

# wxMaxima

## Contents

Підручник користувача wxMaxima	3
1 Вступ до wxMaxima	4
1.1 Maxima і wxMaxima	4
1.1.1 Maxima	4
1.1.2 wxMaxima	4
1.2 Основи роботи із книгою	5
1.2.1 Використання робочих книг	5
1.2.2 Комірки	7
1.2.3 Горизонтальний і вертикальний курсори	9
1.2.4 Надсилання комірок для обчислення у Maxima	9
1.2.5 Автоматичне доповнення команд	9
1.2.6 Бічні панелі	12
1.2.7 Виведення коду MathML	12
1.2.8 Підтримка Markdown	14
1.2.9 Клавіатурні скорочення	14
1.2.10 Код TeX у експортованому TeX	15
1.3 Формати файлів	15
1.3.1 .mac	15
1.3.2 .wxm	15
1.3.3 .wxmx	15
1.4 Параметри налаштування	15
1.4.1 Типова частота кадрів анімації	17
1.4.2 Типовий розмір креслення для нових сеансів maxima	18
1.4.3 Перевіряти баланс дужок у тексті	18
1.4.4 Не зберігати робочий аркуш автоматично	18
1.4.5 Де зберігаються налаштування?	18
2 Розширення Maxima	19
2.1 Змінні із індексами	19
2.2 Візуальний супровід дій користувача у смужці стану	19
2.3 Креслення	20
2.3.1 Вбудовування креслення до робочого аркуша	20
2.3.2 Збільшення або зменшення вбудованих креслень	20
2.3.3 Креслення кращої якості	21

2.3.4	Відкриття вбудованих креслень в інтерактивних вікнах <code>gnuplot</code> . . . . .	21
2.3.5	Відкриття консолі команд <code>gnuplot</code> у вікнах <code>plot</code> . . . . .	21
2.3.6	Вбудовування анімацій до робочого аркуша . . . . .	21
2.3.7	Відкриття декількох креслень у вікнах одночасно . . . . .	23
2.3.8	Бічна панель «Накреслити за допомогою <code>Draw</code> » . . . . .	24
2.4	Вбудовування графіків . . . . .	26
2.5	Файли запуску . . . . .	26
2.6	Спеціальні змінні <code>wx...</code> . . . . .	26
2.7	Структурний друк двовимірних даних . . . . .	27
2.8	Звітування про вади . . . . .	27
2.9	Як зробити так, щоб виведені дані було показано червоним кольором? . . . . .	28
3	Вирішення проблем . . . . .	28
3.1	Не вдається з'єднатися із <code>Mathima</code> . . . . .	28
3.2	Як зберегти дані із пошкодженого файла <code>.wxmtx</code> . . . . .	29
3.3	Хочу бачити діагностичні відомості на екрані, перш ніж команда завершить свою роботу . . . . .	29
3.4	На кресленні показано лише закриту порожню обгортку із повідомленням про помилку . . . . .	29
3.5	Спроба накреслити анімацію призводить до повідомлення «помилка: невизначена змінна» . . . . .	30
3.6	Втрачено вміст комірок і скасування дій не допомагає . . . . .	30
3.7	<code>wxMathima</code> запускається із повідомленням «Процес <code>Mathima</code> несподівано перервав роботу.» . . . . .	31
3.8	<code>Mathima</code> висне при обчислення і не відповідає на команди . . . . .	31
3.9	Версії <code>Mathima</code> на основі <code>SBCL</code> не вистачає оперативної пам'яті . . . . .	31
3.10	При введенні даних програма уповільнюється або пропускає символи в <code>Ubuntu</code> . . . . .	31
3.11	<code>wxMathima</code> припиняє роботу, коли <code>Mathima</code> обробляє грецькі символи або умляути . . . . .	31
3.12	Креслення . . . . .	33
3.12.1	Чи можна зробити так, щоб <code>wxMathima</code> виводила дані одразу до файлів зображень і до вбудованих до аркуша креслень? . . . . .	33
3.12.2	Чи можна встановити співвідношення розмірів зображення креслення? . . . . .	33
4	Поширені питання та відповіді на них . . . . .	34
4.1	Чи є якийсь спосіб умістити більше тексту на сторінку <code>LaTeX</code> ? . . . . .	34
4.2	Чи передбачено у програмі «темний» режим? . . . . .	34
4.3	<code>wxMathima</code> іноді повисає на декілька секунд під час першої хвилини роботи . . . . .	34
5	Параметри командного рядка . . . . .	34

## Підручник користувача wxMaxima

wxMaxima — графічний інтерфейс користувача до системи комп'ютерної алгебри (СКА) Maxima. За допомогою wxMaxima можна користуватися усіма функціями Maxima. Крім того, у програмі передбачено зручні допоміжні програми для доступу до найпоширеніших можливостей СКА. У цьому підручнику наведено опис деяких можливостей, які роблять wxMaxima одним із найпопулярніших графічних інтерфейсів Maxima.



Figure 1: Логотип wxMaxima

Перш ніж ми перейдемо до суті, декілька зауважень щодо навігації підручником. Нижче наведено коротку таблицю змісту. Натискання будь-якого з елементів у списку переведе курсор до початку відповідного розділу підручника. Натискання пункту Зміст переведе курсор до розширеної таблиці змісту. Цією розширеною таблицею можна скористатися для точнішого переходу до окремих частин підручника. У текст буде розташовано посилання на таблицю змісту з навігаційною метою.

- Вступ:

Основи wxMaxima

- Розширення:

Команди, які wxMaxima додає до Maxima

- Усування вад:

Що робити, якщо wxMaxima працює неправильно

- Поширені питання:

Поширені питання та відповіді на них

- Командний рядок:

Параметри командного рядка, підтримку яких передбачено у wxMaxima

# 1 Вступ до wxMaxima

## 1.1 Maxima і wxMaxima

У системі програмного забезпечення з відкритим кодом великі системи, зазвичай, діляться на менші проекти, з якими простіше працювати невеликим групам розробників. Наприклад, програма для запису компакт-дисків складається з засобу командного рядка, який, власне, виконує запис компакт-диска, і графічного інтерфейсу користувача, за допомогою користувачі можуть виконати потрібні їм дії, не запам'ятовуючи параметрів командного рядка і, загалом, взагалі не маючи справи із командним рядком. Однією із переваг цього підходу є те, що робота з розробки засобу командного рядка може бути використана багатьма програмами: той самий засіб командного рядка для запису компакт-дисків можна використати у додатку надсилання даних на компакт-диск у програмі для керування файлами, для функції «записати на компакт-диск» у музичному програвачі та для запису компакт-диска у програмі для резервного копіювання на DVD. Ще однією перевагою є поділ одного великого завдання на менші частини, що надає змогу розробникам надавати декілька інтерфейсів користувача для однієї програми.

Система комп'ютерної алгебри (СКА), подібна до Maxima, дуже добре вкладається у цю схему. СКА може надавати логіку до програми-калькулятора із довільною точністю обчислень або може виконувати автоматичні перетворення формул у фоновому режимі для більших систем (наприклад, Sage). Крім того, нею можна користуватися і безпосередньо, як самостійною системою. Доступ до Maxima можна отримати з командного рядка. Втім, часто інтерфейси, подібні до wxMaxima, є набагато ефективнішим способом доступу до програмного забезпечення, особливо для початківців.

### 1.1.1 Maxima

Maxima є повноцінною системою комп'ютерної алгебри (СКА). СКА — програма, яка може розв'язувати математичні задачі перетворенням формул і визначенням формули, яка розв'язує задачу, замість простого виведення числової форми результату. Іншими словами, Maxima може слугувати калькулятором, який визначає числові значення змінних, а може давати аналітичні розв'язки. Крім того, у програмі передбачено багато обчислювальних методів для аналізу рівнянь і систем рівнянь, які не може бути розв'язано аналітично.

