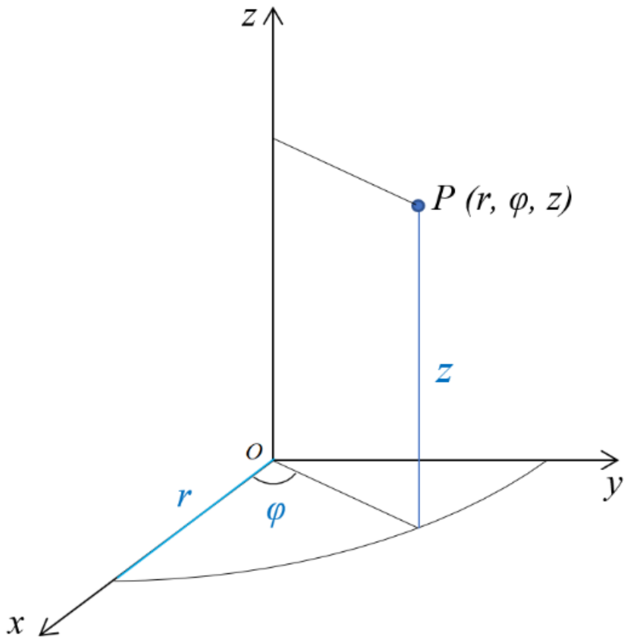


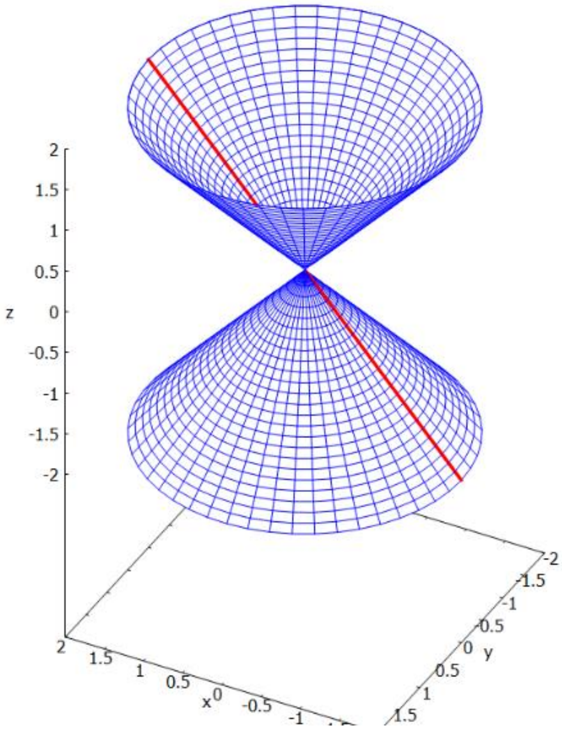
Рис. 2.3 Циліндрична система координат

Координатами точки $P(r,\varphi,z)$ є: r – радіальна відстань від осі до точки *P*; азимут φ – кут між напрямом відліку на вибраній площині та променем, що проходить з точки початку координат до проекції *P* на площині; висота *z* дорівнює відстані від обраної площини до точки *P* (рис.).

Циліндричні координати зручно використовувати під час аналізу поверхонь, симетричних відносно якої-небудь осі, наприклад, якщо вісь *z* взяти в якості осі симетрії, то круговий циліндр висотою *c* в прямокутних декартових координатах має рівняння $x^2 + y^2 = c^2$, а в циліндричних – дуже просте рівняння $r(\varphi,z) = c$, $c = const$ – радіус основи циліндра.

Приклад 2.29. Побудувати конічну поверхню в циліндричній системі координат, задану рівнянням $r(\varphi,z) = z$. Червоним кольором зобразити твірну лінію конічної поверхні.

```
draw3d(nticks=200,surface_hide=true,
  proportional_axes=xyz,
  xlabel="x",ylabel="y", zlabel="z", /·підписи координатних осей·/
  cylindrical(z,z,−2,2,p,0,2·%pi),/·конічна поверхня·/
  color=red,line_width=3,/·колір та ширина твірної·/
  parametric(t,0,t,t,−2,2));/·твірна лінія (параметричне рівняння)·/
```



Сферична система координат

Сферичними координатами називають систему координат для відображення геометричних властивостей фігури в трьох вимірах за допомогою задання трьох координат (r, θ, φ), де:

r – відстань від початку координат до заданої точки *M,* $r \geq 0$;

θ – зеніт, кут між віссю z і відрізком, що з'єднує початок системи координат і точку *M, (*$0 \leq \theta \leq 180°$);

$\varphi \leq 360°$– азимут, кут між віссю *x* і проекцією відрізка, що з'єднує початок координат з точкою *M*, на площину *xy,* $(0 \leq \varphi \leq 360°)$ (рис. 2.4).

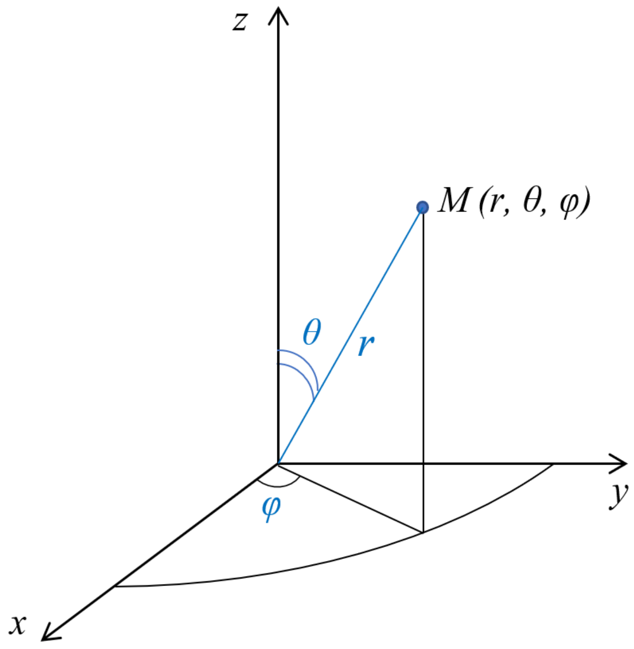
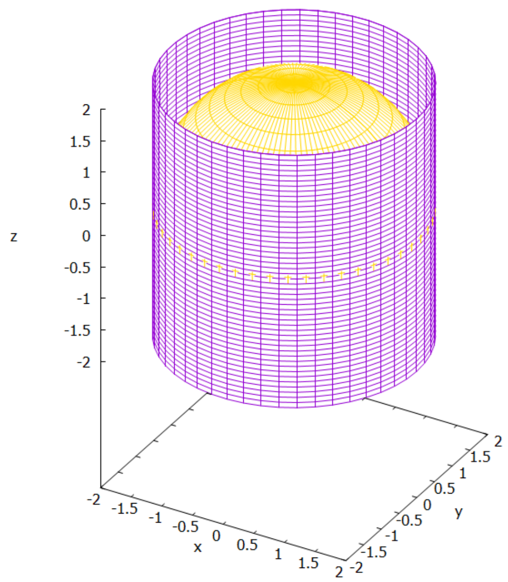


Рис. 2.4 Сферична система координат

Рівняння сфери в сферичній системі координат має вигляд $r(\theta, \phi) = c$, $c = const$ – радіус сфери.

Приклад 2.30. Побудувати зображення сфери, вписаної в циліндр, використовуючи їх рівняння в сферичній і циліндричній системах координат.

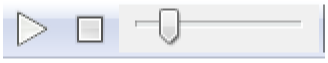
```
sfera: spherical(2, φ, 0, 2·π, θ, 0, 2·π)$
cilind: cylindrical(2, z, −2, 2, φ, 0, 2·π)$
draw3d(nticks=200, surface_hide=true, proportional_axes=xyz,
    xlabel="x",ylabel="y", zlabel="z", /·підписи координатних осей·/
    color=gold, sfera, /·колір сфери, сфера·/
    color=dark_violet, cilind); /·колір циліндра, циліндр·/
```



2.3 Анімація зображень в середовищі програми Maxima

Наочність графічного подання абстрактних математичних співвідношень, результатів моделювання різних об’єктів і явищ та їх дослідження суттєво підвищується за умови використання засобів анімації зображення, за допомогою яких можна спостерігати явища і процеси в динаміці їх перетворення.

Можливість створення анімацій – це специфічна особливість графічної оболонки wxMaxima. Процес показу анімації управляється за допомогою послуг панелі інструментів wxMaxima:



Щоб активізувати ці елементи управління, необхідно клацнути мишкою в області графічного вікна з анімацією. Також управління анімацією можливе за допомогою послуг контекстного меню графічного вікна або прокручування коліщатка миші.

Для створення анімації в інтерфейсі *wxmaxima* використовуються процедури:

`with_slider(k, list, expr, opts)`, – де `k` – параметр, `list` – список значень параметра, `expr` – вираз функції, гафічний об’єкт; `opts` – опції графіка функції. Правила запису даної функції аналогічні plot-процедурам.

`wxanimate_draw(k, list, opts, expr)`, де k – параметр, list – список значень параметра, opts – опції графіка функції, expr – вираз функції, графічний об’єкт. Запис аргументів даної функції аналогічний draw-процедурам.

`with_slider_draw3d(k, list, opts, expr)` – будується послідовність тривимірних графічних об’єктів, залежних від параметра k, list – список значень параметра, opts – опції графіка функції, expr – вираз функції, графічний об’єкт.

Якщо параметр анімації – параметр аналітичного виразу функції, то анімація складається з послідовної демонстрації графіків відповідно до кожного значення параметра. У випадку, якщо діапазон значень змінної графіка функції залежить від параметра анімації, то на анімаційному зображенні демонструється плавна побудова лінії у відповідності до значення змінної.

Слід зазначити, що за допомогою даних процедур створюється анімація, яка може бути відтворена тільки в інтерфейсі *wxmaxima*.

Анімаційне зображення легко зберегти у вигляді файлу gif, який буде створений в поточній папці. Для цього потрібно в процедурі створення анімації визначити опції: `file_name="name"` – задається ім'я файлу, `terminal='animated_gif` – вказується тип файлу, `delay=n` – визначається час затримки між кадрами анімації, за замовчуванням `delay=5`. Дані опції є глобальними.

Розглянемо приклади.

Приклад 2.31. Створити анімаційну демонстрацію розкладу функції $y = sin\, 3\, x$ в точці $x_0 = 0$ в ряд Тейлора в залежності від степеня a.

Спочатку задаємо функцію та початкову точку:

```
f:sin(3*x); x0:0;
```

Параметром анімації буде степінь a полінома Тейлора відповідного розкладу даної функції. Нехай a змінюється від 1 до 20. Тоді записуємо:

```
wxanimate_draw(
        a,makelist(a,a,1,20), /*послідовність значень степеня
/* визначаємо налаштування вигляду координатної площини */
        dimensions=[900,500],
        proportional_axes=xy, yrange=[-2.1,2.1],
        xtics_axis=true, ytics_axis=true,
        grid=true, xaxis=true, yaxis=true,
/* графік даної функції */
        key="f(x)", line_width=2,
        explicit(f,x,-%pi,%pi),
/* графік функції відповідного полінома Тейлора*/
        key="T", color=red, line_width=2,
        explicit(taylor(f,x,x0,a),x,-%pi,%pi),
```

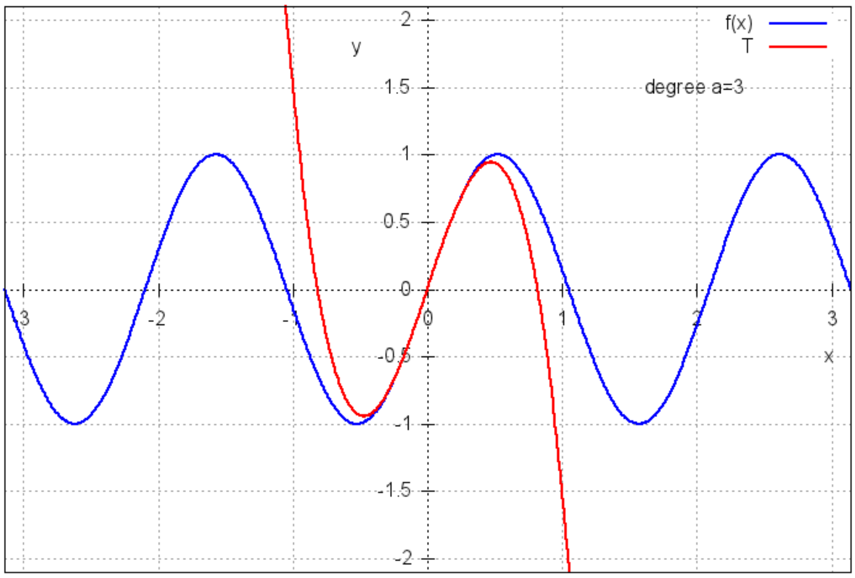
/* написи на графіку */

```
        color=black,
        label([sconcat("degree a=",a),2,1.5]),
        label(["y",-0.5,1.8]), label(["x",3,-0.5]));
```

В результаті одержимо анімаційне зображення, окремі кадри якого зображені на рис. 20.

Для визначення поліному Тейлора в даній програмі використовується команда `taylor(f,x,x0,a)`, де `f` – дана функція, `x` – змінна, `x0` – початкова точка, `a` – степінь.

a=3



a=7

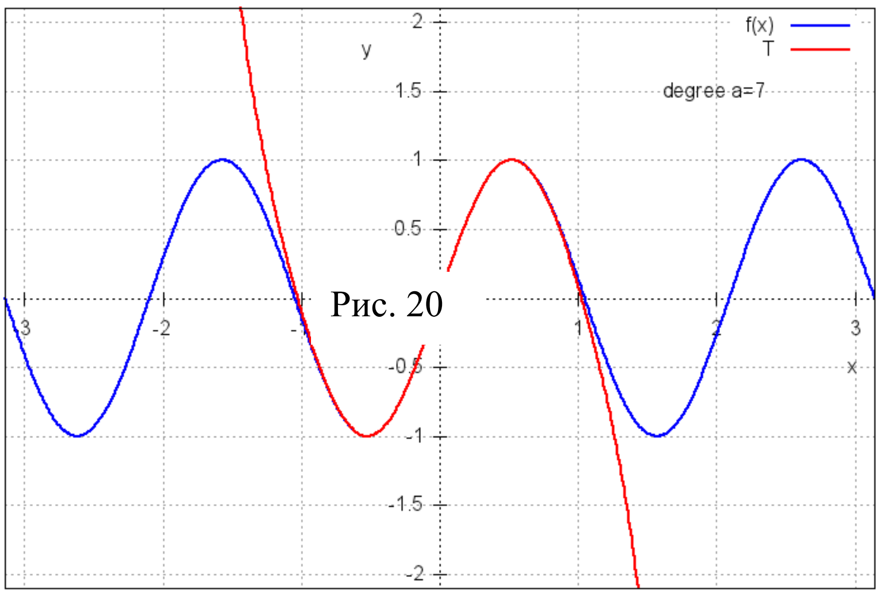
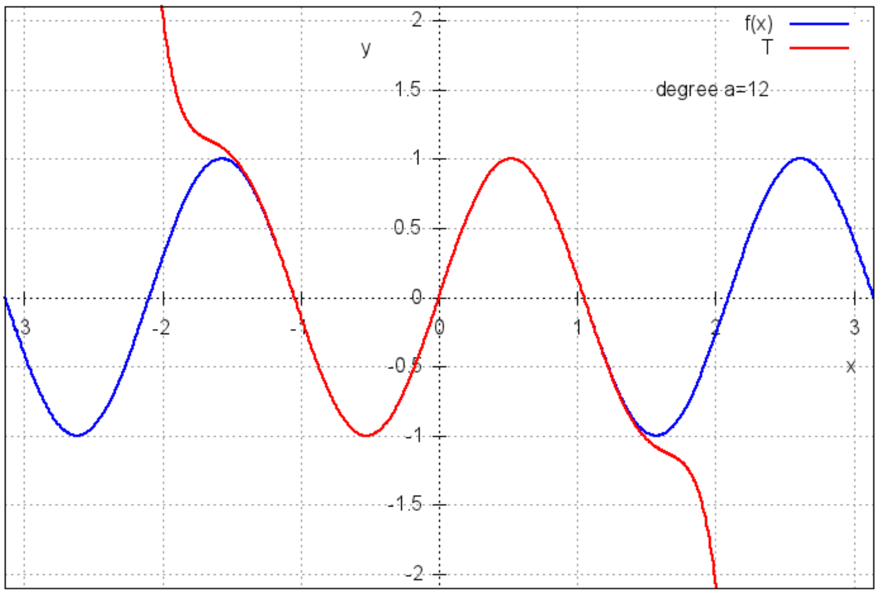
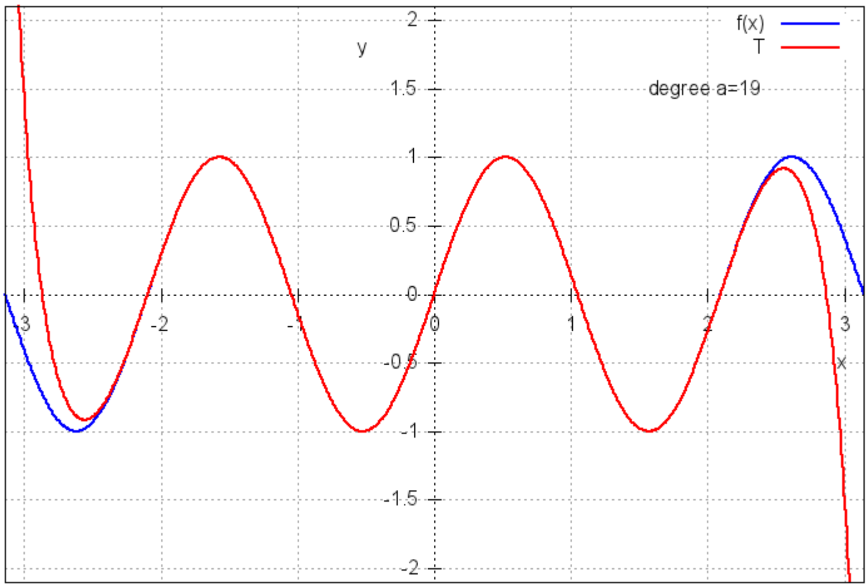


Рис. 20

a=12



a=19



В даному прикладі легко змінити функцію та початкову точку і розглянути подібні приклади.