Із докладною документацією до Maxima можна ознайомитися у інтернеті. Доступ до частини цієї документації можна отримати з меню «Довідка» у wxMaxima. Якщо ви натиснете клавішу довідки (у більшості систем це клавіша F1), використання контекстної довідки wxMaxima відкриє для вас сторінку підручника з Maxima із довідкою щодо команди, у якій перебуває курсор.

### 1.1.2 wxMaxima

wxMaxima — графічний інтерфейс користувача, який надає доступ до усіх функціональних можливостей і гнучкості Maxima. wxMaxima надає у розпорядження

```

dauti : maxima — Konsole
Maxima 5.41.0 http://maxima.sourceforge.net
using Lisp GNU Common Lisp (GCL) GCL 2.6.12
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) R_parallel(R1,R2):=R1*R2/(R1+R2);
(%o1)
          R1 R2
R_parallel(R1, R2) := -----
          R1 + R2
(%i2) Values:[
          R1=10,
          R2=30
];
(%o2) [R1 = 10, R2 = 30]
(%i3) at(R_parallel(R1,R2),Values);
(%o3)
          15
          --
          2
(%i4) █

```

Figure 2: Знімок вікна Maxima, командний рядок

користувачам графічні засоби роботи та можливості, які значно спрощують роботу з Maxima. Наприклад, у wxMaxima передбачено можливість експортування вмісту будь-якої комірки (або, якщо це потрібно, будь-якої частини формули) у форматі тексту, коду LaTeX або специфікації MathML простим клацанням правою кнопкою миші. Насправді, можна експортувати увесь робочий аркуш у форматі HTML або коду LaTeX. Із документацією до wxMaxima, зокрема робочими книгами для ілюстрації аспектів користування, можна ознайомитися на сайті довідки wxMaxima, а також за допомогою меню «Довідка».

Обчислення, які ви наказуєте виконати wxMaxima у фоновому режимі передаються інструментам командного рядка Maxima.

## 1.2 Основи роботи із книгою

Призначення більшої частини інтерфейсу wxMaxima є очевидним, але деякі речі потребують більшої уваги. На цьому сайті можна знайти декілька робочих книг, які присвячено різноманітним аспектами роботи у wxMaxima. Опрацювання деяких з цих зразків (зокрема «Десятихвилинного підручника з (wx)Maxima») надасть змогу краще ознайомитися із роботою Maxima і використанням wxMaxima для взаємодії із Maxima. У цьому ж підручнику ми зосередимо увагу на описі тих аспектів wxMaxima, які, ймовірно, є неочевидними і про які немає відомостей у інтернет-підручнику.

### 1.2.1 Використання робочих книг

Однією із дуже небагатьох речей, які є нестандартними у wxMaxima є те, що дані для Maxima у програмі впорядковано за комірками, які обчислюються (тобто

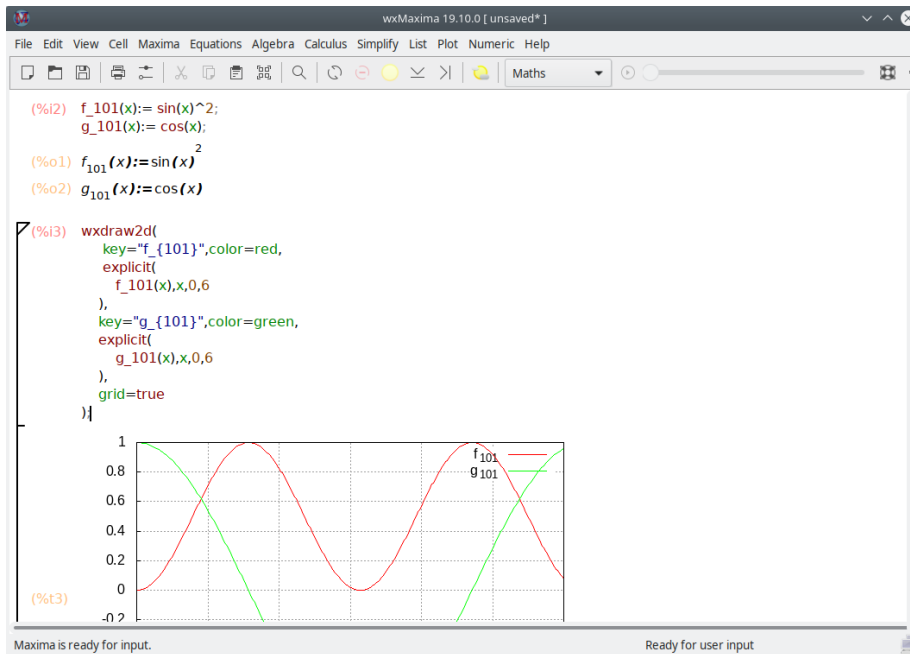


Figure 3: Вікно wxMaxima

надсилаються до Махіма) лише на запит користувача. При обчисленні комірки усі команди у ній, і лише у ній, обчислюються пакетно. (Попереднє твердження є не зовсім точним: можна позначити набір сусідніх комірок і обчислити їх разом. Крім того, можна наказати Махіма обчислити усі комірки у робочій книзі одразу.) Підхід wxMaxima до надсилання команд на виконання може спочатку видатися незвичним. Втім, він значно спрощує роботу із великими документами (у яких користувач не хоче, щоб програма автоматично вмикала повне переобчислення усього документа у відповідь на будь-яку зміну). Крім того, такий підхід є дуже зручним для усування вад.

Якщо ви почнете вводити якийсь текст у вікні wxMaxima, програма автоматично створить нову комірку робочого аркуша. Тип цієї комірки можна вибрати за допомогою панелі інструментів. Якщо створено комірку з кодом, її вміст може бути надіслано до Махіма. Результат обчислень буде показано під рядком коду. Пару таких команд показано нижче.

При обчисленні вмісту комірки вхідних даних Махіма призначає мітку для вхідних даних (типово, таку мітку буде показано червоним кольором, вона матиме формат %i). За цією міткою можна посилатися на вміст далі у сеансі роботи з wxMaxima. Виведені Махіма дані також отримують мітку, яка починається з %, і яку типово буде приховано, хіба що користувач надасть назву виведеними даним. Якщо надано назву, буде показано визначену користувачем мітку. Втім, можна буде

```
(%i2) f_101(x):= sin(x)^2;
      g_101(x):= cos(x);

(%o1) f_101(x):= sin(x)^2
(%o2) g_101(x):= cos(x)
```

Figure 4: Комірка введення-виведення

користуватися і автоматично створеною Maxima міткою у стилі %.

Окрім комірок для вхідних даних, у wxMaxima передбачено комірки для документації, комірки зображень, комірки заголовків, комірки глав та комірки розділів. У кожній комірки є власний буфер скасування дій, тому дуже просто діагностувати проблеми, змінюючи значення у комірках і поступово скасовуючи внесені зміни у різних комірках. Крім того, у самого робочого аркуша також є власний буфер скасування дій, яким може скористатися для скасування редагувань у комірках, додавання і вилучення комірок.

На рисунку нижче показано різні типи комірок (комірки заголовків, комірки розділів, комірки підрозділів, текстові комірки, комірки введення-виведення та комірки зображення).

### 1.2.2 Комірки

Робочий аркуш складається з комірок. У кожній комірці можуть міститися інші комірки або дані таких типів:

- один або декілька рядків вхідних даних Maxima
- одне або декілька зображень
- виведені дані або питання від Maxima
- текстовий блок, який може бути, наприклад, документацією
- заголовок, розділ або підрозділ.

Типовою поведінкою wxMaxima при введенні тексту є автоматичне створення комірки формули. Комірки інших типів можна створити за допомогою меню «Комірка», натискання клавіатурних скорочень для пунктів з цього меню або за допомогою спадного списку на панелі інструментів. Після створення нематематичної комірки усе, що введено у файл, вважатиметься текстом.