Розглянемо приклад, коли від параметра анімації залежать межі побудови графіка. В цьому разі будемо спостерігати плавну побудову лінії графіка функції в залежності від значення аргументу.

Приклад 2.32. Створити анімаційне зображення побудови графіка функції в полярних координатах $r=\dfrac{1}{2}+\sin\dfrac{5t}{3},\ t\in[0;6\pi]$.

```
with_slider_draw(
  ang, makelist(i,i,0,60)·π/10, /·параметр та список його значень ·/
  proportional_axes=xy, xaxis=true, yaxis=true,
  xrange = [−2,2], yrange = [−2,2], nticks = 200,
  key=sconcat("ang=",ang), /·значення параметра в написах до графіка·/
  line_width = 2, color = blue,
  polar(1/2+sin(5·t/3),t,0,ang));
```

В результаті виконання даного програмного коду одержуємо анімаційне зображення побудови графіка, окремі кадри якого зображені на рис. 2.4.

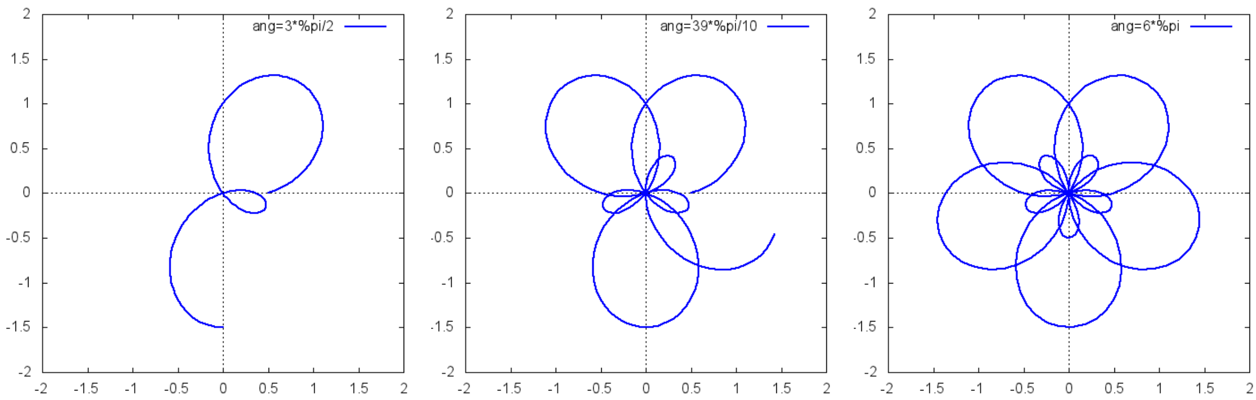


Рис. 2.4 Кадри анімаційної побудови графіка

Рух точки вздовж кривої

Використовуючи процедури створення анімації та процедуру створення списку `makelist()`, можна відтворити рух точки вздовж деякої кривої. Для прикладу розглянемо анімацію, за допомогою якої імітується рух тіла, кинутого під певним кутом до горизонту.

Приклад 2.33. М’яч, кинутий з висоти 2 м від землі з початковою швидкістю $\upsilon_0 = 20$ м/с під кутом 45º. Створити анімацію, за допомогою якої буде імітуватися рух м’яча у вигляді рухомої точки на кривій.

Нехай тіло кинули з точки M$(x_0; y_0)$ з початковою швидкістю $\vec{v}_0$, напрямленою під кутом α до горизонту. Вісь *Ox* спрямуємо горизонтально, а *Oy* – вертикально вгору (рис. 2.5).

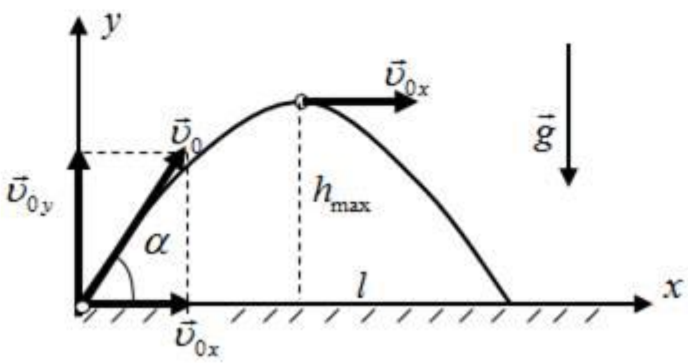


Рис. 2.5 Рух тіла, кинутого над землею

Проєкції вектора швидкості на осі *Ox* та *Oy* відповідно дорівнюють:

$\upsilon_{0x} = \upsilon_0 \cdot cos\,\alpha; \upsilon_{0y} = \upsilon_0 \cdot sin\,\alpha.$

Оскільки на тіло діє сила тяжіння, то під час його руху змінюватиметься тільки проекція υ_{0y}, а проекція υ_{0x} залишатиметься незмінною. Тому координата *x* тіла з плином часу змінюватиметься так само, як у прямолінійному рівномірному русі:

$x = \upsilon_{0x} \cdot t.$

А координата *y* змінюватиметься так само, як у прямолінійному рівноприскореному русі:

$y = \upsilon_{0y} \cdot t - \frac{g \cdot t^2}{2}.$

Таким чином, одержуємо параметричне подання залежності між координатами *x* та *y* від часу у вигляді:

$x(t) = x_0 + \upsilon_0 \cdot cos\,\alpha \cdot t,$

$y(t) = y_0 + \upsilon_0 \cdot sin\,\alpha \cdot t - \frac{g \cdot t^2}{2}.$

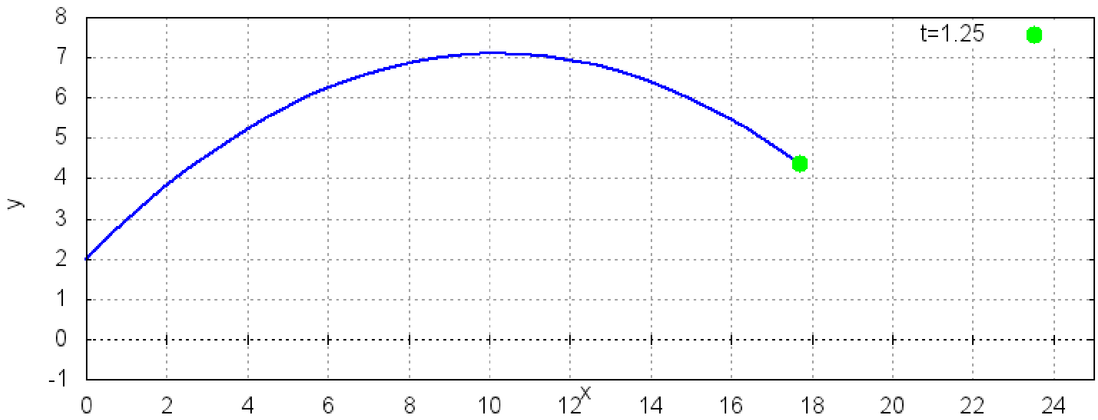
Отже, спочатку визначаємо початкові координати м’яча та початкову швидкість і прискорення земного тяжіння. Координати м’яча в довільний момент часу t задаємо у вигляді параметрично заданої функції $s(t)$. Щоб показати динаміку руху м’яча в часі, введемо параметр анімації *k*, який набуватиме значень часу *t* в межах від 0 до 1,55 секунд з кроком 0,05. Значення параметра *k* подамо у вигляді списку, використовуючи процедуру `makelist()`.

Траєкторію руху м’яча зображуємо як параметрично задану лінію. М’яч зображуємо за допомогою графічної процедури `point()`, аргументом якої є список з координат положення м'яча в певний момент часу, які визначаються за функцією $s(t)$.

```
x0:0$ y0:2$ /· початкові координати м'яча·/
v:20$ g:9.8$ /· початкова швидкість, прискорення·/
/· параметричні рівняння руху м'яча ·/
s(t):=[x0+v·t·cos(π/4), y0+v·t·sin(π/4)−g·t^2]$
/· побудова зображення ·/
wxanimate_draw( k, makelist(i, i, 0, 1.55, 0.05), /· параметр·k/
/· налаштування вигляду координатної площини ·/
  proportional_axes=xy,
  xrange=[0,25], yrange=[0,8],
  xtics_axis=true, xlabel="x", ylabel="y",
  xaxis=true, xtics=[0,2,30], grid=true,

/· траєкторія руху м'яча ·/
  line_width=2,
  parametric(x0+v·t·cos(π/4), y0+v·t·sin(π/4)−g·t^2,t,0,k),
/· точка ·/
  color=green,point_type=filled_circle,
  key=sconcat("t=",k),point_size=2,
  points([s(k)]));
```

В результаті виконання програмного коду одержимо анімаційне зображення, окремий кадр якого має вигляд:



Приклад 2.34. Створити анімаційну демонстрацію утворення циклоїди, яку описує фіксована точка на колі радіуса r, що котиться вздовж осі Ох.

Розглянемо коло радіуса $r = 1$ з центром в точці (0; 1). В результаті руху даного кола вздовж осі *Ox* абсциса центру буде змінюватися на величину параметра анімації, а ордината залишатиметься незмінною.

Нехай *ang* – параметр анімації. Щоб накреслити три арки циклоїди, коло повинно зробити три повних оберти, тому *ang* змінюється від 0 до 6π, причому кожен наступний кадр анімації буде відрізнятися від попереднього на величину $\Delta ang = \dfrac{\pi}{10}$. Зміну величини параметра для кожного окремого кадру відобразимо в легенді, використовуючи опцію `key` та функцію `sconcat()`.

Параметричні рівняння кола матимуть вигляд:

$x(t) = ang + sin\, t,$

$y(t) = 1 + cos\, t.$

Параметричні рівняння циклоїди:

$x(t) = r(t - sin\, t),$

$y(t) = r(1 - cos\, t).$

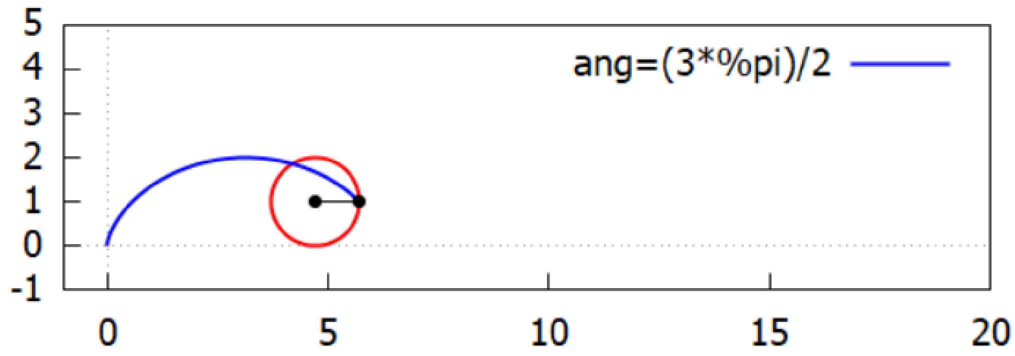
Зміна t від 0 до ang забезпечить ілюстрацію плавної побудови лінії циклоїди як сліду точки на колі з координатами $(r(ang - \sin ang); r(1 - \cos ang))$.

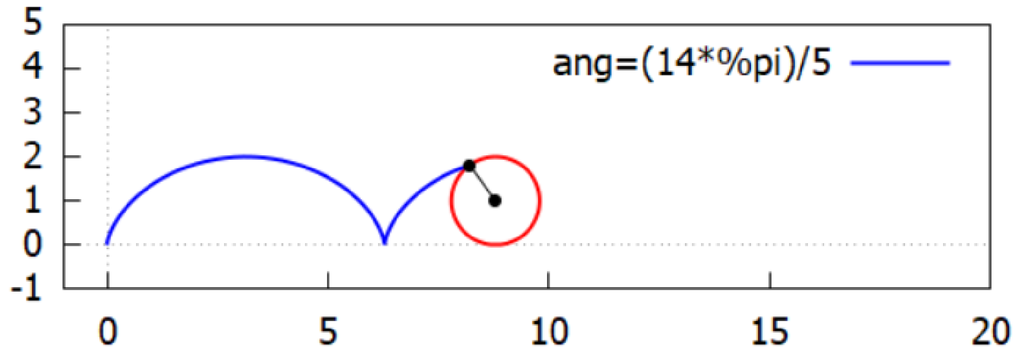
Отже, записуємо процедуру створення анімаційного зображення:

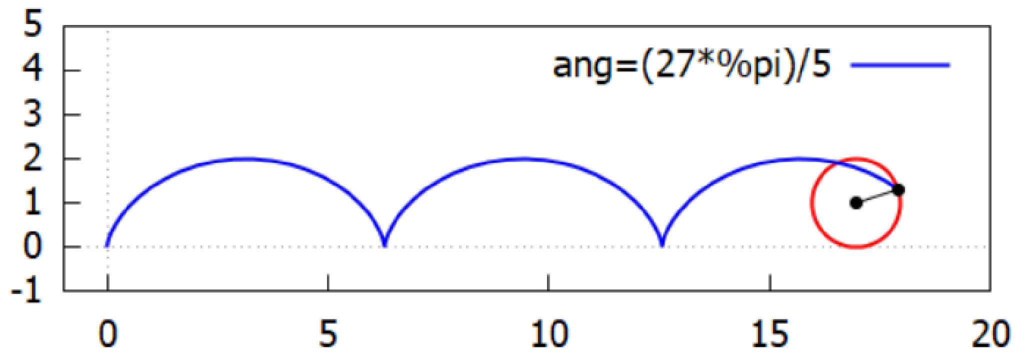
```
with_slider_draw(
ang, makelist(i,i,0,60)·π/10,
proportional_axes=xy, xaxis=true, yaxis=true,
xrange = [−1,20], yrange = [−1,5], nticks = 80,
        /·коло·/
color = red, line_width = 2,
parametric(ang+sin(t), 1+cos(t),t,0,2·π),
       /·циклоїда·/
color = blue, key=sconcat("ang=",ang),
line_width = 2,
parametric(1·(t−sin(t)), 1·(1−cos(t)),t,0,ang),

       /·точки·/
key=false, color = black, point_type=7,
point_size = 1, points_joined = true,
line_width=1,
points([[ang,1], /· центр кола·/
[1·(ang−sin(ang)),  1·(1−cos(ang))]])); /· точка на колі·/
```

В результаті одержуємо анімацію, окремі кадри якої такі:







Приклад 2.35. Створити анімаційну демонстрацію утворення гіпоциклоїди, кривої, що описується точкою кола, яке котиться по внутрішній стороні іншого нерухомого кола без проковзування.

Параметричні рівняння гіпоциклоїди мають вигляд:

$$\begin{cases} x = r(k-1)(cos\, t + \frac{cos(k-1)t}{k-1}), \\ y = r(k-1)(sin\, t - \frac{sin(k-1)t}{k-1}), \end{cases}$$

де $k = \frac{R}{r}$, R – радіус нерухомого кола, r – радіус кола, що котиться.

Розглянемо випадок, коли $R = 8$, $r = 2$.

Нехай ang – параметр анімації. Для того, щоб коло зробило повний оберт, ang змінюється від 0 до 2π, причому різниця між кадрами анімації нехай складає $\Delta ang = \frac{\pi}{10}$.