До математичної комірки може бути введено текст додаткового коментаря у відповідних обмежувачах: /\*Цей коментар не буде надіслано до Maxima для обробки\*/.

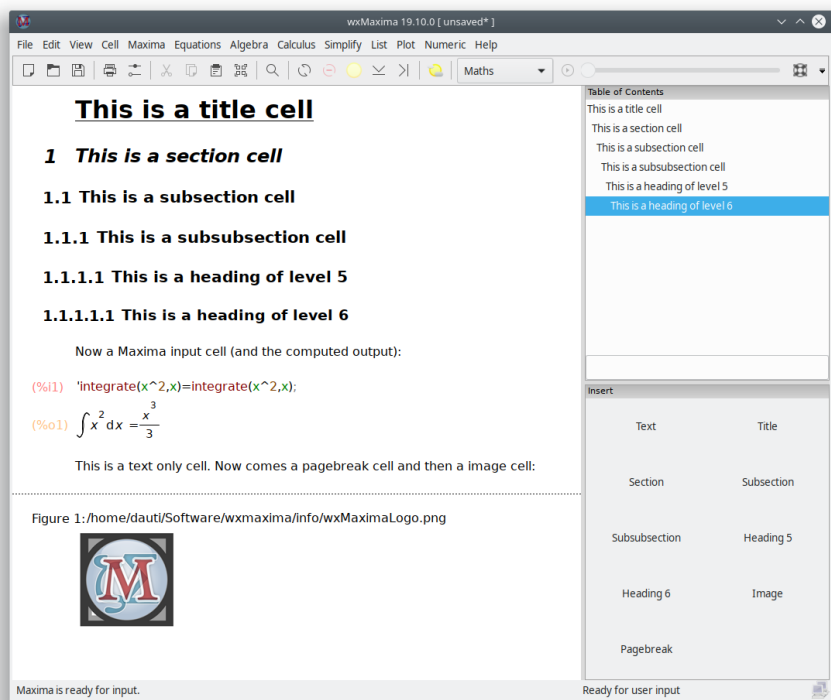


Figure 5: Приклад різних комірок wxMaxima



### 1.2.3 Горизонтальний і вертикальний курсори

Якщо користувач намагається позначити ціле речення, текстовий процесор намагається розширити позначений фрагмент так, щоб він автоматично завершувався на початку і наприкінці слів. Так само, wxMaxima, якщо позначено декілька комірок, доповнюватиме позначений фрагмент до меж комірок.

Нестандартною особливістю є те, що у wxMaxima передбачено гнучку систему перетягування зі скиданням, у якій визначено два типи курсорів. wxMaxima автоматично перемикатиметься між ними, коли у цьому виникатиме потреба:

- Курсор буде намальовано горизонтально, якщо його пересунуто до проміжку між двома комірками, або якщо користувач клацнув кнопкою миші у проміжку між двома комірками.
- Вертикальний курсор, який працює у комірці. Цей курсор активується пересуванням курсора всередину комірки за допомогою вказівника миші або клавіш зі стрілками. Цей курсор працює подібно до курсора у текстовому редакторі.

### 1.2.4 Надсилання комірок для обчислення у Maxima

Команду у комірці буде виконано одразу після натискання комбінацій клавіш CTRL+ENTER, SHIFT+ENTER або клавіші ENTER на клавіатурі. Типовим для wxMaxima є введення команд за допомогою CTRL+ENTER або SHIFT+ENTER, але wxMaxima можна налаштувати на виконання команд у відповідь на натискання ENTER.

### 1.2.5 Автоматичне доповнення команд

У wxMaxima реалізовано можливість автоматичного доповнення коду, якою можна скористатися за допомогою меню (Комірка/Доповнити слово) або натисканням комбінації клавіш CTRL+Пробіл. Автоматичне доповнення є чутливим до регістру символів. Наприклад, якщо його активувати зі специфікацією модулів для ezUnits, воно запропонує список відповідних модулів.

Окрім доповнення, назви файла, назви модуля або назви поточної команди чи змінної, засіб автоматичного доповнення може показувати шаблони для більшості команд, вказуючи тип (і значення) параметрів, які очікує отримати відповідна програма. Щоб задіяти цю можливість, натисніть комбінацію клавіш SHIFT+CTRL+ПРОБІЛ або виберіть відповідний пункт меню (Комірка/Показати шаблон).

#### 1.2.5.1 Грецькі літери

Комп'ютери традиційно зберігають символи у форматі 8-бітових значень. Це уможливорює існування наборів, що складаються лише із 256 різних символів. У такий набір може бути включено усі літери, цифри та символи керування (завершення передавання, кінець рядка, лінії та кутики для малювання меню тощо) майже будь-якої європейської мови.

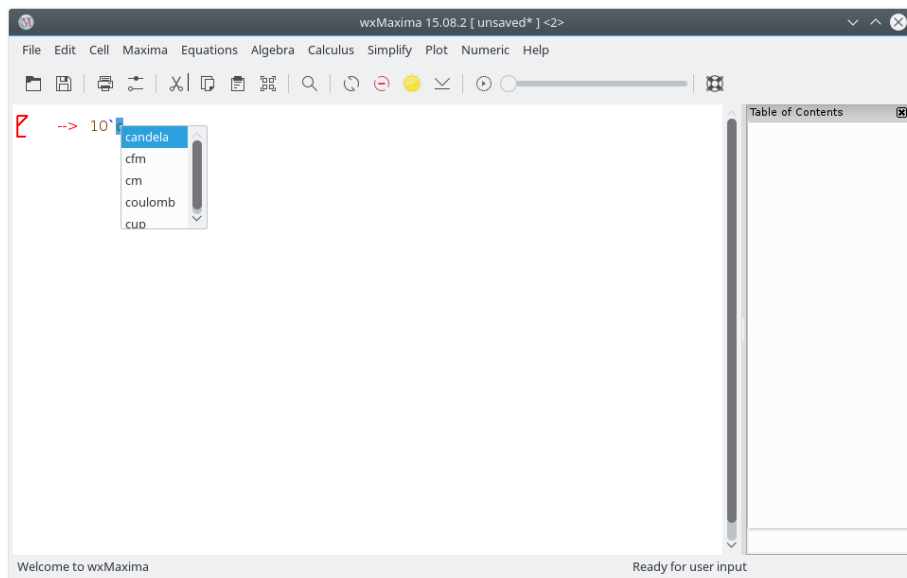


Figure 6: ezUnits

Втім, у більшості мов до вибраної кодової сторінки із 256 символів не можна включити додаткові символи, зокрема літери грецької абетки, які широко використовуються у математичних формулах. Щоб подолати обмеження подібного типу, було винайдено Юнікод — кодування, яке працює для латиниці, але використовує набагато більше за 256 символів.

Maxima уможливило використання символів Юнікоду, якщо програму було зібрано за допомогою компілятора Lisp, у якому передбачено підтримку Юнікоду або такого, який є нейтральним щодо кодування символів. Оскільки, хоча б одна з цих вимог, зазвичай, справджується, у wxMaxima реалізовано спосіб введення грецьких символів з клавіатури:

- Грецьку літеру можна ввести натисканням клавіші ESC з наступним введенням назви символу грецької абетки.
- Крім того, можна натиснути ESC, ввести одну літеру (чи дві літери для грецької літери «омікрон») і натиснути ESC ще раз. У цьому випадку прив'язка до літер є такою:

key	greek letter	key	greek letter	key	greek letter
a	альфа	i	йота	r	ро
b	бета	k	каша	s	сигма
g	гамма	l	лямбда	t	тау
d	дельта	m	мю	u	упсилон
e	епсилон	n	ню	f	фі
z	дзета	x	ксі	c	хі

key	greek letter	key	greek letter	key	greek letter
h	ета	om	омікрон	y	псі
q	тета	p	пі	o	омега
A	Альфа	I	Йота	R	Ро
B	Бета	K	Каппа	S	Сигма
G	Гамма	L	Лямбда	T	Тау
D	Дельта	M	Мю	U	Упсилон
E	Епсилон	N	Ню	P	Фі
Z	Дзета	X	Ксі	C	Хі
H	Ета	Om	Омікрон	Y	Псі
T	Тета	P	Пі	O	Омега