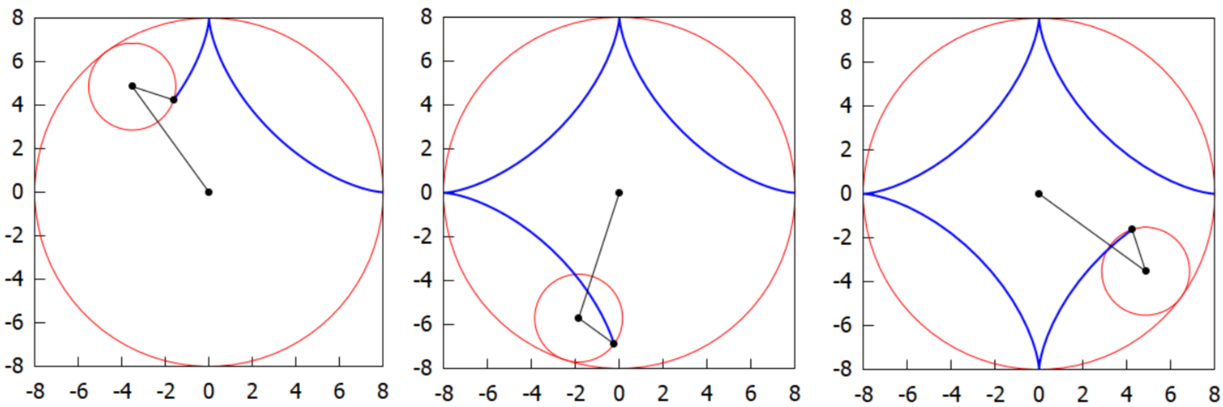
Під час руху кола, його центр рухатиметься по колу з радіусом $R - r$. Отже, параметричні рівняння рухомого кола матимуть вигляд: $(R - r)\cos ang + r \sin ang, (R - r)\sin ang + 2\cos t$.

Зміна t від 0 до ang в параметричних рівняннях гіпоциклоїди забезпечить ілюстрацію плавної побудови лінії (в даному випадку астроїди) як сліду точки на колі з відповідними координатами $((8-2)(cos\, a\, ng + 2\, cos(\, ang((8/2-1)/(8-2));$
$(8-2)(sin\, a\, ng + 2\, sin(\, ang((8/2-1)/(8-2))).$

```
R: 8$ r: 2$ /·радіуси великого і малого кола·/
with_slider_draw(
   ang, makelist(i, i, 0, 20)·π/10, /·параметр анімації·/
   proportional_axes=xy,
   xrange = [−8,8], yrange = [−8,8], nticks = 80,
         /·нерухоме коло·/
   color = red,
   parametric(R·cos(t), R·sin(t), t, 0, 2·π),
         /·рухоме коло·/
   parametric((R−r)·cos(ang)+r·sin(t),
                     (R−r)·sin(ang)+r·cos(t), t,−π,π),
```

```
   /·гіпоциклоїда (астроїда)·/
 color = blue, line_width = 2,
 parametric((R−r)·(cos(t)+r·cos(t·(R/r−1))/(R−r)),
            (R−r)·(sin(t)−r·sin(t·(R/r−1))/(R−r)),
            t, 0, ang),
          /·точки·/
 color = black, point_type=7, point_size = 1,
 points_joined = true, line_width = 1,
 points([ [0,0], /· центр великого кола·/
        (R−r)·[cos(ang), sin(ang)], /· центр малого кола·/
  /· точка на малому колі ·/
       [(R−r)·(cos(ang)+r·cos(ang·(R/r−1))/(R−r)),
        (R−r)·(sin(ang)−r·sin(ang·(R/r−1))/(R−r))]]));
```

В результаті одержимо анімацію, окремі кадри якої такі:

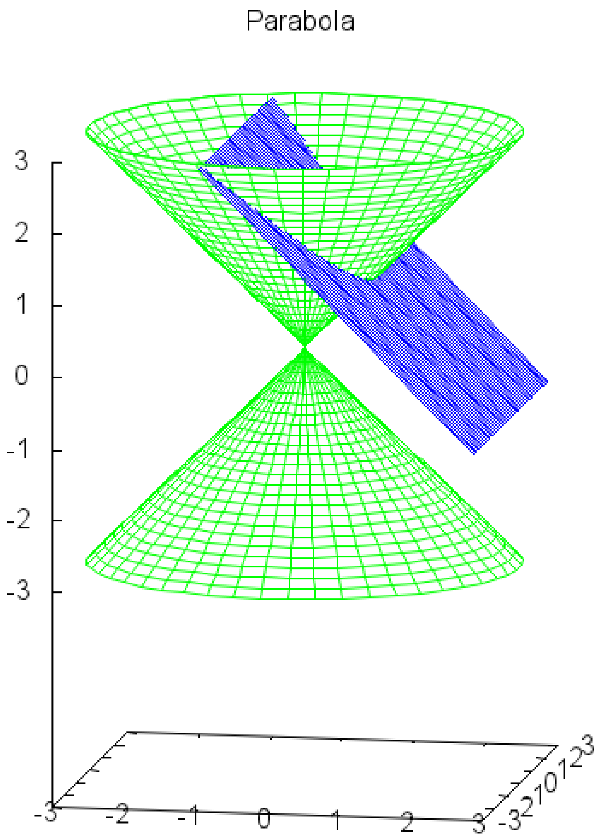


Приклад 2.36. Створити анімаційну демонстрацію перетину конічної поверхні площиною.

Розглянемо конічну поверхню, задану в циліндричних координатах $r(\phi, z) = z$. Нехай $z(x, y) = 1 + k(x - 1)$ – рівняння січної площини, де k – коефіцієнт нахилу площини, змінюючи який, одержуємо в результаті перетину конуса і січної площини коло, еліпс, параболу (коли площина паралельна одній із твірних конуса) та гіперболу.

```
/· підписи до малюнків ·/
s[0]:"Circle"$
s[−1/4]:"Elipse"$
s[−1/2]:"Elipse"$
s[−1]:"Parabola"$
s[−4]:"Hiperbola"$
s[−8]:"Hiperbola"$

with_slider_draw3d(
    k,[0,−1/4,−1/2,−1,−4,−8], /· коефіцієнт нахилу площини·/
    proportional_axes=xyz,  surface_hide=true,
    view=[80,10], xu_grid=50, yv_grid=50,
    zrange=[−3,3], color=blue,
    explicit(1+k·(x−1), x,−3,3, y,−3,3), /· площина·/
    color=green,
    cylindrical(z, z,−3,3, t,0,2·π), /· конус·/
    color=black,
    label([s[k],0,0,4.5])); /· напис·/
```



Задачі для самостійного розв’язування

1. Побудувати графік функції від двох змінних. Зобразити лінії рівня, встановити градієнтне забарвлення поверхні графіка. Показати вигляд поверхні зверху та у фронтальній проєкції.

1) $f(x,y)=(x-2)^2+(y-1)^2-4$;

2) $f(x,y)=2x+4y-2xy$;

3) $f(x,y)=2y^2+(x-1)^2$;

4) $f(x,y)=2xy-3x^2-3y^2+10$;

5) $f(x,y)=xy^2-y^3+3$;

6) $f(x,y)=x^2+x+3y^2-y$;

7) $f(x,y)=x^3+y^3-2xy+1$;

8) $f(x,y)=3x^2+3y^2-2x-2y+2$;

9) $f(x,y)=x^2-y^2+8$;

10) $f(x,y)=\frac{1}{x}+\frac{1}{y}-xy$.

2. Побудувати: а) три паралельні між собою площини; б) три площини, що паралельні координатним площинам, в) три площини, паралельні до координатних осей. Задати різний колір побудованих площин.

3. Побудувати поверхню, задану рівнянням. Налаштувати вигляд поверхні.

1) $3x^2+4y^2+6z^2-6x+16y-36z+49=0$ (еліпсоїд);

2) $\frac{x^2}{4}+\frac{y^2}{9}-\frac{z^2}{16}=1$ (однопорожнинний гіперболоїд);

3) $-\frac{x^2}{4}+\frac{y^2}{9}+\frac{z^2}{9}=-1$ (двопорожнинний гіперболоїд);

4) $\frac{x^2}{4}+\frac{y^2}{9}+z=0$ (еліптичний параболоїд);

5) $\frac{x^2}{4}-\frac{y^2}{9}=z$ (гіперболічний параболоїд);

6) $x^2-y^2-z^2-4=0$;

7) $x^2=y^2+z^2$;

8) $x^2+y^2+z^2+2x-4y+8z-4=0$;

9) $3x^2-y^2+6x-4y-10=0$;

10) $y^2+2x-4y+16=0$.

4. Зобразити переріз однопорожнинного гіперболоїда $\frac{x^2}{4}+\frac{y^2}{9}-\frac{z^2}{16}=1$ площинами: а) *Oxz*; б) *Oxy*; в) $x = 4$.

5. Побудувати тор та тороїдну спіраль, що задані параметрично:

$$\begin{cases} x(t,p) = (R + r\,cos\,p)\,cos\,t\,, \\ y(t,p) = (R + r\,cos\,p)\,sin\,t\,, \\ z(t,p) = r\,sin\,p\,,\ \ t,p \in [0;\ 2\pi] \end{cases} \quad \text{та} \quad \begin{cases} x(t) = (2 - 0{,}5\,cos\,t)\,sin(\,t/4), \\ y(t) = (2 - 0{,}5\,cos\,t)\,cos(\,t/4), \\ z(t) = 0{,}5\,sin\,t\,,\ \ t \in [0;\ 8\pi]. \end{cases}$$

Значення параметрів R та r дібрати самостійно, задати їх значення перед використанням графічних процедур. Дослідити вплив числових коефіцієнтів на вигляд тороїдної спіралі.

6. Побудувати тіла обертання, задані рівняннями в циліндричній системі координат:

1) $r(\theta,z) = z^2$;

2) $r(\theta,z) = z^3$;

3) $r(\theta,z) = \sin z$;

4) $r(\theta,z) = \cos z$;

5) $r(\theta,z) = z - 3$;

6) $r(\theta,z) = z + 5$;

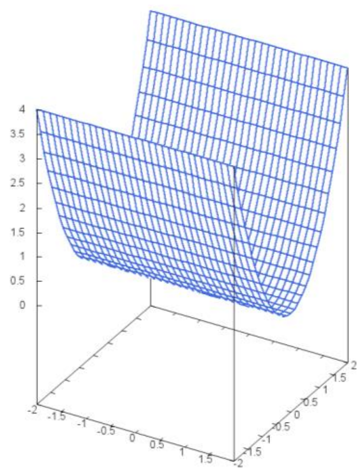
7) $r(\theta,z) = z\sin z$;

8) $r(\theta,z) = z\,cos\,z$;

9) $r(\theta,z) = \frac{1}{z}$;

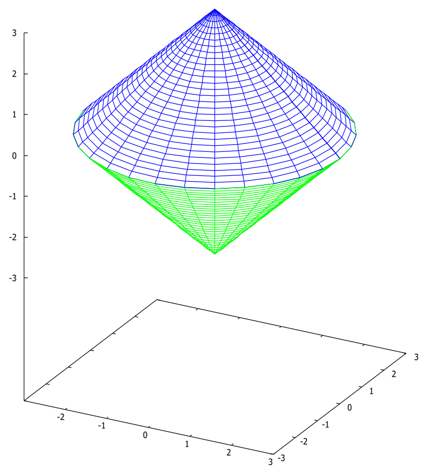
10) $r(\theta,z) = z^e$

7. Знайти рівняння поверхні, графік якої має вигляд як на рисунку. Побудувати цей графік.

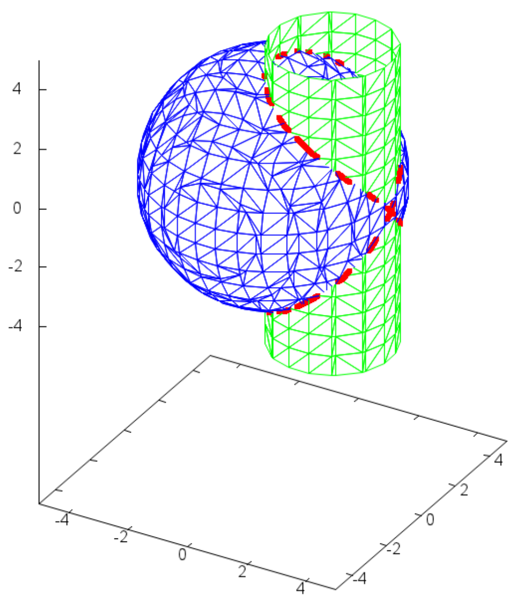


8. Побудувати циліндричну поверхню, в якій поперечними перерізами (перпендикулярними її осі) є спіраль Архімеда.

9. Знайти рівняння поверхні, зображення якої в циліндричній системі координат подано на рисунку:



10. Побудувати криву Вівіані як лінію перетину сфери $x^2 + y^2 + z^2 = 4a^2$ з поверхнею циліндра вдвічі меншого радіуса $(x - a)^2 + y^2 = a^2$, яка проходить через центр сфери. Параметричні рівняння кривої Вівіані знайти у Вікіпедії.



11. Створити анімацію, на якій демонструється графік троянди $r = a\sin t$ в полярних координатах в залежності від параметра a. Дослідити вигляд, якщо a: 1) ціле, 2) раціональне, 3) ірраціональне.

12. Створити анімаційне зображення побудови кривої, що задана параметрично рівняннями: $x = \sin 3t$, $y = \sin 4t$, $t \in [0; 2\pi]$.

13. Створити анімаційну демонстрацію утворення епіциклоїди, кривої, що описується точкою кола, яке котиться по зовнішній стороні іншого нерухомого кола без проковзування. Створити анімаційний файл з розширенням gif.

Контрольні питання

1. Який з графічних об'єктів використовують у функції draw3d() для побудови лінії, що задана параметрично?

a) parametric_surface();

b) parametric();

c) cylindrical();

d) spherical();

2. Який з графічних об'єктів використовують у функції draw3d() для побудови графіка неявно заданої функції?

a) implicit(); b) explicit(); c) cylindrical(); d) spherical();

3. Що означають координати [4; 5; 6] записі vector([1; 1; 1], [4; 5; 6]) ?

a) координати початку вектора;

b) координати кінця вектора;

c) координати вектора;

d) координати середини вектора;

8. В якого графічного формату файл можна зберегти анімаційне зображення в програмі Maxima?

a) .gif; b) mp4; c) avi; d) swf.

4. Встановіть відповідність між опцією функції draw3d() (1-4) та її призначенням (a-e):

1) wired_surface=true	a) приховати невидиму частину поверхні;
2) enhanced3d=true	b) сховати сітку на поверхні;
3) palette=gray	c) градієнтне забарвлення поверхні;
4) surface_hide=true	d) поверхня в градаціях сірого;
	e) показати сітку на поверхні.

5. Яка опція використовується у функції draw3d() для зображення контурних ліній поверхні у проєкції на площину Oxy?

a) contur=none;

b) contur=both;

c) contur=xy;

d) contur=base.

6. Які функції в програмі Maxima призначені для побудови анімаційних зображень?

a) wxanimate_draw();

b) animate_draw();

c) contur=xy;

d) contur=base.

7. Який із записів графіка явно заданої функції в команді wxanimate_draw() задає анімаційну демонстрацію плавної побудови лінії графіка залежно від параметра k?

a) explicit(k*x^2, x, -10, 10) ;

b) explicit(x^2, x, 0, k) ;

c) with_slider_draw3d();

d) with_slider_draw2d().

3 Розвязування задач лінійної алгебри в Maxima

3.1 Матриці та дії з матрицями

В системі Maxima є значна кількість послуг та функцій для виконання дій з матрицями і векторами та розв'язування задач лінійної алгебри.

Розглянемо основні способи подання даних у вигляді матриць, їх перетворень та виконання дій з матрицями. Введення матриці виконується за допомогою послуги головного меню *Алгебра/Ввести матрицю…* або за вбудованою функцією. Функція для введення прямокутної матриці розмірності $m \times n$, тобто матриці з m рядків і n стовпчиків, в системі Maxima має вигляд:

`matrix(`[$a_{11}, a_{12}, ..., a_{1n}$], [$a_{21}, a_{22}, ..., a_{2n}$], . . . [$a_{m1}, a_{m2}, ..., a_{mn}$])

Якщо в матриці кількість рядків m дорівнює кількості стовпчиків n, то її називають квадратною матрицею n-го порядку.