У той самий спосіб можна вводити різноманітні математичні символи:

введений рядок	математичний символ
hbar	стала Планка: $\hbar$ із горизонтальною рисою згори
Hbar	$\bar{H}$ із горизонтальною рисою згори
2	квадрат
3	куб
/2	1/2
partial	знак частинної похідної ( $d y dx/dt$ )
integral	символ інтеграла
sq	квадратний корінь
ii	уявна одиниця
ee	елемент
in	включається
impl implies	впливає
inf	нескінченність
empty	порожня множина
TB	великий трикутник праворуч
tb	малий трикутник праворуч
and	і
or	або
xor	виключне або
nand	ні-і
nor	ні-або
equiv	еквівалентний
not	ні
union	об'єднання
inter	перетин
subseteq	підмножина або рівна множина
subset	підмножина
notsubseteq	не підмножина або рівна множина
notsubset	не підмножина

введений рядок	математичний символ
approx	приблизно
propto	пропорційне до
neq != /= або #	не рівне
+/- або pm	плюс-мінус
<= або leq	менше або дорівнює
>= або geq	більше або дорівнює
<< або ll	набагато менше
>> або gg	набагато більше
qed	що і слід було довести
nabla	набла-оператор
sum	знак суми
prod	знак добутку
exists	знак «існує»
nexists	знак «не існує»
parallel	знак «паралельно»
perp	знак «перпендикулярно»
leadsto	знак «впливає»
->	стрілка праворуч
->	довга стрілка праворуч

Якщо спеціального символу немає у списку, ви можете ввести його як довільний символ Юнікод натисканням послідовності клавіш ESC [номер символу (шістнадцятковий)] ESC.

Отже, ESC 61 ESC дасть «а».

Будь ласка, зауважте, що більшість цих символів (виключеннями є логічні символи) не мають спеціального значення у Maxima і тому можуть бути оброблені як звичайні символи. Якщо Maxima зібрано із Lisp, у якому не передбачено обробки символів Юнікоду, використання цих символів може призвести до показу повідомлення про помилку.

### 1.2.6 Бічні панелі

Доступ до найважливіших команд Maxima або речей, подібних до таблиці змісту, вікон із діагностичними повідомленнями або журналу останніх виданих команд можна отримати за допомогою бічних панелей. Увімкнути бічні панелі можна за допомогою меню «Перегляд». Панелі можна пересувати як всередині вікна wxMaxima, так і ззовні. Ще однією корисною панеллю є панель введення грецьких літер за допомогою вказівника миші.

### 1.2.7 Виведення коду MathML

Деякі текстові процесори і подібні програми або розпізнають вхідні дані MathML і автоматично вставляють їх до придатного для редагування рівняння, або (подібно

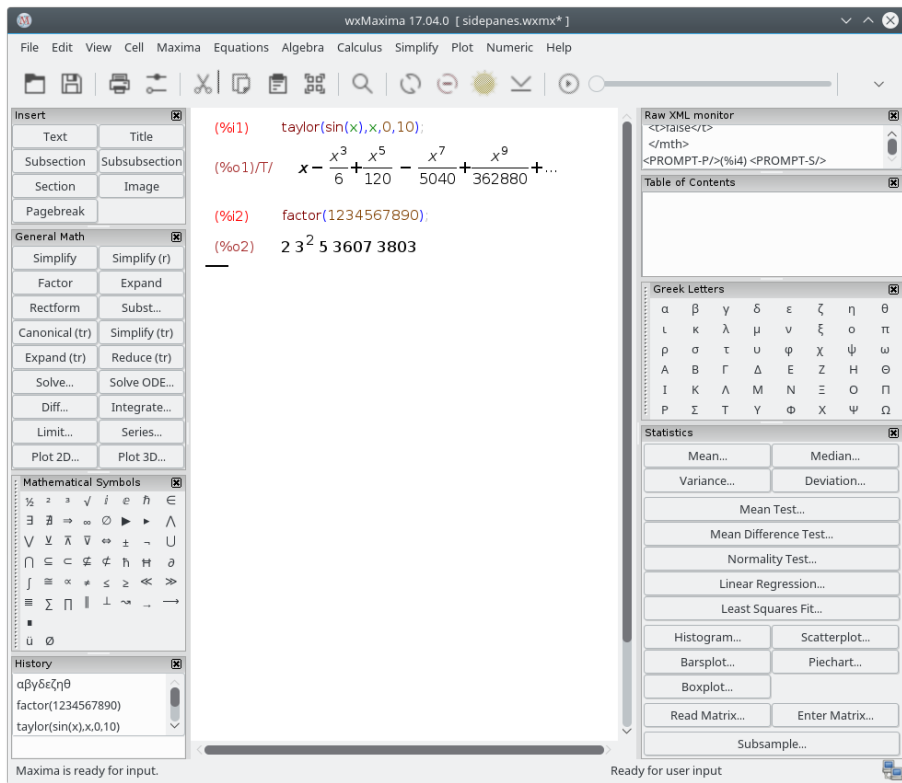


Figure 7: Приклад різних бічних панелей

до LibreOffice 5.1) мають у складі редактор рівнянь, у якому передбачено можливість імпортування даних MathML з буфера обміну даними. У інших редакторах передбачено підтримку формул у RTE. Через це у wxMaxima передбачено декілька відповідних пунктів у контекстному меню.

### 1.2.8 Підтримка Markdown

У wxMaxima ви зможете скористатися стандартними варіантами розмітки, які не суперечать математичним позначенням. Одним із елементів розмітки є невпорядковані списки.

Звичайний текст

- \* Один пункт, рівень відступу 1
- \* Інший пункт на рівні відступу 1
  - \* Пункт на другому рівні відступу
  - \* Другий пункт на другому рівні відступу
- \* Третій пункт на першому рівні відступу

Звичайний текст

У wxMaxima фрагменти тексту, які розпочинаються з символів > вважаються блоками цитат:

Звичайний текст

```
> цитата цитата цитата цитата
> цитата цитата цитата цитата
> цитата цитата цитата цитата
```

Звичайний текст

У коді TeX та HTML wxMaxima також розпізнається рядок =>, який програма замінює на відповідний символ Юнікод:

```
cogito => sum.
```

Іншими символами, які буде розпізнано під час експортування до HTML або TeX, є <= i >= для порівнянь, двобічна стрілка (<=>), одинарні стрілки (<->, -> i <-), а також +/- як відповідний знак. Для експортування у код TeX передбачено також підтримку перетворення << i >>.

### 1.2.9 Клавіатурні скорочення

Більшість клавіатурних скорочень можна знайти у тексті відповідних меню. Оскільки їх взято з тексту меню і, отже, може бути змінено під час перекладу wxMaxima так, щоб вони відповідали потребам користувачів локалізованих клавіатур, ми не наводимо їх у цьому підручнику. Втім, декілька клавіатурних скорочень все ж не документовано у меню:

- CTRL+SHIFT+DELETE вилучає увесь вміст комірки.
- CTRL+TAB або CTRL+SHIFT+TAB перемикає механізм автоматичного доповнення.
- SHIFT+ПРОБІЛ вставляє нерозривний пробіл.

#### 1.2.10 Код TeX у експортованому TeX

Якщо текстову комірку буде розпочато з TeX: у експортованих у форматі TeX даних буде буквально відтворено текст, який вказано після позначки TeX:. За допомогою цієї можливості можна використовувати розмітку TeX у робочій книзі wxMaxima.