Приклад 3.1. Створити квадратну матрицю *A* третього порядку.

```
(%i1) A: matrix([1.5, 1.3, -2],
                [2.2, -0.5, 2],
                [3.5, -3.1, 3]);
```

(A) $\begin{pmatrix} 1.5 & 1.3 & -2 \\ 2.2 & -0.5 & 2 \\ 3.5 & -3.1 & 3 \end{pmatrix}$

Приклад 3.2. Створити матрицю *B* розмірністю 2×3.

```
(%i2)B: matrix([2, a, a+b],[3, b, a-b]);
```

(B) $\begin{pmatrix} 2 & a & a+b \\ 3 & b & a-b \end{pmatrix}$

Для звернення до елемента матриці після її імені в квадратних дужках записують індекси елемента матриці. Перший індекс позначає номер рядка, другий – номер стовпця. Наприклад, запис `A[1][3]` позначає елемент, що знаходиться на перетині першого рядка і третього стовпчика матриці `A`. Записом `A[n]` звертаються до n-го рядка матриці.

Приклад 3.3. Замінити другий елемент першого рядка матриці *B* в прикладі 2 на число 5.

```
(%i4) B[1][2]: 5$
      B;
```

(%o4) $\begin{pmatrix} 2 & 5 & a+b \\ 3 & b & a-b \end{pmatrix}$

В середовищі програми Maxima матрицю можна створити на основі попередньо заданого вкладеного списку даних. І навпаки, будь-яку матрицю можна конвертувати у список. Це виконується за допомогою функцій `apply(‵matrix, list)` та `args(M)`, де `list` – попередньо заданий список, M – матриця.

Приклад 3.4. Перетворити список [[1, 2], [3, 4]] в матрицю L.

```
(%i5)     list1: [[1,2], [3,4]];
          L: apply('matrix, list1);
```

(list1) $[[1,2],[3,4]]$

(L) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

Приклад 3.5. Перетворити матрицю B в список list2.

```
(%i8)     list2: args(B);
```

(list2) $[[2,5,b+a],[3,\alpha,a-b]]$

Матриці спеціального вигляду в програмі Maxima генеруються за допомогою вбудованих функцій:

`zeromatrix(n, m)` – матриця розмірністю n×m, всі елементи якої нулі.

`ident(n)` – одинична матриця.

`diagmatrix(n, x)` – квадратна матриця n-го порядку, в якій елементи головної діагоналі дорівнюють x, а всі інші елементи – нулі.

`transpose(M)` – транспонування матриці M; якщо в матриці M розміру $n\times m$ поміняти місцями рядки та стовпці, то вийде матриця розміру $m\times n$, яка називається транспонованою до матриці M.

`col(M,i)` – i-тий стовпчик матриці M.

`row(M,i)` – i-тий рядок матриці M.

`addcol(M, list1, …, listn)` – розширення матриці *M* стовпчиками справа на основі даних зі списків list1, …, listn; кількість елементів списку повинна дорівнювати кількості рядків матриці.

`addrow(M, list1, …, listn)` – розширення матриці *M* рядками вниз на основі даних зі списків list1, …, listn; кількість елементів списку повинна дорівнювати кількості стовпців матриці.

`submatrix(i1, …, in, M, j1, …, jm)` – матриця, що утворюється з матриці *M* викреслюванням рядків *i1, …, in* та стовпців *j1, …, jm*.

`triangularize(M)` – зведення матриці *M* до трикутного вигляду; квадратна матриця *M* є трикутною, якщо її елементи, що розташовані вище або нижче головної діагоналі, дорівнюють нулю.

`echelon(M)` – зведення матриці M до трикутного вигляду, в якій перший відмінний від нуля елемент рядка дорівнює одиниці.

Приклад 3.6. Створити діагональну матрицю *M* (3х3), де $a_{11} = a_{22} = a_{33} = 5$.

```
(%i1) A: diagmatrix([3,5]);
```

(M) $\begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$

Приклад 3.7. Створити матрицю *B*, яка утворюється з матриці *A* розширенням стовпчиками [1,3,5], [2,4,6].

```
(%i1) A: matrix([2,3],[4,9],[8,27]);
```

(A) $\begin{pmatrix} 2 & 3 \\ 4 & 9 \\ 8 & 27 \end{pmatrix}$

```
(%i2) B: addcol(A, [1,3,5], [2,4,6]);
```

(B) $\begin{pmatrix} 2 & 3 & 1 & 2 \\ 4 & 9 & 3 & 4 \\ 8 & 27 & 5 & 6 \end{pmatrix}$

Приклад 3.8. Створити матрицю *C*, яка утворюється з матриці *B*, якщо в ній видалити третій рядок і третій та четвертий стовпчики.

```
(%i3) C: submatrix(3,B,3,4);
```

(C) $\begin{pmatrix} 2 & 3 \\ 4 & 9 \end{pmatrix}$

Приклад 3.9. Перетворити матрицю A у трикутну матрицю.

```
(%i1) A: matrix([1,2,3], [4,5,6], [7,8,8]);
```

(A) $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{pmatrix}$

```
(%i2) triangularize(A);
```

(%o2) $\begin{pmatrix} 1 & 2 & 3 \\ 0 & 3 & 6 \\ 0 & 0 & 3 \end{pmatrix}$

```
(%i3) echelon(A);
```

(%o3) $\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$

Над матрицями виконують дії додавання і віднімання, множення матриці на число, множення двох матриць, піднесення квадратної матриці до степеня.

В системі Maxima виконуються також поелементні дії над матрицями, які не розглядаються в математиці. До поелементних операцій над матрицями відносяться такі, як поелементне множення і ділення матриць однакових розмірів, а також поелементне піднесення до степеня. Крім того, поелементно обчислюються математичні функції від матриць, внаслідок чого утворюється матриця тих самих розмірів, елементи якої дорівнюють значенням функції від відповідних елементів початкової матриці. В Таблиці 1 подано оператори, за допомогою яких виконують дії над матрицями.

Дії над матрицями X та Y

`X+Y`	додавання двох матриць
`Y−X`	віднімання двох матриць
`X*Y`	поелементне множення двох матриць
`X/Y`	поелементне ділення двох матриць
`k*X`	множення матриці X на число k
`X.Y`	добуток матриць X та Y за правилом множення рядка на стовпчик
`X^n`	поелементне піднесення до степеня *n* елементів матриці
`X^^n`	множення матриці самої на себе *n* разів

`X^^(-1)`	Обчислення оберненої матриці
`exp(X)`	створення матриці, елементами якої є число e в степені, що дорівнює відповідному елементу матриці X.
`sqrt(X)`	створення матриці, елементами якої є квадратні корені елементів матриці X.

Приклад 3.10. Для матриць *A*, *B* і *C* знайти $4 \cdot A$, A^{-1}, B^{T}, C^{2}, $A \cdot B$, $B \cdot A$.

```
(%i1) A: matrix([1,-1,-6], [2, -3, 4]);
      B: matrix([7,4], [8,1], [3,2]);
      C: matrix( [1,0,1], [3,-1,2], [0,2,-3]);
```

(A) $\begin{pmatrix} 1 & -1 & -6 \\ 2 & -3 & 4 \end{pmatrix}$

(B) $\begin{pmatrix} 7 & 4 \\ 8 & 1 \\ 3 & 2 \end{pmatrix}$

(C) $\begin{pmatrix} 1 & 0 & 1 \\ 3 & -1 & 2 \\ 0 & 2 & -3 \end{pmatrix}$

```
(%i4) 4·A;
```

(%o4) $\begin{pmatrix} 4 & -4 & -24 \\ 8 & -12 & 16 \end{pmatrix}$

```
(%i5) A^(-1);
```

(%o5) $\begin{pmatrix} 1 & -1 & -\frac{1}{6} \\ \frac{1}{2} & -\frac{1}{3} & \frac{1}{4} \end{pmatrix}$

```
(%i6) transpose(B);
```

(%o6) $\begin{pmatrix} 7 & 8 & 3 \\ 4 & 1 & 2 \end{pmatrix}$

```
(%i7) C^2;  /*поелементне піднесення до квадрату елементів матриці*/
```

(%o7) $\begin{pmatrix} 1 & 0 & 1 \\ 9 & 1 & 4 \\ 0 & 4 & 9 \end{pmatrix}$

```
(%i8) C^^2;  /*множення матриці на саму себе*/
```

(%o8) $\begin{pmatrix} 1 & 2 & -2 \\ 0 & 5 & -5 \\ 6 & -8 & 13 \end{pmatrix}$

```
(%i9) A.B;  /*множення матриці на матрицю*/
```

(%o9) $\begin{pmatrix} -19 & -9 \\ 2 & 13 \end{pmatrix}$

```
(%i10) B.A;
```

(%o10) $\begin{pmatrix} 15 & -19 & -26 \\ 10 & -11 & -44 \\ 7 & -9 & -10 \end{pmatrix}$

```
(%i11) is(A.B = B.A);
(%o11) false
```

3.2 Визначник матриці та ранг матриці

Визначником (детермінантом) квадратної матриці

$$D = \Delta = det\,A = det[\,a_{i,j}] = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

називається число, яке дорівнює сумі добутків елементів довільного рядка (або стовпця) на їх алгебраїчне доповнення: $D = \sum_{i=1}^{n} a_{ij}A_{ij} = \sum_{k=1}^{n} a_{jk}A_{jk}$.

Алгебраїчним доповненням елемента a_{ij} визначника Δ називається мінор M_{ij} цього елемента, узятий із знаком $(-1)^{i+j}$.

Мінором M_{ij} елемента a_{ij} визначника Δ називається новий визначник, який виходить з даного викреслюванням рядка і стовпця, що містить елемент a_{ij}.

Для обчислення визначника матриці та мінорів в системі Maxima використовуються вбудовані функції:

`determinant(A)` – обчислення визначника матриці *A*.

`minor(A, i, j)` – матриця, яка утворюється з матриці *A* видаленням *i*-го рядка та *j*-го стовпчика.

`determinant(minor(A, i, j))` – обчислення визначника матриці, яка утворюється з матриці А видаленням і-го рядка та j-го стовпчика.

Властивості визначника:

1. В результаті транспонування матриці значення її визначника не змінюється.
2. В результаті перестановки двох рядків (або стовпців) матриці знак її визначника змінюється на протилежний.

3. Якщо всі елементи деякого рядка (або стовпця) матриці мають спільний множник, то його можна винести за знак визначника матриці.
4. Визначник матриці, що має два однакових рядки або стовпці дорівнює нулю.
5. Якщо всі елементи деякого рядка або стовпця матриці визначника дорівнюють нулю, то визначник матриці дорівнює нулю.
6. Якщо до будь-якого рядка (стовпця) визначника матриці додати відповідні елементи іншого рядка (стовпця), помножені на одне і те саме число, то значення визначника не зміниться.
7. Трикутний визначник, у якого всі елементи, що лежать вище (нижче) головної діагоналі, є нулями, дорівнює добуткові елементів головної діагоналі.

Приклад 3.11. Обчислити визначник матриці A, мінор M_{12} та алгебраїчне доповнення A_{12}.

```
(%i1) A: matrix([1,2,3],[1,4,7],[2,6,13]);
```

(A) $$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 4 & 7 \\ 2 & 6 & 13 \end{pmatrix}$$

```
(%i2) determinant(A);
(%o2) 6
(%i3) M12: determinant(minor(A,1,2));
(M12) -1
(%i4) A12: (-1)^(1+2)M12
(A12) 1
```

Квадратна матриця A називається невиродженою, якщо визначник $\Delta = det\, A \neq 0$.

Якщо A – квадратна матриця, то оберненою до A називається матриця, в результаті множення якої на A^{-1} (як зліва, так і справа) отримується одинична матриця: $A^{-1} \cdot A = A \cdot A^{-1} = E$.

Якщо обернена матриця A^{-1} існує, то матриця A називається оборотною. Для того, щоб квадратна матриця мала обернену, необхідно і достатньо, щоб матриця A була невиродженою.

Обчислення оберненої матриці в явному вигляді (обчислення визначника і відповідних алгебраїчних доповнень) досить громіздка операція. Для обчислення оберненої матриці в системі Maxima використовується вбудована в функція `invert(A)` або застосовується оператор піднесення до степеня матриці `A^^(-1)`.

Рангом матриці A називається найбільший з порядків відмінних від нуля мінорів. Якщо ранг матриці A дорівнює r, то це означає, що матриця A має хоча б один відмінний від нуля мінор порядку r, але будь-який мінор порядку більшого, ніж r, дорівнює нулю. Властивості рангу матриці:

1. Ранг матриці не змінюється після транспонування.
2. Ранг матриці не змінюється після перестановки її рядків (або стовпців).
3. Ранг матриці не змінюється після множення рядка (або стовпця) на відмінне від нуля число.
4. Ранг матриці не змінюється після додавання між собою рядків або стовпців (без зміни їх кількості).

Дві матриці називаються еквівалентними, якщо ранг однієї матриці дорівнює рангу іншої ($A \sim B$).

Приклад 3.12. Перевірити матриці A і B на рівність та еквівалентність.

Для перевірки рівності використаємо функцію `is`, результатом виконання якої є `true`, якщо рівність правильна, або `false`, якщо рівність неправильна.

```
(%i1) A: matrix([1,2,3],[1,4,7],[2,6,13]);
      B: matrix([1,4,2],[2,5,4],[3,1,7]);
```

(B) $$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 4 & 7 \\ 2 & 6 & 13 \end{pmatrix}$$

(B) $$\begin{pmatrix} 1 & 4 & 2 \\ 2 & 5 & 4 \\ 3 & 1 & 7 \end{pmatrix}$$

```
(%i2) is(A=B);
(%o2) false
(%i3) is(rank(A)=rank(B));
(%o3) true
```

Отже, матриці A і B не є рівними, але мають однаковий ранг матриці.

```
(%i5) rank(A); rank(B);
(%o4) 3
(%o5) 3
```

3.3 Розв'язування систем лінійних алгебраїчних рівнянь

Під час розв'язування прикладних задач досить часто доводиться розв'язувати системи лінійних рівнянь. Системою лінійних рівнянь, що містить m рівнянь і n невідомих, називається система рівнянь вигляду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_2 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots, \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} \tag{3.1}$$

де a_{ij}, $i = \overline{1,m}$, $j = \overline{1,n}$ коефіцієнти системи, b_i – вільні члени, x_j – невідомі. Розв'язком даної системи називається сукупність чисел x_1, x_2,..., x_n, в результаті підстановки яких у всі рівняння системи вони перетворюються в тотожності.

Система лінійних рівнянь в матричній формі має вигляд $A \cdot X = B$, де A – матриця коефіцієнтів системи, X – вектор-стовпець невідомих і B – вектор-стовпець вільних членів:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Розширеною матрицею системи називається матриця $\overline{A}$, яка утворюється після доповнення матриці коефіцієнтів стовпцем вільних членів:

$$\overline{A} = \begin{pmatrix} a_{11} & a_{12} & \mathrm{K} & a_{1n} & b_1 \\ a_{21} & a_{22} & \mathrm{K} & a_{2n} & b_2 \\ \mathrm{K} & \mathrm{K} & \mathrm{K} & \mathrm{K} & \mathrm{K} \\ a_{m1} & a_{m2} & \mathrm{K} & a_{mn} & b_m \end{pmatrix}$$

В програмі Maxima є вбудована функція `augcoefmatrix([eq1, …, eqm],[x1, ..., xn])`, якою створюється розширена матриця коефіцієнтів системи лінійних рівнянь `eq1, …, eqm` з n

невідомими `x1,...,xn`, але слід зауважити, що вектор-стовпчик вільних членів в даній функції формується з протилежним знаком.

Система лінійних рівнянь називається сумісною, якщо вона має хоча б один розв'язок, і несумісною, якщо вона не має жодного розв'язку. Сумісна система може бути визначеною (має один розв'язок) або невизначеною (має безліч розв'язків).

Для того, щоб система лінійних рівнянь була сумісною, необхідно і достатньо, щоб ранг матриці A дорівнював рангу розширеної матриці $\overline{A}$ (теорема Кронекера-Капеллі).

Якщо ранг матриці системи $r(A)$ дорівнює числу невідомих і $\Delta = det\, A \neq 0$, то система має один розв'язок.

Якщо ранг матриці системи менший числа невідомих і система сумісна, то вона має безліч розв'язків.

Приклад 3.13. Дослідити на сумісність систему лінійних рівнянь:

$$\begin{cases} 2x_1 - x_2 + 3x_3 = 9, \\ 3x_1 - 5x_2 + x_3 = -4, \\ 4x_1 - 7x_2 + x_3 = 5. \end{cases}$$

Для створення матриці коефіцієнтів даної системи лінійних рівнянь скористаємося функцією `matrix()`.

```
(%i1) A: matrix([2,-1,3],[3,-5,1],[4,-7,1]);
```

(A) $\begin{pmatrix} 2 & 1 & 3 \\ 3 & 5 & 1 \\ 4 & -7 & 1 \end{pmatrix}$

```
(%i2) AA: addcol(A,[9,-4,5]); /*розширена матриця*/
```

(AA) $\begin{pmatrix} 2 & 1 & 3 & 9 \\ 3 & 5 & 1 & 4 \\ 4 & 7 & 1 & 5 \end{pmatrix}$

```
(%i4) rank(A); rank(AA);
(%o3) 2
(%o4) 3
```

Оскільки ранг розширеної матриці не дорівнює рангу матриці коефіцієнтів системи лінійних рівнянь, то дана система несумісна, тобто не має розв'язку.

Розв'язок системи лінійних рівнянь можна знайти різними способами. Розглянемо матричний метод та метод Крамера.

Якщо основна матриця системи лінійних рівнянь (1) є квадратною і невиродженою, то систему можна розв'язати матричним способом за допомогою оберненої матриці. Записавши систему в матричному вигляді, дістанемо розв'язок X системи рівнянь у вигляді:

$AX = B \;\Leftrightarrow\; A^{-1}(AX) = A^{-1}B \;\Leftrightarrow\; (A^{-1}A)X = A^{-1}B \;\Leftrightarrow\; X = A^{-1}B,$ де A^{-1} – обернена квадратна матриця.

Отже, для розв'язування системи лінійних рівнянь матричним методом необхідно виконати наступні кроки:

1. Переконатися в тому, що матриця A є невиродженою (визначник її не дорівнює нулю), тобто існує обернена матриця.
2. Знайти обернену матрицю A^{-1}.
3. Обчислити добуток оберненої матриці на вектор-стовпець вільних членів $A^{-1}B$.
4. Отриманий вектор-стовпець буде розв'язком X.

Метод Крамера розв'язування системи лінійних рівнянь зводиться до обчислення визначників.

Якщо у визначнику основної матриці коефіцієнтів системи (1) замінити по черзі стовпці коефіцієнтів при x_1, x_2,..., x_n, на стовпець вільних членів, отримаємо n визначників:

$$\Delta_1 = \begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ b_n & a_{n2} & \dots & a_{nn} \end{vmatrix}, \Delta_2 = \begin{vmatrix} a_{11} & b_1 & \dots & a_{1n} \\ a_{21} & b_2 & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & b_n & \dots & a_{nn} \end{vmatrix}, \Delta_n = \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_n \end{vmatrix}$$

Формули Крамера для розв'язування системи рівнянь мають вигляд:

$x_1 = \frac{\Delta_1}{\Delta}, x_2 = \frac{\Delta_2}{\Delta}, \dots , x_n = \frac{\Delta_n}{\Delta}$ або $x_i = \frac{\Delta_i}{\Delta}$, де $i = 1,2, \dots , n$.

Якщо $\Delta = 0$, а хоча б один з Δ_1, Δ_1, …, Δ_1 відмінний від нуля, то система (3.1) розв'язків немає. Якщо ж $\Delta_1 = \Delta_2 = \dots = \Delta_n = 0$, то система має безліч розв'язків.

Приклад 3.14. Розв'язати систему лінійних рівнянь матричним способом та за методом Крамера:

$$\begin{cases} 7x_1 - 2x_2 + 3x_3 = 15, \\ 5x_1 - 3x_2 + 2x_3 = 15, \\ 10x_1 - 11x_2 + 5x_3 = 36. \end{cases}$$

Для розв'язування даної системи в програмі Maxima записуємо:

```
(%i1) A: matrix([7,-2,3],[5,-3,2],[10,-11,5]);
```

(A) $\begin{pmatrix} 7 & -2 & 3 \\ 5 & -3 & 2 \\ 10 & -11 & 5 \end{pmatrix}$

```
(%i2) B: matrix([15],[15],[36]);
```

(B) $\begin{pmatrix} 15 \\ 15 \\ 36 \end{pmatrix}$

```
(%i3)D: determinant(A);
(%o3) -16
/*Δ не дорівнює нулю, отже обернена матриця існує*/
(%i4) X: invert(A).B;
```

(X) $\begin{pmatrix} \frac{15}{4} \\ -\frac{9}{4} \\ -\frac{21}{4} \end{pmatrix}$

```
(%i5) X, numer;
```

(%o5) $\begin{pmatrix} 3.75 \\ -2.25 \\ -5.25 \end{pmatrix}$

```
/* розв'язок системи:
   X1=3,75
   X2=-2,25
   X3=-5,25*/
```

Виведення розв'язку системи лінійних рівнянь в системі Maxima виконується у дробовому форматі подання чисел. За необхідності дробові числа можна подати у вигляді десяткових дробів за допомогою параметра `numer`.

Далі розв'яжемо дану систему рівнянь методом Крамера.

/*виконуємо заміну стовпчиків матриці A на вектор-стовпчик вільних членів та обчислюємо додаткові визначники*/

```
(%i6) A1: matrix([15,-2,3],[15,-3,2],[36,-11,5])$
      A2: matrix([7,15,3],[5,15,2],[10,36,5])$
      A3: matrix([7,-2,15],[5,-3,15],[10,-11,36])$
      D1: determinant(A1);
      D2: determinant(A2);
      D3: determinant(A3);
(D1)  -60
(D2)  36
(D3)  84
```

/* знаходимо розв'язок у вигляді x1=D1/D, x2=D2/D, x3=D3/D*/

```
(%i7)x1: D1/D; x2: D2/D; x3: D3/D;
```

(x1) $\frac{15}{4}$

(x2) $-\frac{9}{4}$

(x3) $-\frac{21}{4}$

Одним із способів розв’язування систем лінійних рівнянь є метод Гауса, що полягає в послідовному виключенні невідомих. Систему з n рівнянь спочатку приводять до рівносильної системи з трикутною матрицею коефіцієнтів (прямий хід). Потім виконують обернений хід, обчислюючи значення x_i і послідовно підставляючи їх в рівняння для отримання значень x_{i-1}. Розв’яжемо систему з прикладу 14 методом Гауса.

`(%i8) R: addcol(A,B);` /*створюємо розширену матрицю*/

(R) $\begin{pmatrix} 7 & 2 & 3 & 15 \\ 5 & -3 & 2 & 15 \\ 10 & -11 & 5 & 36 \end{pmatrix}$

Використовуючи функцію `echelon()`, розширену матрицю коефіцієнтів легко звести до трикутного вигляду:

```
(%i9) echelon(R);
```

(%o9) $\begin{pmatrix} 1 & -\frac{2}{7} & \frac{3}{7} & \frac{15}{7} \\ 0 & 1 & \frac{1}{11} & -\frac{30}{11} \\ 0 & 0 & 1 & -\frac{21}{4} \end{pmatrix}$

```
(%i10) x3: -21/4;
       x2: -30/11-(1/11)*x3;
       x1: 15/7-(3/7)*x3+(2/7)*x2;
```

(x1) $\frac{15}{4}$

(x2) $-\frac{9}{4}$

(x3) $-\frac{21}{4}$

Для розв’язування системи лінійних рівнянь за допомогою програми Maxima можна також скористатися вбудованою функцією `solve([eq1,…,eqm], [x1,...,xn])`, яка призначена для розв’язування рівнянь і систем рівнянь і за допомогою якої автоматично обирається метод для розв’язування заданої системи рівнянь. В цьому випадку розв’язування системи лінійних рівнянь будь-якого порядку досягається за однією командою.

```
(%i1) solve([7*x1-2*x2+3*x3= 15,
             5*x1-3*x2+ 2*x3= 15,
             10*x1-11*x2+ 5*x3= 36], [x1,x2,x3]);
```

(%o1) $[[\frac{15}{4},-\frac{9}{4},-\frac{21}{4}]]$

Розглянуті методи розв’язування систем лінійних рівнянь відносяться до точних, тобто в результаті їх застосування отримується точний розв’язок задачі після скінченного числа арифметичних і логічних операцій за відсутності округлень.

Наближеними (ітераційними) методами розв’язування системи лінійних рівнянь є методи Якобі, метод Гауса-Зейделя тощо, які теж можна виконати з використанням засобів системи Maxima.

Задачі для самостійного розв'язування

1. Використовуючи засоби програми Maxima, перевірити властивості визначника і рангу матриці.

2. Знайти транспоновану та обернену матрицю.

1) $A=\begin{pmatrix} 2 & 1 & 1 \\ 5 & 1 & 3 \\ 2 & 1 & 2 \end{pmatrix}$ 2) $A=\begin{pmatrix} 7 & 2 & 3 \\ 5 & -3 & 2 \\ 10 & -4 & 5 \end{pmatrix}$

3) $A=\begin{pmatrix} 2 & -2 & -3 \\ 1 & 1 & 2 \\ 2 & -1 & -1 \end{pmatrix}$ 4) $A=\begin{pmatrix} 1 & -2 & 1 \\ 1 & 3 & 2 \\ 5 & -2 & 4 \end{pmatrix}$

5) $A=\begin{pmatrix} 1 & 1 & 3 \\ 3 & 2 & 2 \\ 1 & -1 & 5 \end{pmatrix}$ 6) $A=\begin{pmatrix} 4 & -1 & 3 \\ 3 & 2 & 5 \\ 2 & 2 & 4 \end{pmatrix}$

7) $A=\begin{pmatrix} 2 & 1 & -1 \\ 1 & -1 & -3 \\ 3 & 2 & 4 \end{pmatrix}$ 8) $A=\begin{pmatrix} 1 & 1 & -2 \\ 2 & -3 & 1 \\ 3 & -2 & 6 \end{pmatrix}$

9) $A=\begin{pmatrix} 2 & 2 & 1 \\ 3 & 1 & 2 \\ 4 & -1 & -1 \end{pmatrix}$ 10) $A=\begin{pmatrix} 5 & -3 & 6 \\ 2 & -1 & -3 \\ 1 & 4 & -2 \end{pmatrix}$

3. Дослідити систему лінійних рівнянь на сумісність і у випадку її сумісності знайти розв'язок різними способами.

1.	$\begin{cases} x_1 - 3x_2 + x_3 = 4, \\ 2x_1 - x_2 - x_3 = 2, \\ x_1 + 2x_2 + x_3 = -1, \\ 4x_1 - 2x_2 + x_3 = 5 \end{cases}$	6.	$\begin{cases} 3x + y = 5, \\ y + 2z = -1, \\ -x + 3z = 21 \end{cases}$
2.	$\begin{cases} 2x + 3y + 5z = 8, \\ x + 4y + 5z = 2, \\ x + y - 10z = 2 \end{cases}$	7.	$\begin{cases} 3x_1 + 2x_2 + x_3 = 200, \\ x_1 + 6x_2 + 2x_3 = 260, \\ 4x_1 + x_2 + 5x_3 = 290 \end{cases}$
3.	$\begin{cases} 3x + y - 2z = 1, \\ x + 2y + z = -3, \\ 4x + y + 3z = 8 \end{cases}$	8.	$\begin{cases} x_1 + 3x_2 - 4x_3 = 5, \\ 2x_1 + x_2 + 3x_3 = -1, \\ 3x_1 - 2x_2 + x_3 = 2 \end{cases}$
4.	$\begin{cases} x + 3y + 4z = 1, \\ x + y + 14z = 3, \\ 2x + 5y + z = 3 \end{cases}$	9.	$\begin{cases} x_1 + 5x_2 - x_3 = 3, \\ 2x_1 + 4x_2 - 3x_3 = 2, \\ 3x_1 - x_2 - 3x_3 = -7 \end{cases}$

5.	$\begin{cases} x+y+2z=4, \\ 2x+y-3z=1, \\ x-y+z=-3 \end{cases}$	10.	$\begin{cases} 2x+y-z=2, \\ x+3y+2z=6, \\ 7x-9y+16z=-18 \end{cases}$

4. Підприємство випускає продукцію двох видів, використовуючи сировину трьох типів. Витрати сировини на виробництво продукції задаються матрицею $S = s_{i,j} = \begin{pmatrix} 7 & 6 \\ 9 & 3 \\ 5 & 2 \end{pmatrix}$, де $s_{i,j}$ – кількість одиниць сировини i-го типу, що використовується на виготовлення продукції j-го типу. План щоденного випуску продукції передбачає 80 одиниць продукції першого виду і 125 одиниць продукції другого виду. Вартість одиниці кожного типу сировини відповідно дорівнює 7, 8 і 11 грошових одиниць. Визначити загальні витрати сировини, необхідні для щоденного випуску продукції, а також загальну вартість цієї сировини.

5. Закодувати і розкодувати повідомлення, використовуючи обернену матрицю.

Контрольні питання

1. Як звернутися до елемента матриці М, який знаходиться на перетині першого рядка і другого стовпчика?

a) M[1,2]; b) M[1][2]; c) M(1,2); d) M([1],[2]).

2. Запис В[3] для квадратної матриці В означає:

a) звернення до 3-го стовпчика матриці В;

b) звернення до 3-го рядка матриці В;

c) звернення до 3-го елемента матриці В;

3. Яка з наведених функцій призначена для зведення матриці М до трикутного вигляду:

a) triangularize(M);

b) transpose(M);

c) diagmatrix(M);

d) submatrix(M);

4. Яка з функцій призначена для створення матриці розміром *n*x*n*, в якій по діагоналі всі елементи – одиниці, а всі інші – нулі?

a) ident(n);

b) echelon(n);

c) diagmatrix(n);

d) zeromatrix(n);

5. Як записати добуток матриць A і B за правилом множення рядка на стовпчик?

a) A.B; b) A*B; c) (AB); d) product(A,B).

6. Як знайти обернену матрицю до матриці A?

a) 1/A; b) A^1; c) A^^(-1); d) A^(-1).

7. Щоб перевірити матриці A і B на рівність, потрібно виконати команду:

a) is(A=B);

b) if(A=B);

c) is(rank(A)=rank(B));

d) is(A~B);

8. Щоб перевірити матриці A і B на еквівалентність, потрібно виконати команду:

a) is(A=B);

b) if(A=B);

c) is(rank(A)=rank(B));

d) is(A~B);

9. Вкажіть способи як з даної матриці M створити нову матрицю шляхом вилучення 2-го рядка і 3-го стовпчика.

a) minor(A, 2, 3);

b) submatrix(A, 2, 3);

c) submatrix(2, 3, A);

d) submatrix(2, A, 3);

10. Яку функцію можливо використати для створення розширеної матриці?

a) addcol(); b) addrow(); c) echelon(); d) submatrix();

4 Розв'язування задач математичного аналізу в Maxima

4.1 Основні функції програми Maxima для розв'язування задач математичного аналізу

В програмі Maxima є достатньо багато засобів для розв'язування задач математичного аналізу, зокрема обчислення границь та дослідження функцій на неперервність, обчислення похідних, пошук екстремумів, дослідження властивостей функції за допомогою похідної, інтегрування, розв'язування диференціальних рівнянь та багато інших.

Засоби розв'язування задач математичного аналізу знаходяться в розділі головного меню *Аналіз*. Також в програмі Maxima використовуються функції:

`limit(f(x), x, a, n)` – знаходження границі $\lim\limits_{x \to a} f(x)$, де `f(x)` – вираз функції, `x` – змінна, `a` – значення змінної, для якого знаходимо границю (у випадку нескінченності `inf` або `minf`), `n` – необов'язковий параметр, який використовується у випадку обчислення односторонніх границь і набуває значення `plus` (права границя) `minus` (ліва границя).

`diff(f(x), x, n)` – похідна функції `f(x)` за змінною `x`, `n` – порядок похідної (якщо похідна першого порядку, цей параметр не є обов'язковим)

`diff(f(x,y), x, n, y, m)` – похідна функції `f(x,y)` від двох змінних, де `n` – порядок диференціювання по `x`, `m` – порядок диференціювання по `y`

`integrate(f(x), x)` – невизначений інтеграл функції `f(x)`

`integrate(f(x), x, a, b)` – визначений інтеграл функції `f(x)`, де `a`, `b` – межі інтегрування

`sum(expr,i,n,k)` – сума членів ряду, якщо `i` дорівнює від `n` до `k`

`product(expr,i,n,k)` – добуток членів ряду, якщо `i` дорівнює від `n` до `k` `taylor(f(x),x,a,n)` – розклад функції в ряд Тейлора, де `n` – показник основи *(x–a)* степеня

Розглянемо використання цих функцій на прикладах.