### 1.3 Формати файлів

Матеріал, який було отримано під час сеансу роботи у wxMaxima може бути збережено для наступного використання у три способи:

#### 1.3.1 .mac

Файли .mac є звичайними текстовими файлами, які містять команди Maxima. Їх можна прочитати за допомогою команди batch() або load() Maxima або пункту меню «Файл/Пакетний файл».

Один з прикладів наведено нижче. Quadratic.mac визначає функцію і після цього створює креслення за допомогою wxdraw2d(). Далі, буде виведено вміст файла Quadratic.mac і виконано обчислення нової визначеної функції f().

Увага! Хоча назва файла Quadratic.mac містить звичайний для Maxima суфікс (.mac), прочитати дані з нього зможе лише wxMaxima, оскільки команда wxdraw2d() є wxMaxima-розширенням до Maxima.

Ви можете використовувати файли .mac для написання власної бібліотеки макросів. Але оскільки у них не міститься достатніх даних щодо структури, їх не можна читати як сеанс wxMaxima.

#### 1.3.2 .wxm

Файли .wxm містять дані робочого аркуша, окрім виведених даних Maxima. У версіях Maxima >5.38 їх можна прочитати за допомогою функції load() Maxima, так само, як можна прочитати файли .mac. У цьому текстовому форматі іноді є неусувна проблема — робочі аркуші із новими можливостями не є зворотно сумісними із застарілими версіями wxMaxima.

#### 1.3.3 .wxmx

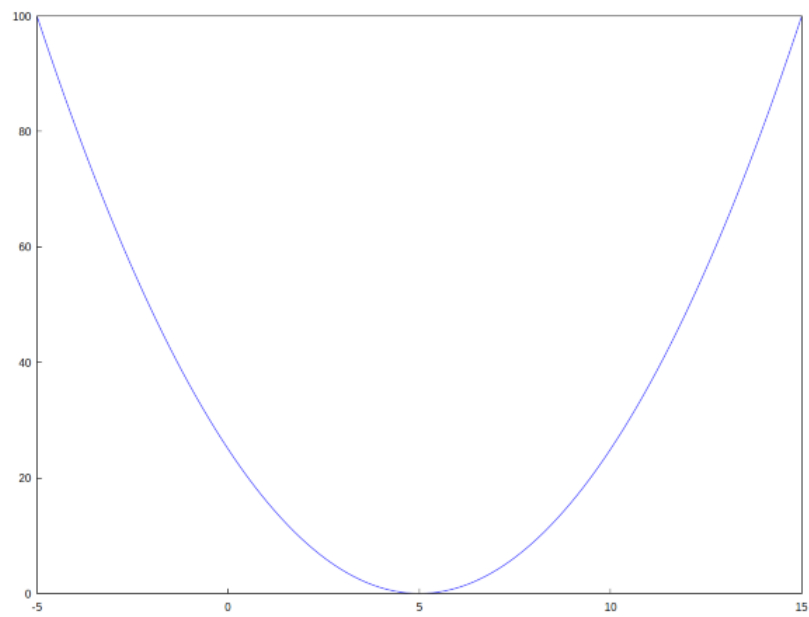
У цьому заснованому на XML форматі файлів зберігається увесь робочий аркуш, включно із даними щодо масштабу та списком спостереження. Цей формат файлів є рекомендованим.

### 1.4 Параметри налаштування

Для деяких типових змінних налаштувань у wxMaxima передбачено два способи налаштування:

```
(%i1) batch("Quadratic.mac");  
read and interpret /tmp/Quadratic.mac  
(%i2) f(x):=(x-5)^2  
(%i3) wxdraw2d(explicit(f(x),x,-5,15))
```

(%t3)



(%o4) /tmp/Quadratic.mac

```
(%i5) printfile("Quadratic.mac")$  
f(x):=(x-5)^2$  
wxdraw2d(explicit(f(x),x,-5,15))$
```

(%i6) f(7);

(%o6) 4

Figure 8: Завантаження файла .mac за допомогою batch()



- За допомогою діалогової панелі налаштування, знімок якої наведено нижче, ви можете змінити типові значення для поточного і наступних сеансі роботи програми.
- Крім того, значення більшості змінних налаштувань може бути змінено лише для поточного сеансу перезаписуванням їхніх значень з робочого аркуша, як це показано нижче.

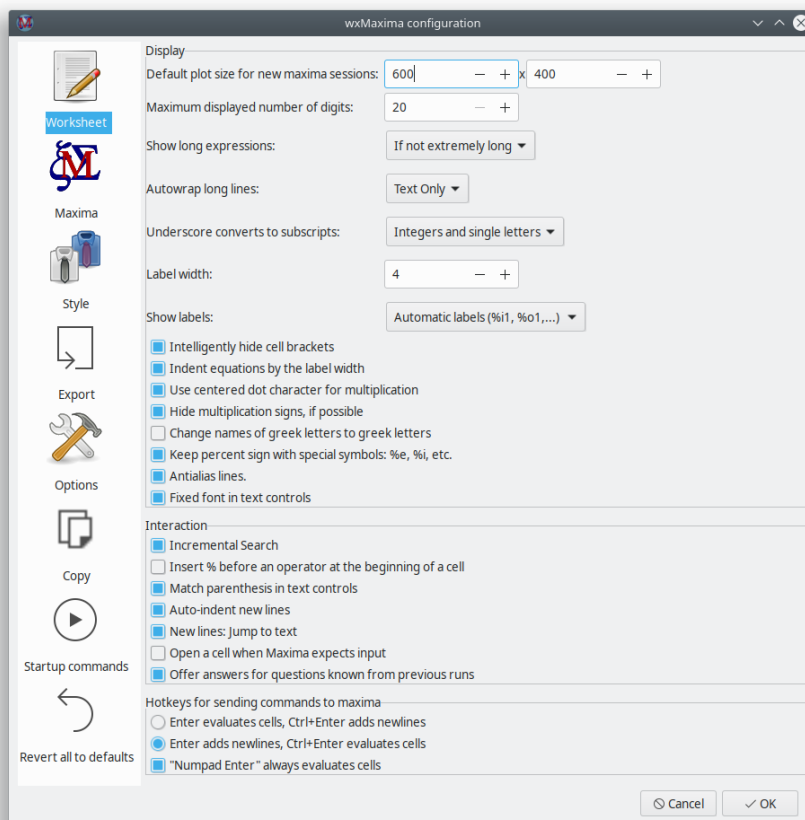


Figure 9: Налаштування wxMaxima 1

#### 1.4.1 Типова частота кадрів анімації

Частота кадрів анімації, яка використовуватиметься для нових анімацій, зберігається у змінній `wxanimate_framerate`. Початкове значення цієї змінної, яке міститиметься у новому робочому аркуші, можна змінити за допомогою діалогового вікна налаштування.

#### 1.4.2 Типовий розмір креслення для нових сеансів wxima

Після наступного запуску креслення, які вбудовано до робочого аркуша, буде створено з цим розміром, якщо значення `wxplot_size` не було змінено wxima.

Щоб встановити розмір графіка для окремого графіка, можна скористатися наведеною нижче конструкцією, яка встановить значення змінної лише для одної команди:

```
wxdraw2d(  
    explicit(  
        x^2,  
        x, -5,5  
    )  
) , wxplot_size=[480,480]$
```

#### 1.4.3 Перевіряти баланс дужок у тексті

За допомогою цього пункту можна увімкнути дві речі:

- Якщо ви введете у wxMaxima початкову круглу або квадратну дужку або подвійні лапки, програма автоматично вставить завершальну дужку або лапки за нею.
- Якщо при натисканні якоїсь із цих комбінацій клавіш було позначено фрагмент тексту, цей текст буде вставлено між відповідними символами.

#### 1.4.4 Не зберігати робочий аркуш автоматично

Якщо позначено цей пункт, файл, у якому зберігається робочий аркуш, буде перезаписано лише за запит користувача. Втім, у випадку аварії, втрати живлення тощо, у каталозі тимчасових даних завжди лишатиметься резервна копія.

Якщо не встановлено цей параметр, wxMaxima поводитиметься подібно до сучасної програми для мобільних телефонів:

- Файли автоматично зберігаються під час виходу з програми
- І файл буде автоматично збережено кожні 3 хвилини.

#### 1.4.5 Де зберігаються налаштування?

Якщо ви користуєтеся Unix/Linux, дані налаштувань буде збережено до файла `.wxMaxima` у вашому домашньому каталозі (якщо ви користуєтеся wxWidgets < 3.1.1), або `.config/wxMaxima.conf` ((стандартний каталог XDG), якщо використано wxWidgets >= 3.1.1). Визначити версію wxWidgets можна за допомогою команди `wxbuild_info()`; або пункту меню «Довідка -> Про програму». wxWidgets є бібліотекою графічного інтерфейсу користувача, яка працює на багатьох платформах і є основою wxMaxima (звідси wx у назві програми). (Оскільки назви файлів починається з крапки, файл `.wxMaxima` і каталог `.config` є прихованими).

Якщо ви користуєтеся Windows, налаштування буде збережено у реєстрі. Записи wxMaxima можна знайти у цій гілці: HKEY\_CURRENT\_USER\Software\wxMaxima

---

## 2 Розширення Maxima

Основним призначенням wxMaxima є забезпечення графічного інтерфейсу до Maxima. Отже, основним завданням програми є передавання команд до Maxima і сповіщення користувача про результат виконання цих команд. Втім, у деяких випадках, wxMaxima розширює функціональні можливості Maxima. Ми вже згадували про здатність wxMaxima створювати звіти експортуванням вмісту робочої книги до файлів HTML і LaTeX. У цьому розділі ми розглянемо деякі способи, у які wxMaxima удосконалює включення графіки до сеансу.

### 2.1 Змінні із індексами

якщо встановлено значення true для wxsubscripts, назви змінних у форматі x<sub>y</sub> буде показано як літеру із нижнім індексом, якщо

- y є одинарною літерою
- y є цілим числом

Якщо назва змінної не задовольняє ці вимоги, змінну все одно можна оголосити як змінну із нижнім індексом, за допомогою команди wxdeclare\_subscript(назва\_змінної); або wxdeclare\_subscript([назва\_змінної1,назва\_змінної2,...]); Оголошення змінної як змінної із нижнім індексом можна скасувати за допомогою такої команди: wxdeclare\_subscript(назва\_змінної,false);

### 2.2 Візуальний супровід дій користувача у смужці стану

Команди, які виконуються довго, можуть надавати користувачам інформацію на смужці стану. Цю інформацію буде замінено на будь-яку нову інформацію, яку програма надсилатиме туди (це надає змогу використовувати смужку стану як індикатор поступу), і буде вилучено, щойно буде завершено виконання поточної команди, яку надіслано до Maxima. Команду wxstatusbar() можна безпечно використовувати навіть у бібліотеках, які може бути використано у звичайній Maxima (без wxMaxima): якщо wxMaxima немає, команда wxstatusbar() просто не оброблятиметься.

```
for i:1 thru 10 do (  
  /* Повідомляємо користувачу, наскільки далеко зайшли */  
  wxstatusbar(concat("Pass ",i)),  
  /* (sleep n) — функція Lisp, якою можна скористатися */  
  /* із додаванням перед нею символу «?». Команда затримує */  
  /* виконання програми (у нашому прикладі, на 3 секунди) */
```

```
?sleep(3)
)$
```

## 2.3 Креслення

Креслення (яке фундаментально пов'язано із графікою) є місцем, де графічний інтерфейс користувача має надавати якісь додаткові можливості до оригінальної програми.

### 2.3.1 Вбудовування креслення до робочого аркуша

Зазвичай, Maxima наказує зовнішній програмі gnuplot відкривати окреме вікно для кожної створеної програмою діаграми. Оскільки набагато зручніше вбудувати графіку безпосередньо до робочого аркуша, у wxMaxima передбачено власний набір команд для креслення, які не відрізняються від відповідних функцій maxima майже нічим, окрім назви: в усіх цих командах використано префікс «wx». Наприклад, команда wxplot2d відповідає команді plot2d, команда wxplot3d — команді plot3d, команда wxdraw — команді draw, а команда wxhistogram — команді histogram.

### 2.3.2 Збільшення або зменшення вбудованих креслень

Як ми вже згадували вище, діалогове вікно налаштування надає змогу змінити типові розміри креслень. Унаслідок застосування цього способу змінюється початкове значення wxplot\_size. Підпрограми створення креслень wxMaxima враховують це значення, яке визначає розмір креслення у пікселях. Це значення завжди можна отримати або використати для встановлення розмірів наступних креслень:

```
wxplot_size:[1200,800]$
wxdraw2d(
  explicit(
    sin(x),
    x,1,10
  )
)$
```

Якщо потрібно змінити розміри лише одного креслення, у Maxima передбачено канонічний спосіб зміни атрибута лише для поточної комірки. У цьому випадку до команди wxdraw2d( ) додається специфікація wxplot\_size = [значення1, значення2], яка не є частиною команди wxdraw2d.

```
wxdraw2d(
  explicit(
    sin(x),
    x,1,10
  )
),wxplot_size=[1600,800]$
```

### 2.3.3 Креслення кращої якості

Gnuplot, здається, не надає придатного до портування на різні платформи способу визначення, чи передбачено у програмі підтримку виведення даних у форматі високоякісного растрового зображення з використанням бібліотеки `cairo`. У системах, де `gnuplot` зібрано з використанням відповідної бібліотеки, пункт `pngcairo` у меню налаштування (який може бути перевизначено за допомогою змінної `wxplot_pngcairo`) вмикає підтримку згладжування та додаткових стилів ліній. Якщо встановлено `wxplot_pngcairo` без підтримки у `gnuplot`, результатом буде повідомлення про помилку замість графіка.

### 2.3.4 Відкриття вбудованих креслень в інтерактивних вікнах `gnuplot`

Якщо креслення було створено за допомогою команд типу `wxdraw` (у `wxplot2d` і `wxplot3d` підтримки цієї можливості не передбачено), і розмір файла базового проекту `gnuplot` не є надто великим, `wxMaxima` пропонує контекстне меню, за допомогою якого можна відкрити креслення у інтерактивному вікні `gnuplot`.

### 2.3.5 Відкриття консолі команд `gnuplot` у вікнах `plot`

У MS Windows, якщо у змінній `Maxima gnuplot_command` замінити «`gnuplot`» на «`wgnuplot`», `gnuplot` запропонує можливість відкрити вікно консолі, у якому ви зможете вводити команди `gnuplot`. На жаль, вмикання цієї можливості призводить до того, що `gnuplot` на короткий період «викрадає» фокусування клавіатури кожного разу, коли готується креслення.

### 2.3.6 Вбудовування анімацій до робочого аркуша

На просторових кресленнях доволі важко читати кількісні дані. Альтернативним рішенням є прив'язка третього параметра до коліщатка миші. Команда `with_slider_draw` є версією `wxdraw2d`, яка готує декілька креслень і надає змогу перемикатися між ними пересуванням повзунка у верхній частині вікна. `wxMaxima` надає змогу експортувати анімацію у форматі анімації `gif`.