Приклад 4.1. Обчислити границю функції $f(x)=\frac{x^3-3x-2}{2x^3-x-3}$, якщо $x\to\infty$.

```
(%i2) f(x):=(x^3-3·x-2)/(2·x^3-x-3);
      limit(f(x), x, inf);
```

(%o1) $f(x):=\frac{x^3-3x-2}{2x^3-x-3}$

(%o2) $\frac{1}{2}$

Приклад 4.2. Обчислити границю функції $f(x)=\frac{a^x-1}{x}$, якщо $x\to 0$, $a\in R$.

```
(%i2) f(x):=(a^x-1)/x;
      limit(f(x), x, 0);
```

(%o1) $f(x):=\frac{a^x-1}{x}$

(%o2) $\log(a)$

Приклад 4.3. Знайти похідну функції $f(x)=4x\cos^2\frac{x}{2}$, обчислити значення похідної функції в точці $x_0=\frac{\pi}{2}$.

```
(%i4) f(x):=4·x·(cos(x/2))^2; /·визначаємо функцію f(x)·/
      df: diff(f(x), x); /·знаходимо похідну f(x), вираз зберігаємо в df·/
      define(df(x), df); /·визначаємо похідну як функцію·/
      df(π/2); /·знаходимо значення похідної в точці π/2 ·/
```

(%o1) $f(x):=4x\cos\left(\frac{x}{2}\right)^2$

(df) $4\cos\left(\frac{x}{2}\right)^2-4\cos\left(\frac{x}{2}\right)\sin\left(\frac{x}{2}\right)x$

(%o3) $df(x):=4\cos\left(\frac{x}{2}\right)^2-4\cos\left(\frac{x}{2}\right)\sin\left(\frac{x}{2}\right)x$

(%o4) $2-\pi$

Приклад 4.5. Обчислити інтеграл функції $\int ax^2(x+1)^2dx$.

```
(%i1) integrate(a·(x^2)·(x+1)^2, x);
```

(%o1) $\frac{a\,(6x^5+15x^4+10x^3)}{30}$

Приклад 4.6. Знайти визначений інтеграл $\int_0^2 \frac{1}{e^x+1}dx$, вивести його наближене числельне значення.

```
(%i3) I: integrate(1/%e^x+1, x, 0,2);
      I, numer; /·чисельне значення·/
(I)   %e^-2 (2 %e^2 - 1) + 1
(%o3) 2.864664716763388
```

Приклад 4.7. Обчислити суму ряду $\sum_{a=1}^{8} a^2$.

```
(%i1) sum(a^2, a, 1, 8);
(%o1) 204
```

Приклад 4.8. Розкласти функцію $y = \sin x + \cos x$ в ряд Тейлора 5-го степеня в точці x=0.

```
(%i1) taylor(sin(x)+cos(x),x,0,5);
```

(%o1)/T/ $1 + x - \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \ldots$

4.2 Дослідження функції на неперервність

Функцію $f(x)$ називають *неперервною* в точці $x_0 \in D(f)$, якщо $\lim\limits_{x \to x_0} f(x) = f(x_0)$. Якщо функція $f(x)$ – неперервна в точці x_0, то $f(x) \approx f(x_0)$ при $x \approx x_0$.

Точку x_0 називають *точкою розриву* функції $f(x)$, якщо ця функція визначена в точках, як завгодно близьких до точки x_0, і вона не є неперервною в цій точці.

Точку x_0 називають *точкою розриву I роду* якщо існують скінченні односторонні границі в цій точці. При цьому якщо $\lim\limits_{x \to x_0 - 0} f(x) = \lim\limits_{x \to x_0 + 0} f(x)$, то x_0 – *точка усувного розриву* I роду (розрив можна усунути, якщо покласти $f(x_0) = \lim\limits_{x \to x_0} f(x)$). Якщо ж $\lim\limits_{x \to x_0 - 0} f(x) \neq \lim\limits_{x \to x_0 + 0} f(x)$, то x_0 – точка *неусувного розриву* I роду, а число $\delta = \left| \lim\limits_{x \to x_0 + 0} f(x) - \lim\limits_{x \to x_0 - 0} f(x) \right|$ називають *стрибком* функції в точці x_0.

Точку x_0 називають *точкою розриву II роду* якщо в цій точці принаймні одна з односторонніх границь не існує, або є нескінченною.

Приклад 4.9. Дослідити на неперервність, знайти точки розриву, визначити їх вид та побудувати графік функції $f(x) = (x+1)arctg\frac{1}{x}$.

Дана функція невизначена в точці $x = 0$. Тому функція в цій точці розривна. Щоб визначити характер розриву, знайдемо ліву та праву границі:

```
(%i3) f(x):=(x+1)·atan( 1/x);
      limit(f(x), x, 0, minus); /·лівостороння границя·/
      limit(f(x), x, 0, plus); /·правостороння границя·/
```

(%o1) $f(x):=(x+1)\operatorname{atan}\left(\frac{1}{x}\right)$

(%o2) $-\frac{\pi}{2}$

(%o3) $\frac{\pi}{2}$

Оскільки ліва і права границі існують та не є рівними, то в точці $x = 0$ функція має неусувний розрив I роду. Стрибок знаходимо як різницю правої та лівої границі функції $\delta = \pi$:

```
(%i4) delta: (%o3)−(%o2) ; /·стрибок в точці x=0·/
```

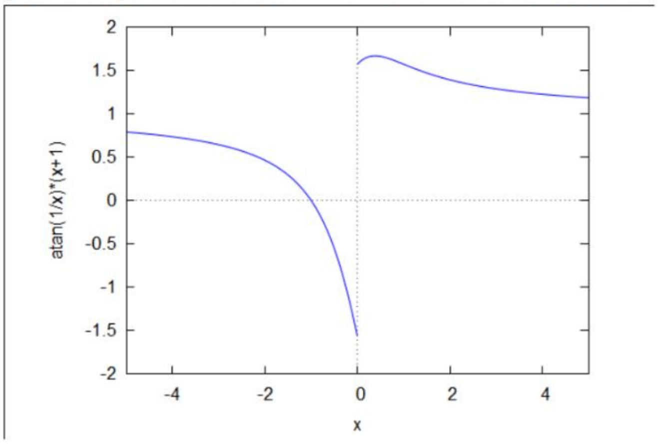
(delta) π

Побудуємо графік даної функції, на якому бачимо розрив в точці $x = 0$:

```
(%i5) wxplot2d(f(x), [x,−5,5]);
```

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t5)



(%o5)

Приклад 4.10. Дослідити на неперервність, знайти точки розриву, визначити їх вид та побудувати графік функції $f(x) = \frac{e^{2x}-1}{3x}$.

Дана функція невизначена в точці $x = 0$, для встановлення характеру розриву знаходимо ліву та праву границі.

```
(%i3) f(x):=((%e)^(2·x)−1)/(3·x);
      limit(f(x), x, 0, minus); /·лівостороння границя·/
      limit(f(x), x, 0, plus); /·правостороння границя·/
```

(%o1) $f(x) := \frac{\%e^{2x} - 1}{3x}$

(%o2) $\frac{2}{3}$

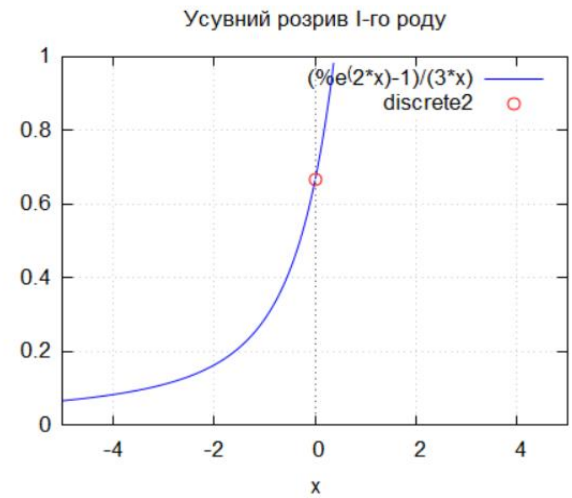
(%o3) $\frac{2}{3}$

Одержали, що ліва границя дорівнює правій границі, отже, в точці $x = 0$ усувний розрив І роду. Якщо за значення функції у точці $x = 0$ взяти число $\frac{2}{3}$, то функція стане неперервною у цій точці.

Побудуємо графік даної функції. Оскільки розрив такого типу на графіку мало помітний, то додатково зобразимо на графіку і точку розриву.

```
(%i6) x0: [0]$ y0: [2/3]$ /·координати точки розриву·/
      wxplot2d([f(x), [discrete, x0, y0]], [x,−5,5], [y,0,1], grid2d,
                      [style, lines, points],
                      [point_type, circle],
                      [color, blue, red],
                      [title, "Усувний розрив І−го роду"]);
```

(%t6)



Приклад 4.11. Дослідити на неперервність, знайти точки розриву, визначити їх вид та побудувати графік функції $f(x) = \frac{1}{x^2-4}$.

Задана функція невизначена в точках $x=2$, $x=-2$. Дослідимо якого типу розриви функція має в цих точках. Знайдемо односторонні границі в цих точках:

(%i1) f(x):=1/(x^2-4);

(%o1) $f(x):=\frac{1}{x^2-4}$

(%i3) limit(f(x), x, −2, minus); /·лівостороння границя в x=−2·/
limit(f(x), x, −2, plus); /·правостороння границя в x=−2·/

(%o2) ∞

(%o3) $-\infty$

(%i5) limit(f(x), x, 2, minus); /·лівостороння границя в x=2·/
limit(f(x), x, 2, plus); /·правостороння границя в x=2·/

(%o4) $-\infty$

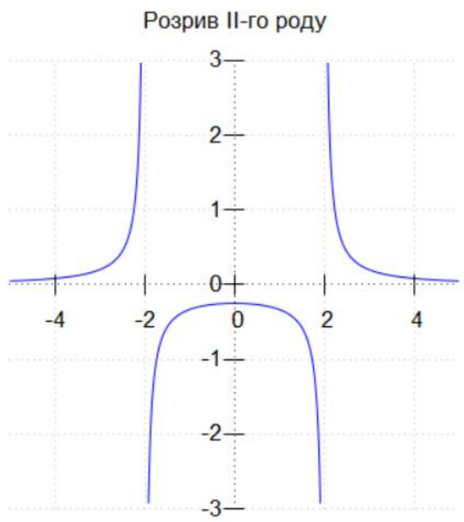
(%o5) ∞

Одержуємо, що границі в точках $x=2$, $x=-2$ дорівнюють нескінченності. Отже, в цих точках функція має розрив ІІ роду. Побудуємо графік, щоб наочно побачити розривність функції.

(%i8) wxplot2d(f(x), [x,−5,5], [y,−3,3],
[box, false], grid2d, [yx_ratio,1],
[title, "Розрив II−го роду"]);

plot2d: some values were clipped.

(%t8)



4.2 Застосування похідної

Засоби створення графічних об'єктів програми Maxima можуть успішно використовуватися для розв'язування задач на виконання повного дослідження функції та побудови її графіка і дотичних ліній; знаходження многочленів Тейлора та створення графічної демонстрації розкладу функції в ряд в залежності від степеня многочлена; обчислення найбільшого та найменшого значень функції тощо.

Геометричний зміст похідної

Нехай функція f є диференційовною в точці x_0. Тоді до графіка функції f у точці з абсцисою x_0 можна провести дотичну пряму (рис. 4.1). Загальне рівняння прямої має вигляд $y = kx + b$, де $k -$ кутовий коефіцієнт цієї прямої. Причому $k = tg\alpha$, α – кут нахилу, що утворює пряма з додатним напрямом осі Ох.

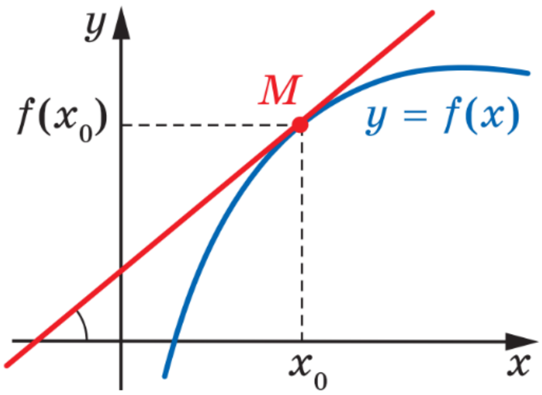


Рис. 4.1 Дотична до графіка функції в точці x_0.

За геометричним змістом похідної кутовий коефіцієнт дотичної, проведеної до графіка функції f у точці з абсцисою x_0, дорівнює значенню похідної функції у точці x_0, тобто $k = f'(x_0)$. Тоді рівняння дотичної до функції f в точці $M(x_0; f(x_0))$ має вигляд $y = f'(x_0)(x - x_0) + f(x_0)$.

Нормаллю до кривої в даній точці називається пряма, яка проходить через цю точку і перпендикулярна до дотичної в ній. Із означення нормалі випливає, що її кутовий коефіцієнт нормалі пов'язаний з кутовим коефіцієнтом дотичної рівністю: $k_{\text{нормалі}} = -\frac{1}{k_{\text{дотичної}}}$.

Приклад 4.12. Знайти рівняння дотичної та нормалі до функції $f(x) = 3x^4 - 5x^2 + 4$ в точці $x_0 = 1$.

Знаходимо похідну функції та значення функції і її похідної в точці $x_0 = 1$.

```
(%i4)  f(x):=3·x^4−5·x^2+4; /·визначаємо функцію·/
       df: diff(f(x), x); /·знаходимо похідну даної функції·/
       fx0: f(−1); /·значення функції в точці x0=−1·/
       dfx0: at(df, x=−1); /·значення похідної функції в точці x0=−1·/
```

(%o1) $f(x) := 3x^4 - 5x^2 + 4$

(df) $12x^3 - 10x$

(fx0) 2

(dfx0) -2

Складаємо рівняння дотичної, для спрощення виразу використаємо функцію rat().

```
(%i5)  dotycna: rat(−2·(x+1)+2, x);
```

(dotycna) $-2x$

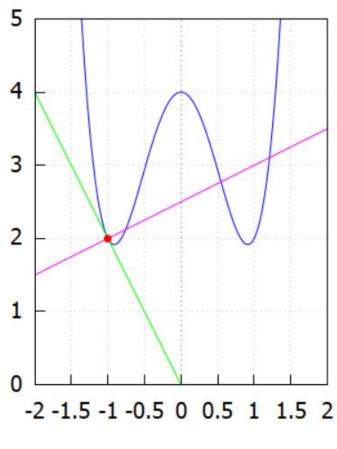
```
(%i6)  normal: rat((−1/−2)·(x+1)+2, x);
```

(normal) $\frac{x+5}{2}$

Таким чином, одержали рівняння дотичної $y = 2x$ та рівняння нормалі $y = \frac{1}{2}x + \frac{5}{2}$. Побудуємо графік функції, дотичну і нормаль за допомогою функції draw2d(). Також на графічному зображенні позначимо точку дотику.

```
(%i7)
       wxdraw2d(grid=true, proportional_axes=xy,
              xaxis=true, yaxis=true, yrange=[0,5],
              explicit(3·x^4−5·x^2+4, x,−2,2),
              color=green, explicit(dotycna, x,−2,2),
              color=magenta, explicit(normal, x,−2,2),
               color=red, point_type=7, points([[−1,2]]));
```

(%t7)



Монотонність, максимуми й мінімуми функцій.

Якщо для всіх x з проміжку $(a; b)$ виконується нерівність $f'(x) > 0$, то на цьому проміжку функція *зростає*. Якщо ж для всіх x з проміжку $(a; b)$ виконується нерівність $f'(x) < 0$, то на цьому проміжку функція *спадає*. Проміжки зростання і спадання функції називають *проміжками монотонності*.

Щоб *дослідити функцію $f(x)$ на монотонність*, слід:

1) знайти її похідну $f'(x)$;

2) знайти критичні точки (точки, в яких похідна рівна нулю або не існує);

3) визначити знак похідної функції на кожному з проміжків, на які критичні точки розбивають область визначення функції;

4) визначити проміжки зростання та спадання функції.

Функція $y = f(x)$ має в точці x_0 *максимум (мінімум)*, якщо існує такий окіл точки x_0, що для всіх x з цього околу виконується нерівність $f(x) \leq f(x_0)$ ($f(x) \geq f(x_0)$). Точки максимуму і точки мінімуму називають *точками екстремуму*. Значення функції в точці максимуму або мінімуму називають *екстремумом* (*максимумом* або *мінімумом*) *функції*.

Достатня умова екстремуму: якщо похідна функції $y = f(x)$ перетворюється в нуль в точці x_0 і при переході через цю точку в напрямку зростання x змінює знак з «+» на «-» (з «-» на «+»), то в точці x_0 функція має максимум (мінімум). Якщо ж при переході через точку x_0 похідна функції не змінює знак, то в цій точці функція $y = f(x)$ екстремуму не має (наприклад, має перегин).

За допомогою похідної другого порядку досліджують опуклість (вгнутість) функції. Якщо для всіх $x \in (a; b)$ виконується нерівність $f''(x) \geq 0$ ($f''(x) \leq 0$), то графік функції $f(x)$ є вгнутим (опуклим) на даному проміжку.

Дослідження функції на побудову її графіка виконують за схемою:

1. Знайти область визначення функції.