Першими двома аргументами `with_slider_draw` є назва змінної, яка покроково змінюватиметься між кресленнями, і список значень цієї змінної. Наступними аргументами є звичайні аргументи `wxdraw2d`:

```
with_slider_draw(  
  f, [1,2,3,4,5,6,7,10],  
  title=concat("f=", f, "Hz"),  
  explicit(  
    sin(2*pi*f*x),  
    x, 0, 1  
  ), grid=true  
);
```

Доступ до тих самих можливостей для просторових креслень можна отримати за

допомогою функції `with_slider_draw3d`, яка надає можливість обертати просторові креслення:

```
wxanimate_autoplay:true;
wxanimate_framerate:20;
with_slider_draw3d(
  a,makelist(i,i,1,360,3),
  title=sconcat("a=",a),
  surface_hide=true,
  contour=both,
  view=[60,a],
  explicit(
    sin(x)*sin(y),
    x,-π,π,
    y,-π,π
  )
)$
```

Якщо важливою є загальна форма креслення, може бути достатнім трошки пересунути креслення з метою зробити його просторову природу інтуїтивно зрозумілою:

```
wxanimate_autoplay:true;
wxanimate_framerate:20;
with_slider_draw3d(
  t,makelist(i,i,0,2*π,.05*π),
  title=sconcat("a=",a),
  surface_hide=true,
  contour=both,
  view=[60,30+5*sin(t)],
  explicit(
    sin(x)*y^2,
    x,-2*π,2*π,
    y,-2*π,2*π
  )
)$
```

Для тих, хто знайомий більше з `plot`, аніж з `draw`, другий набір функцій:

- `with_slider i`
- `wxanimate`.

Зазвичай, анімації відтворюються і експортуються із частотою кадрів, яку можна вибрати у налаштуваннях `wxMaxima`. Щоб встановити швидкість відтворення окремої анімації, можна скористатися змінною `wxanimate_framerate`:

```
wxanimate(a, 10,
  sin(a*x), [x,-5,5]), wxanimate_framerate=6$
```

Функції анімації використовують команду `maxima makelist` і тому мають спільну

ваду — значення змінної повзунка замінюється на вираз, лише якщо змінна є безпосередньо видимою у виразі. Тому такий приклад не працюватиме:

```
f:sin(a*x);
with_slider_draw(
  a,makelist(i/2,i,1,10),
  title=concat("a=",float(a)),
  grid=true,
  explicit(f,x,0,10)
)$
```

А якщо Махіма явно попросити підставити значення повзунка, креслення працюватиме як слід:

```
f:sin(a*x);
with_slider_draw(
  b,makelist(i/2,i,1,10),
  title=concat("a=",float(b)),
  grid=true,
  explicit(
    subst(a=b,f),
    x,0,10
  )
)$
```

### 2.3.7 Відкриття декількох креслень у вікнах одночасно

Хоча wxMaхіма не надає доступу до цієї можливості, ця функція Махіма (у конфігураціях, де передбачено її підтримку) іноді є дуже зручною. Наведений нижче приклад походить із допису Маріо Родрігеса (Mario Rodriguez) у списку листування Махіма:

```
load(draw);

/* Парабола у вікні 1 */
draw2d(terminal=[wxt,1],explicit(x^2,x,-1,1));

/* Парабола у вікні 2 */
draw2d(terminal=[wxt,2],explicit(x^2,x,-1,1));

/* Параболоїд у вікні 3 */
draw3d(terminal=[wxt,3],explicit(x^2+y^2,x,-1,1,y,-1,1));
```

Також можна накреслити декілька креслень в одному вікні:

```
wxdraw(
  gr2d(
    key="sin (x)",grid=[2,2],
    explicit(sin(x),x,0,2*pi)),
```

```

gr2d(
key="cos (x)",grid=[2,2],
explicit(cos(x),x,0,2*pi))
);

```

### 2.3.8 Бічна панель «Накреслити за допомогою Draw»

Бічна панель «Накреслити за допомогою Draw» приховує простий генератор коду, який уможливило створення сцен, які використовують певну гнучкість пакунка draw, з яким постачається maxima.

#### 2.3.8.1 Двовимірний

Створює каркас команди draw(), яка малює плоску сцену. Цю сцену пізніше може бути заповнено за допомогою команд, які створюють вміст сцени, наприклад, за допомогою кнопок у рядках під кнопкою «Плоский».

Однією з корисних можливостей кнопки «Плоский» є те, що вона надає змогу налаштувати сцену як анімацію, у якій змінна (типовою змінною є t) має різне значення для кожного з кадрів: часто динамічне плоске креслення надає змогу простіше інтерпретувати дані за ті самі у формі нерухомого просторового креслення.

#### 2.3.8.2 Тривимірний

Створює каркас команди draw(), яка малює просторову сцену. Якщо не налаштовано ні двовимірну, ні тривимірну сцену, усі інші кнопки налаштуватимуть двовимірну сцену, яка містить команду, яку створює кнопка.

#### 2.3.8.3 Вираз

Додає стандартне креслення виразу, подібного до  $\sin(x)$ ,  $x \cdot \sin(x)$  або  $x^2 + 2x - 4$  до команди draw(), у якій зараз перебуває курсор. Якщо команди draw немає, буде створено двовимірну сцену із кресленням. Кожну сцену можна заповнити довільною кількістю креслень.

#### 2.3.8.4 Графік неявної функції

Спробувати знайти усі точки, у яких справджуються вирази, подібні до  $y = \sin(x)$ ,  $y \cdot \sin(x) = 3$  або  $x^2 + y^2 = 4$ , і накреслити криву-результат за допомогою команди draw() у поточній позиції курсора. Якщо команди draw виявлено не буде, буде створено плоску сцену із кресленням.

#### 2.3.8.5 Графік параметричної функції

Покроково проходить змінну від нижньої межі до верхньої межі і використовує два виразу, подібні до  $t \cdot \sin(t)$  і  $t \cdot \cos(t)$  для створення координат точок x, y (у просторових кресленнях використовується і z) кривої, які буде передано до поточної команди draw.



#### 2.3.8.6 Точки

Малює точки, які, якщо вказано, буде з'єднано. Координати точок буде взято зі списку списків, двовимірного масиву або списку масивів для кожної з осей.

#### 2.3.8.7 Заголовок діаграми

Малює заголовок у верхній частині діаграми,

#### 2.3.8.8 Вісь

Налаштовує вісі.

#### 2.3.8.9 Контур

(Лише для просторових креслень) Додає контурні лінії, подібні до тих, які можна знайти на карті висот, до команд креслення, які йдуть за нею у поточній команді «draw()» і/або до площини основи діаграми. Крім того, цей майстер надає змогу повністю пропустити малювання кривих і показати лише контурне креслення.

#### 2.3.8.10 Назва графіка

Додає запис умовних позначень із назвою наступного креслення на панель умовних позначень діаграми. Порожнє значення назви вимикає створення записів умовних позначень для наступних креслень.

#### 2.3.8.11 Колір лінії

Встановлює колір лінії для наступних креслень у поточній команді малювання.

#### 2.3.8.12 Колір заповнення

Встановлює колір заповнення для наступних креслень у поточній команді малювання.

#### 2.3.8.13 Сітка

Відкриває допоміжне вікно, за допомогою якого можна налаштувати лінії сітки.

#### 2.3.8.14 Точність

Надає змогу вибрати адекватну позицію у компромісному питанні «швидкість чи точність?», яке є частиною будь-якої програми для креслення.