2. Дослідити функцію на парність, непарність, періодичність.

3. Знайти точки перетину з осями координат та проміжки знакосталості функції.

4. Знайти проміжки зростання, спадання, точки екстремуму та екстремуми функції.

5. Дослідити функцію на опуклість (вгнутість), знайти точки перегину.

6. Знайти асимптоти графіка функції. Асимптоти є вертикальні, горизонтальні та похилі.

Пряма $x = a$ є *вертикальною асимптотою* графіка функції $f(x)$, якщо $\lim\limits_{x \to a} f(x) = \infty$.

Пряма $y = a$ є *горизонтальною асимптотою* графіка функції $f(x)$, якщо $\lim\limits_{x \to +\infty} f(x) = a$ або $\lim\limits_{x \to -\infty} f(x) = a$.

Пряма $y = kx + b$ є похилою асимптотою в тому випадку, коли границі для знаходження k та b мають скінчені значення (якщо виявиться, що $k = 0$, а b — скінчене число, то асимптота буде горизонтальна); причому $k = \lim\limits_{x \to \pm\infty} \frac{f(x)}{x}$, $b = \lim\limits_{x \to \pm\infty} (f(x) - kx)$.

Приклад 4.13. Дослідіть властивості функцію $f(x) = \frac{x^3}{2(x+1)^2}$. Побудуйте графік функції, асимптоти, позначте на графіку точки екстремуму, точки перегину.

Визначаємо дану функцію в Maxima:

```
(%i2) f: x^3/(2·(x+1)^2);
      define(f(x),f); /·визначення функції·/
```

$$(f) \quad \frac{x^3}{2\,(x+1)^2}$$

$$(\%o2) \quad f(x) := \frac{x^3}{2\,(x+1)^2}$$

Область визначення – всі дійсні числа, окрім точки $x = -1$.

За допомогою функції `is(expr)`, яка повертає `true` в разі істинності виразу і `false` – в разі хибності, перевіримо функція парна чи непарна:

```
(%i4) is(f=subst(x=-x, f));
      is(f=subst(x=-x, -f));
(%o3) false
(%o4) false
```

Отже, функція ні парна, ні непарна.

Знаходимо вертикальні та горизонтальні асимптоти. Обчислюємо границі:

```
(%i8) limit(f,x,minf);
      limit(f,x,inf);
      limit(f,x,-1,minus);
      limit(f,x,-1,plus);

(%o5) -∞
(%o6) ∞
(%o7) -∞
(%o8) -∞
```

При $x \to \pm\infty$ дана функція не має скінченної границі, тому горизонтальних асимптот немає. Оскільки правостороння та лівостороння границі функції при $x \to -1$ прямують до мінус нескінченності, то пряма $x = -1$ є вертикальною асимптотою.

Обчислюємо коефіцієнти похилих асимптот $y = kx + b$, де $k = \lim\limits_{x\to\infty} \frac{f(x)}{x}$, $b = \lim\limits_{x\to\infty}(f(x) - kx)$:

```
(%i10) km: limit(f/x,x,minf); /·ліва границя·/
       kp: limit(f/x,x,inf);  /·права границя·/
(km)   1/2
(kp)   1/2
(%i12) bm: limit(f-km·x,x,minf); /·ліва границя·/
       bp: limit(f-kp·x,x,inf); /·права границя·/
(bm)   -1
(bp)   -1
```

Одержали $k = \frac{1}{2}$, $b = -1$, тобто $y = \frac{1}{2}x - 1$ – похила асимптота.

Знаходимо точки екстремуму та інтервали зростання і спадання функції.

Для цього обчислюємо похідну:

```
(%i13) df: diff(f,x); /·похідна·/
```

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

Знаходимо критичні точки, тобто точки, в яких похідна дорівнює нулю або не існує:

```
(%i15) solve(df,x); /*точки, в яких похідна дорівнює 0*/
       solve(1/df,x); /*точки, в яких похідна не існує*/
```

(%o14) $[x=-3, x=0]$

(%o15) $[x=-1]$

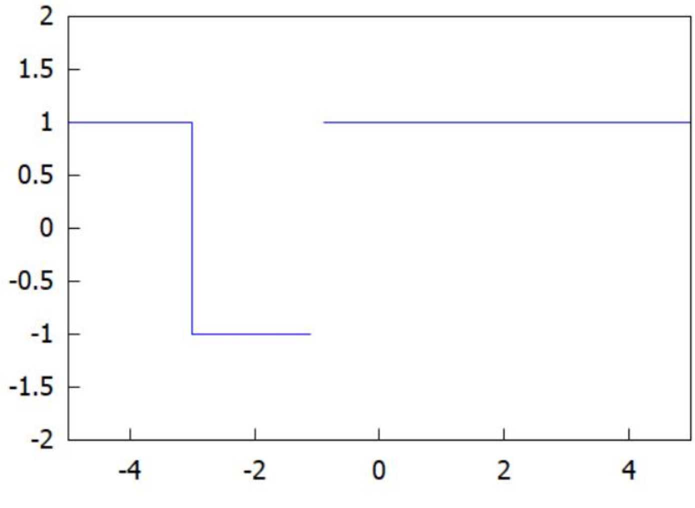
Визначимо знак похідної на кожному з проміжків знакосталості графічним способом, використовуючи функцію `signum()`, яка повертає значення 1 або –1, або 0, якщо функція відповідно додатна, від'ємна, рівна нулю. Знак похідної можна визначити за допомогою команди:

Побудуємо графік функції sdf, обійшовши особливу точку $x=-1$, використовуючи процедури пакету draw:

```
(%i16) sdf: signum(df)$

(%i17) wxdraw2d(
         yrange=[-2,2], /*діапазон значень вздовж осі Oy*/
         explicit(sdf,x,-5,-1.1),
         explicit(sdf,x,-0.9,5))$
```

(%t17)



Одержали, що на проміжках $(-\infty; -3)$ і $(-1; +\infty)$ похідна функції додатна, а на проміжку $(-3; -1)$ – від'ємна. Тому точка $x = -3$ – точка максимуму, в точці $x=-1$ функція невизначена, а в точці $x = 0$ екстремуму немає.

Знаходимо значення функції в точках $x = -3$ та $x = 0$:

```
(%i19) f(−3); f(0);
```

(%o18) $-\frac{27}{8}$

(%o19) 0

Таким чином, $\left(-3; -\frac{27}{8}\right)$ – точка максимуму.

Знайдемо точки перегину та інтервали вгнутості, опуклості. Обчислюємо другу похідну та її критичні точки.

Досліджуємо знаки другої похідної графічно:

```
(%i20) ddf: diff(df,x); /·похідна другого порядку функції f·/
```

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

Знаходимо критичні точки для похідної другого порядку:

```
(%i22) solve(ddf,x); /·точки, в яких друга похідна дорівнює 0·/
       solve(1/ddf,x); /·точки, в яких друга похідна не існує ·/
```

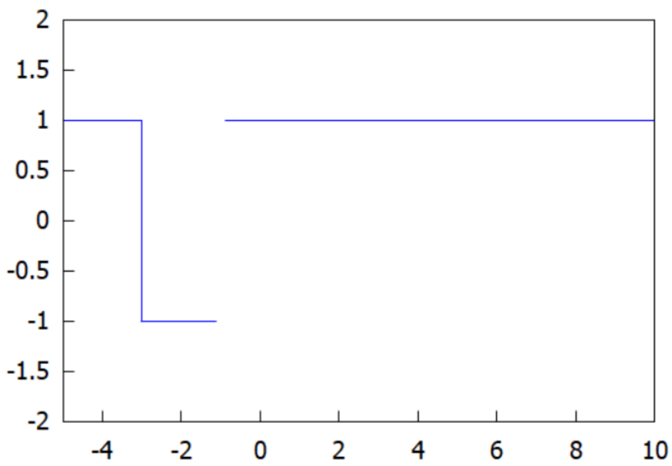
(%o21) $[x=0]$

(%o22) $[x=-1]$

Визначаємо знак похідної другого порядку на проміжках знакосталості, дивимося графічну інтерпретацію:

```
(%i24) sddf: signum(ddf)$ /·знак похідної другого порядку·/
       wxdraw2d(yrange=[−2,2],
         explicit(sdf,x,−5,−1.1),
         explicit(sdf,x,−0.9,10));
```

(%t24)



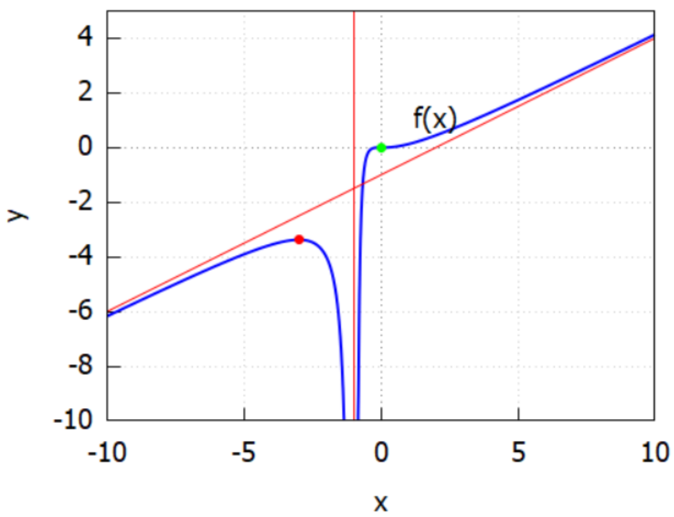
Отже, $x=0$ є точка перегину і на проміжку $(-\infty;-1)\cup(-1;0)$ функція опукла вгору, а на проміжку $(0;\infty)$ – опукла вниз.

Використовуючи процедуру `wxdraw2d()`, створимо графічне зображення, на якому буде графік функції (синій), асимптоти (червоні), точка максимуму (червона) і точка перегину(зелена):

```
(%i26) text: label(["f(x)",2,1])$ /·підпис на графіку·/
       wxdraw2d(
         proportional_axes=xy,/·пропорційна система координат·/
         yrange=[−10,5], /·діапазон значень вздовж осі Oy·/
         grid=true, /·відображення ліній сітки·/
         xaxis=true, yaxis=true,/·відображення осей Ox, Oy ·/
         xaxis_type=dots,yaxis_type=dots,/·тип лінії осей ·/
         xlabel="x", ylabel="y", /·підписи осей Ox та Oy ·/
         line_width=2, /·товщина лінії графічного об’єкта·/
         explicit(f,x,−10,10), /·графік явно заданої функції·/
         color=red, /·колір графічного об’єкта (асимптоти)·/
         line_width=1,/·товщина лінії графічного об’єкта (асимптоти)·/
         parametric(−1,y,y,−10,10), /·вертикальна асимптота·/
         explicit(km·x+bm, x,−10,10), /·похила асимптота·/
         color=black,text, /·підпис на графіку, його колір·/
         color=red, /·колір графічного об’єкта (точки)·/
         point_type=filled_circle, /·тип точки·/

         points([[−3,subst(x=−3,f)]]), /·точка максимуму·/
         color=green, point_type=filled_circle,
         points([[0, subst(x=0,f)]]))$/·точка перегину·/
```

(%t26)



4.3 Інтегрування, застосування визначеного інтеграла

Фігура, що обмежена графіком функції $y = f(x)$ і прямими $y = 0$, $x = a$, $x = b$ називають криволінійною трапецією (рис. 4.2). Площу криволінійної трапеції обчислюють за формулою $S = \int_a^b f(x)dx$.

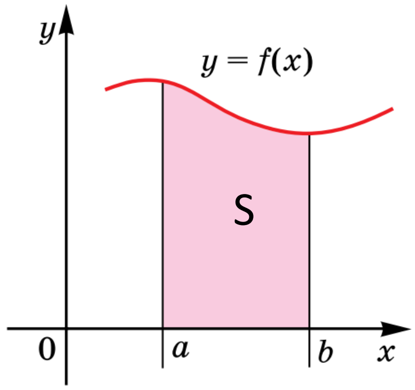


Рис. 4.2 Криволінійна трапеція

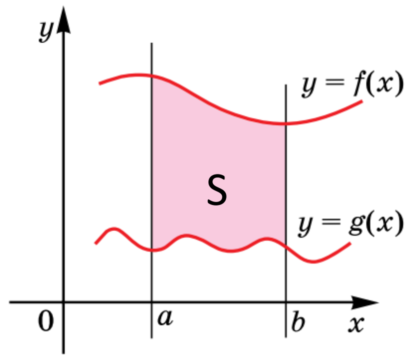


Рис. 4.3 Фігура, обмежена лініями

Якщо на проміжку $[a; b]$ $f(x) \leq 0$, то площу криволінійної трапеції обчислюють за формулою $S = \left|\int_a^b f(x)dx\right|$.

Нехай $y = f(x)$ та $y = g(x)$ – неперервні на проміжку $[a; b]$ функції і для всіх $x \in [a; b]$ виконується нерівність $f(x) \geq g(x)$. Тоді площу фігури, обмеженої функціями $y = f(x)$ та $y = g(x)$ на проміжку $[a; b]$ (рис. 4.3) обчислюють за формулою $S = \int_a^b (f(x) - g(x))dx$.

Приклад 4.14. Побудувати криворінійну трапецію, що обмежена функцією $f(x)$=$e^{-0,2x}\cos x$ на проміжку $\left[0; \frac{\pi}{2}\right]$. Обчислити площу заданої фігури.

Визначаємо задану функцію. Для спрощення символьних перетворень число 0,2 подамо у вигляді звичайного дробу $\frac{1}{5}$.

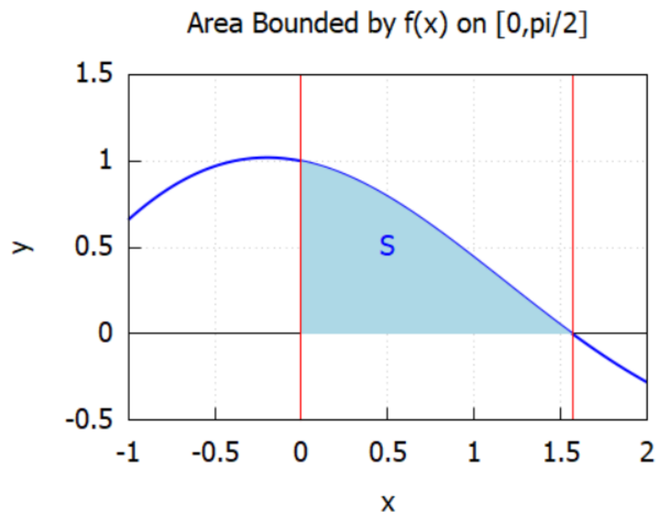
```
(%i1) f(x):=((%e)^(-(1/5)·x))·cos(x);
```

(%o1) $f(x) := \%e^{\left(-\frac{1}{5}\right)x}\cos(x)$

Використовуючи функцію `draw2d()`, створюємо зображення графіка заданої функції та заданої криволінійної трапеції.

```
(%i2)  wxdraw2d(proportional_axes=xy,/·пропорційна система координат·/
         yrange=[−0.5,1.5], /·діапазон значень вздовж осі Оу·/
         grid=true, /·відображення ліній сітки·/
         xaxis=true, yaxis=true,/·відображення осей Ох, Оу ·/
         xaxis_type=solid,yaxis_type=solid,/·тип лінії осей ·/
         xlabel="x", ylabel="y", /·підписи осей Ох та Оу ·/
         title="Area Bounded by f(x) on [0,pi/2]",
         line_width=2, color=blue,/·товщина лінії графічного об’єкта·/
         explicit((%e)^(−(1/5)·x)·cos(x),x,−1,2), /·графік явно заданої функції·/
         fill_color=light_blue, /·колір заливки фігури·/
         filled_func=true, /·параметр графічного поля, обмеженого f(x)·/
         filled_func=f(x), explicit(0,x,0,%pi/2), /· графічне поле, обмежене f(x) від 0 до π/2·/
         label(["S",0.5,0.5]),
         color=red, /·колір вертикальних прямих·/
         line_width=1,/·товщина вертикальної лінії ·/
         parametric(π/2,y,y,−0.5,1.5), /·вертикальна лінія x=π/2·/
         parametric(0,y,y,−0.5,1.5));/·вертикальна лінія x=0·/
```

(%t2)



Обчислюємо площу заданої криволінійної трапеції:

```
(%i3)  S: integrate(f(x), x, 0, π/2);
```

(S) $$\frac{25\,\%e^{-\frac{\pi}{10}}}{26}+\frac{5}{26}$$

```
(%i4)  float(S);
(%o4)  0.8946179721621592
```

Приклад 4.15. Обчислити площу фігур, що обмежені лініями $y = 3x - x^2$ та $y = x^2 - 2$. Побудувати зображення фігури.

Визначаємо задані функції.

```
(%i2) f(x):=3·x−x^2;
      g(x):=x^2−2;
```

(%o1) $f(x):=3x-x^2$

(%o2) $g(x):=x^2-2$

Визначаємо інтервал інтегрування. Знаходимо точки перетину графіків, прирівнявши вирази заданих функцій і розв’язуємо відповідне рівняння:

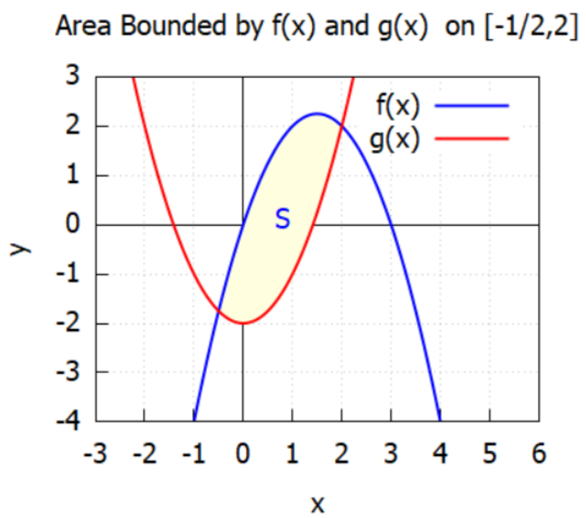
```
(%i3) solve(3·x−x^2=x^2−2);
```

(%o3) $[x=-\frac{1}{2}, x=2]$

Використовуючи функцію `draw2d()`, створюємо зображення графіків заданих функцій та зафарбовуємо фігуру, обмежену функціями $f(x)$ та $g(x)$ на проміжку $\left[-\frac{1}{2}; 2\right]$, додаємо написи в графічній області:

```
(%i6) wxdraw2d(proportional_axes=xy,/·пропорційна система координат·/
   yrange=[−4,3], /·діапазон значень вздовж осі Oy·/
   grid=true, /·відображення ліній сітки·/
   xaxis=true, yaxis=true,/·відображення осей Ox, Oy ·/
   xaxis_type=solid,yaxis_type=solid,/·тип лінії осей ·/
   xlabel="x", ylabel="y", /·підписи осей Ox та Oy ·/
   title="Area Bounded by f(x) and g(x)  on [−1/2,2]",
   fill_color=light_yellow, filled_func=true, filled_func=f(x), /·заливка фігури·/
   explicit(g(x),x,−1/2,2), filled_func=false, /·заливка фігури·/
   label(["S",0.8,0.1]),
   line_width=2, key="f(x)",/·товщина лінії графічного об’єкта·/
   color=blue, explicit(3·x−x^2, x, −3, 6), /·графік явно заданої функції·/
   key="g(x)", color=red, explicit(x^2−2, x, −3, 6));
```

(%t6)



За графіком помічаємо, що $f(x) \geq g(x)$ на проміжку $\left[-\frac{1}{2}; 2\right]$.

Обчислюємо площу обмеженої фігури S за допомогою визначеного інтеграла. Знаходимо відповідь у вигляді звичайного дробу та обчислюємо наближене чисельне значення у вигляді десяткового дробу:

```
(%i7) S: integrate((f(x)-g(x)), x, -1/2, 2);
(S)   125/24
(%i8) S, numer;
(%o8) 5.208333333333333
```

Задачі для самостійного розв'язування

1. Обчислити границю функції.

1.	$\lim\limits_{x\to 0}\frac{1+\sin x-\cos x}{1-\sin x-\cos x}$	6.	$\lim\limits_{x\to 0}\frac{arctg(2x)}{5x}$
2.	$\lim\limits_{x\to 0}\frac{1-\cos^3 x}{x\sin(2x)}$	7.	$\lim\limits_{x\to -1}\frac{\sin(x+1)}{1-x^2}$
3.	$\lim\limits_{x\to\infty}(\sqrt{x^2+1}-\sqrt{x^2-1})$	8.	$\lim\limits_{x\to 1}(3x-2)^{\frac{5x}{x^2-1}}$
4.	$\lim\limits_{n\to\infty}\frac{\sqrt[3]{n^3+2n-1}}{n+2}$	9.	$\lim\limits_{x\to 1}(3x-5)^{\frac{2x}{x^2-4}}$
5.	$\lim\limits_{n\to\infty}\left(\frac{n+1}{n-1}\right)^x$	10.	$\lim\limits_{x\to 0}\frac{1-\cos 8x}{1-\cos 4x}$

2. Дослідити на неперервність, знайти точки розриву, визначити їх вид та побудувати графік функції.

1.	$y=\frac{\lvert x+1\rvert}{x+1}x-1$	6.	$y=\frac{x+2}{\lvert x+2\rvert}+x$
2.	$y=\frac{x}{x^2-9}$	7.	$y=\frac{x^2+1}{x}$
3.	$y=\frac{4}{x^2-2x+1}$	8.	$y=\frac{x^2-3x}{x-4}$
4.	$y=\frac{x}{x^3-3x^2-4x}$	9.	$y=\begin{cases}\frac{1}{x}, x\leq -2,\\ x^2 > -2\end{cases}$
5.	$y=e^{\frac{1}{x+1}}$	10.	$y=\begin{cases}-x, x\leq -1,\\ \frac{2}{x-1} > -1\end{cases}$

3. Знайти похідну функції, обчислити значення похідної в точці та знайти рівняння дотичної і нормалі до функції в цій точці. Побудувати графік функції, дотичну та нормаль в заданій точці.

1.	$f(x) = \left(\sqrt{x} - 1\right)^5,\ x_0 = 4$	6.	$f(x) = \sqrt[3]{(x-2)^2},\ x_0 = 2$
2.	$f(x) = \left(\sqrt{x} + 2\right)^4,\ x_0 = 1$	7.	$f(x) = e^{x^2-5x+6},\ x_0 = 1$
3.	$f(x) = \sqrt{5x^2 - 2x},\ x_0 = 2$	8.	$f(x) = 5^{3x-4},\ x_0 = 2$
4.	$f(x) = \sqrt{-x^2 + 5x - 3},\ x_0 = 1$	9.	$f(x) = \frac{1}{6}x^3 + 4x,\ x_0 = -2$
5.	$f(x) = \cos^3 x,\ x_0 = \pi$	10.	$f(x) = x^4 - x^2 + x - 7,\ x_0 = -1$

4. Виконайте повне дослідження функції, побудуйте її графік та асимптоти.

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

5. Обчисліть невизначений інтеграл.

1.	$\int x\cos^2 x dx$	6.	$\int \frac{\ln x}{x^2} dx$
2.	$\int x e^{-2x} dx$	7.	$\int x \sin\frac{x}{2} dx$
3.	$\int x\cos 3x dx$	8.	$\int x\sin(5x) dx$
4.	$\int \frac{1}{36+5x^2} dx$	9.	$\int x^2 e^{3x} dx$
5.	$\int x^2 \ln(1+x) dx$	10.	$\int \frac{x}{\sin^2 x} dx$

11.	$\int \frac{1}{2+\sqrt{x}}dx$	12.	$\int \frac{x^2}{x^2+1}dx$

6. Обчисліть інтеграл у символьному та чисельному вигляді.

1.	$\int_4^9 \frac{\sqrt{x}}{\sqrt{x}-1}dx$	7.	$\int_1^2 x\ln(x+1)dx$
2.	$\int_1^{\sqrt{3}} \frac{1}{x^2\sqrt{1+x^2}}dx$	8.	$\int_0^1 x^2 e^{-2x}dx$
3.	$\int_1^2 \frac{1}{x+x^3}dx$	9.	$\int_0^{\frac{\pi}{2}} (x^2-1)\sin 2x dx$
4.	$\int_{-1}^0 \frac{1}{x^2+2x+2}dx$	10.	$\int_0^1 x\ln(x^2+1)dx$
5.	$\int_0^{\frac{\pi}{2}} \sin\frac{x}{2}dx$	11.	$\int_1^e \ln^2 x dx$
6.	$\int_0^{\frac{\pi}{4}} x\mathrm{tg}^2 x dx$	12.	$\int_1^2 \frac{\ln x}{x^5}dx$

7. Побудуйте графік фігури, обмеженої лініями, обчисліть площу цієї фігури.

1) параболою $y = x^2 - 2$ і прямою $y = x$.
2) гіперболою $xy = 3$ і прямою $y = 4 - x$.
3) параболами $y = 4 - x^2$ і $y = x^2 - 2x$.
4) параболою $y = 5x - 2x^2$ і прямою $y = 2x - 2$.
5) параболами $y = 8 - x^2$ і $y = x^2$.
6) кривими $y = \frac{8}{4+x^2}$ та $y = \frac{x^2}{4}$.
7) параболою $y = 2x^2 + 6x$ і прямою $y = x - 2$.
8) параболами $y = 3 - x^2$ і $y = (x-1)^2$.
9) кривою $y = \sqrt{x+1}$ і прямою $y = 0.5x + 0.5$.
10) кривою $y = -\sqrt{2x}$ і прямою $y = -x$.
11) параболою $y = -x^2 + 4x$ і прямою $y = 4 - x$.
12) параболою $y = -x^2 - x + 2$ і прямою $y = 2 + x$.

Контрольні питання

1. Яка функція призначена для обчислення значення невизначеного інтеграла?

a) integrat(); b) integrate(); c) integral(); d) int().

2. Який запис команди для обчислення границі $\lim\limits_{x \to 0} \frac{x}{\sin x}$ правильний?

a) limit(x/sin(x), x, 0);

b) lim(x/sin(x), x, 0);

c) limite(x/sin(x), x, 0);

d) limit(x/sin(x), 0);.

3. Виберіть правильну команду, щоб знайти похідну функції $f(x) = x^3 - 2x^2 + 5x - 3$.

a) diff(x^3 - 2x^2 + 5x - 3, x);

b) derivative(x^3 - 2x^2 + 5x - 3, x);

c) dx(x^3 - 2x^2 + 5x - 3);

d) df(x^3 - 2x^2 + 5x - 3, x);

4. Яку команду потрібно виконати, щоб знайти точки екстремуму функції $f(x) = x^3 - 6x^2 + 9x + 2$?

a) solve(diff(x^3 - 6x^2 + 9x + 2, x) = 0, x);

b) minmax(x^3 - 6x^2 + 9x + 2, x);

c) extremum(x^3 - 6x^2 + 9x + 2, x);

d) find(x^3 - 6x^2 + 9x + 2, x);

5. Яку команду потрібно виконати, щоб знайти визначений інтеграл $\int_1^2 e^{-x^2} dx$?

a) area(e^(-x^2), x, 1, 2);

b) int(e^(-x^2), x, 1, 2);

c) integrat(e^(-x^2), dx, 1, 2)

d) integrate(e^(-x^2), x, 1, 2);

6. Яку команду потрібно виконати, щоб знайти абсциси точок перетину графіків функцій $f(x) = x^2$, $g(x) = -2x - 1$.

a) solve(x^2 = -2x - 1, x);

b) intersections(x^2, 2x - 1, x);

c) solve[[x^2 = 2x - 1], [x]];

d) plot([x^2, 2x - 1], [x]);

7. Щоб побудувати вертикальну асимптоту $x = \pi$ за допомогою функції draw2d(), потрібно створити графічний об'єкт:

a) implicit(π, x, x, -2, 2);

b) line(π, x, x, -2, 2);

c) parametric(π, x, x, -2, 2);

d) parametric(π, y, y, -2, 2);

8. Що визначають за допомогою функції signum(f) ?

a) проміжки знакосталості функції;

b) знак функції;

c) значення функції;

d) область визначення функції;

9. Яка функція призначена для розкладу функції в ряд?

a) sum();

b) range();

c) series();

d) taylor();

10. Яку команду потрібно виконати, щоб знайти площу фігури, обмеженої лініями $f(x) = x^2 - 2$ та $g(x) = -x^2 + 1$?

a) integrate(f(x)-g(x), x, -2,1);

b) integrate(g(x)-f(x), x, -2,1);

c) integrate(g(x)-f(x), x, -1,2);

d) integrate(f(x)-g(x), x, -1,2);

Література

1. Бугаєць Н.О. Засоби програми Maxima для створення графічних зображень та математичних досліджень // Науковий часопис НПУ ім. М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання. К.: НПУ ім. М.П. Драгоманова. 2015. №15(22). С. 105 – 114.

2. Бугаєць Н.О. Моделювання анімаційних наочностей засобами графічного середовища програми Maxima [Електронний ресурс] / Н.О.Бугаєць // Інформаційні технології і засоби навчання. 2015. Том 47, №3. С.67 – 79. – Режим доступу: http://journal.iitta.gov.ua/index.php/itlt/article/view/1244

3. Дюженкова Л.І., Дюженкова О.Ю., Михалін Г.О. Вища математика: Приклади і задачі. – К.: Видавничий центр «Академія», 2002. – 624 с.

4. Жалдак М.І. Математика з комп'ютером / Жалдак М.І., Горошко Ю.В., Вінниченко Є.Ф.; Посібник для вчителів. 3-тє видання. К.: Видавництво НПУ імені М.П.Драгоманова, 2015. 315 с.

5. Жалдак М.І. Математичний аналіз з елементами інформаційних технологій: навч. посібник / М.І.Жалдак, Г.О.Михалін, С.Я.Деканов. К.: Редакції газет природничо-математичного циклу, 2012. 128 с.

6. Жалдак М.І., Триус Ю.В. Основи теорії і методів оптимізації. Навчальний посібник. – Черкаси: Брама-Україна, 2005. – 608 с.

7. Інноваційні інформаційно-комунікаційні технології навчання математики: навчальний посібник / В.В. Корольський, Т.Г. Крамаренко, С.О. Семеріков, С.В. Шокалюк; науковий редактор академік АПН України, д. пед. н., проф. М.І. Жалдак. – Кривий Ріг: Книжкове видавництво Кирєєвського, 2009. –316 с.

8. Кобильник Т.П., Когут У.П. Використання системи Maxima для розв'язування оптимізаційних задач на графах // Науковий часопис НПУ ім. М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання. – К.: НПУ ім. М.П. Драгоманова. 2011. №12(19). С. 62 – 67.

9. Клепко В.Ю., Голець В.Л. Вища математика в прикладах і задачах: Навчальний посібник. 2+ге видання. К.: Центр учбової літератури, 2009. 594 с.

10. Компьютерная математика с Maxima: Руководство для школьников и студентов / Е.А.Чичкарев. М.: ALT Linux, 2012. 384 с.

11. Maxima Manual Version 5.41.0. 2019. 1172 p.

12. The Maxima Book Paulo Ney de Souza Richard J. Fateman Joel Moses Cliff Yapp. 2003. 121 p.

13. Zachary Hannan. wxMaxima for Calculus II. Solano Community College. 2015. 176 p.

14. Wilhelm Haager. Graphics with MAXIMA. HTL St. Pölten, Department Electrical Engineering, 2011. 34 p.

ДОДАТОК 1

Значення опції colorname для гафічних об’єктів інтерфейсу Draw

`color=colorname`

Значення `colorname`:

white	black	gray0	grey0
gray10	grey10	gray20	grey20
gray30	grey30	gray40	grey40
gray50	grey50	gray60	grey60
gray70	grey70	gray80	grey80
gray90	grey90	gray100	grey100
gray	grey	light_gray	light_grey
dark_gray	dark_grey	red	light_red
dark_red	yellow	light_yellow	dark_yellow
green	light_green	dark_green	spring_green
forest_green	sea_green	blue	light_blue
dark_blue	midnight_blue	navy	medium_blue
royalblue	skyblue	cyan	light_cyan
dark_cyan	magenta	light_magenta	dark_magenta
turquoise	light_turquoise	dark_turquoise	pink
light_pink	dark_pink	coral	light_coral
orange_red	salmon	light_salmon	dark_salmon
aquamarine	khaki	dark_khaki	goldenrod
light_goldenrod	dark_goldenrod	gold	beige
brown	orange	dark_orange	violet
dark_violet	plum	purple	

Навчальне видання

Бугаєць Н.О.

МАТЕМАТИКА З ПРОГРАМОЮ MAXIMA

Навчальний посібник

Технічний редактор – І. П. Борис
Верстка, макетування – В. М. Косяк

Друкується за авторським редагуванням

Підписано до друку 06.08.23	Формат 60х84/16	Папір офсетний
Гарнітура Arial	Обл.-вид. арк. 4,74	Тираж 30 прим.
Замовлення №	Ум. друк. арк. 9,53	

Ніжинський державний університет
імені Миколи Гоголя.
м. Ніжин, вул. Воздвиженська, 3А
(04631)7-19-72
E-mail: vidavn_ndu@ukr.net
www.ndu.edu.ua

Свідоцтво суб'єкта видавничої справи
ДК № 2137 від 29.03.05 р.