## 2.4 Вбудовування графіків

Якщо використано формат файлів `.wxm`, вбудовування файлів до проекту `wxMaxima` можна виконати простим перетягуванням зі скиданням. Але іноді (наприклад, якщо вміст зображення може бути змінено згодом протягом роботи у сеансі), краще наказати програмі завантажувати файл зображення при обробці команд:

```
show_image("man.png");
```

## 2.5 Файли запуску

У діалоговому вікні налаштування `wxMaxima` передбачено можливість редагування двох файлів із командами, які виконуються під час запуску програми:

- Файла, у якому містяться команди, які виконуються під час запуску `Maxima`: `maxima-init.mac`
- одного файла із додатковими командами, які виконуються, якщо `wxMaxima` запускає `Maxima`: `wxmaxima-init.mac`

Ці файли зберігаються у каталозі користувача `Maxima` (зазвичай, каталозі `maxima` у `Windows`, `.maxima/` у інших системах) у домашній теці користувача. Розташування файлів можна визначити за допомогою такої команди: `maxima_userdir`;

## 2.6 Спеціальні змінні `wx...`

- `wxsubscripts` вказує `Maxima`, чи слід перетворювати назви змінних, які містять символ підкреслювання (`R_150` або подібні змінні) на змінні із нижніми індексами. Див. `wxdeclare_subscript`, щоб дізнатися більше про те, які саме назви змінних буде перетворено автоматично.
- `wxfilename`: у цій змінній зберігається назва файла, який зараз відкрито у `wxMaxima`.
- `wxplot_pngcairo` визначає те, чи намагатиметься `wxMaxima` скористатися терміналом `pngcairo gnuplot`, який надає доступ до ширшого вибору стилів ліній та кращої якості зображення.
- `wxplot_size` визначає роздільну здатність для вбудованих креслень.
- `wxchangedir`: у більшості операційних систем `wxMaxima` автоматично встановлює для робочого каталогу `Maxima` каталог поточного файла. Це уможливує роботу введення-виведення до файла (наприклад, за допомогою `read_matrix`) без визначення усього шляху до файла, який має бути прочитано або записано. У `Windows` ця можливість іноді спричиняє повідомлення про помилку, тому варто встановити для цього параметра значення `false` у діалоговому вікні налаштування.
- `wxanimate_framerate`: частота у кадрах на секунду, із якою відтворюватимуться наступні анімації.
- `wxanimate_autoplay`: типово автоматично відтворювати анімації?

## 2.7 Структурний друк двовимірних даних

Функція `table_form()` показує двовимірний список у формі, яка є зручнішою для читання, за виведені дані типової підпрограми виведення даних `Maxima`. Вхідними даними є список одного або декількох списків. Подібно до команди друку, ця команда виводить дані, навіть якщо її завершити символом долара. Завершення команди символом крапки з комою призводить до виведення тієї самої таблиці разом із інструкцією «done».

```
table_form(  
  [  
    [1,2],  
    [3,4]  
  ]  
)$
```

Як показує наступний приклад, списки, які зібрано командою `table_form`, можна створювати до виконання команди.

```
(%i9)      titleList:["1st value", "2nd value", "3rd value"];  
          xList : makelist(x,x,1,3);  
          xsqList : makelist(x^2,x,1,3);  
          table_form([ titleList,xList,xsqList ])$  
  
(titleList) [1st value,2nd value,3rd value]  
(titleList) [1,2,3]  
(titleList) [1,4,9]  
  
          1st value 2nd value 3rd value  
(titleList) 1      2      3  
              1      4      9
```

Figure 10: Третій приклад таблиці

Крім того, оскільки матриця — список списків, матриці можна перетворювати на таблиці у подібний же спосіб.

## 2.8 Звітування про вади

У `wxMaxima` передбачено декілька функцій, які збирають дані щодо поточної системи для звітування щодо вад:

- `wxbuild_info()` збирає відомості щодо поточної запущеної версії `wxMaxima`
- `wxbug_report()` надає відомості щодо того, як і де слід повідомляти про вади

```
(%i17) M : matrix(titleList,xList,xsqList);
      table_form(M)$
```

	<i>1st value</i>	<i>2nd value</i>	<i>3rd value</i>
(M)	1	2	3
	1	4	9

```
(%t17)      1st value  2nd value  3rd value
           1           2           3
           1           4           9
```

Figure 11: Ще один приклад table\_form

2.9 Як зробити так, щоб виведені дані було показано червоним кольором?

Команда `box()` Махіма призводить до того, що `wxМахіма` виводить її аргумент червоним кольором.

### 3 Вирішення проблем

#### 3.1 Не вдається з'єднатися із Махіма

Оскільки Махіма (програма, яка виконує самі обчислення) і `wxМахіма` (програма, яка надає простий у користуванні інтерфейс користувача) є окремими програмами, які обмінюються даними за допомогою з'єднання локальною мережею. Тому найімовірнішою причиною є те, що це з'єднання чомусь не працює. Наприклад, брандмауер може бути налаштовано таким чином, що він не просто запобігає неуповноваженим з'єднанням з інтернету (і, можливо, також перехоплює з'єднання з інтернетом), але і блокує обмін даними між процесами на тому самому комп'ютері. Зауважте, що оскільки Махіма виконується процесором Lisp, процес. із яким блокується обмін даними, не обов'язково називається «махіма». Типовими назвами програм, які відкривають з'єднання мережі, можуть бути `sbcl`, `gcl`, `ccl`, `lisp.exe` або щось подібне.

У системах Un\*x іншою можливою причиною є те, що петльову мережу, яка забезпечує з'єднання мережі між двома програмами, які працюють на одному комп'ютері, не налаштовано належним чином.

### 3.2 Як зберегти дані із пошкодженого файла .wxmx

На внутрішньому рівні більшість сучасних форматів на основі XML є звичайними файлами zip. wxMaxima не увімкнула стискання, що робить вміст файлів .wxmx придатним до перекладу у будь-якому текстовому редакторі.

Якщо підпис zip наприкінці файла є неушкодженим після перейменування пошкодженого файла .wxmx на .zip, у більшості операційних систем передбачено спосіб видобування будь-якої частини даних, яка у ньому зберігається. Виконати таке видобування можна, якщо потрібно відновити початкові файли зображень з документа текстового процесора. Якщо підпис zip пошкоджено, це ще не кінець світу: якщо wxMaxima під час збереження виявила якусь помилку, має ще бути файл wxmx~, чий вміст може допомогти у відновленні даних.

І навіть якщо такого файла не існує — якщо встановлено параметр налаштування для оптимізації файлів .wxmx для системи керування версіями, можна перейменувати файл .wxmx на файл .txt і скористатися текстовим редактором для відновлення частини XML із вмісту файла (він починається з `<?xml version="1.0" encoding="UTF-8"?>` і завершується `</wxMaximaDocument>`. До і після цього фрагмента тексту у текстовому редакторі ви побачите непридатні до читання двійкові дані).

Якщо текстовий файл, що містить ці дані (наприклад скопійовані і вставлені як текст до звичайного файла), збережено як файл із суфіксом назви .xml, wxMaxima відомий спосіб відновлення тексту документа з цього файла.

### 3.3 Хочу бачити діагностичні відомості на екрані, перш ніж команда завершить свою роботу

Зазвичай, wxMaxima очікує, доки буде передано усю двовимірну формулу, перш ніж починає виводити дані. Таким чином заощаджується час, який могло б бути витрачено на постійні спроби вивести незавершену формулу. Втім, існує команда `disp`, яка може виводити діагностичні дані негайно, не очікуючи на завершення поточної команди Maxima:

```
for i:1 thru 10 do (  
  disp(i),  
  /* (sleep n) — функція Lisp, якою можна скористатися */  
  /* із додаванням перед нею символу «?». Команда затримує */  
  /* виконання програми (у нашому прикладі, на 3 секунди) */  
  ?sleep(3)  
)$
```

### 3.4 На кресленні показано лише закриту порожню обгортку із повідомленням про помилку

Це означає, що wxMaxima не вдалося прочитати файл, який Maxima мала наказати створити `gnuplot`.

Можливими причинами цієї помилки є такі:

- Команда креслення є частиною стороннього пакунка, наприклад `implicit_plot`, але цей пакунок не було завантажено командою `load()` `Maxima` до того, як команду було передано на виконання.
- `Maxima` спробувала зробити щось, що не змогла обробити поточна встановлена версія `gnuplot`. У цьому випадку, файл із суфіксом назви `.gnuplot` у каталозі, на який вказує змінна `Maxima maxima_userdir`, містить команди, які `Maxima` передавала `gnuplot`. Здебільшого, вміст цього файла є дуже корисним для діагностування проблеми.
- `Gnuplot` було наказано використовувати бібліотеку `pngcairo`, яка надає можливості згладжування зображення та додаткових стилів ліній, але саму програму не було зібрано із підтримкою цієї бібліотеки. Вирішення: зніміть позначку з пункту «Використовувати термінал `cairo` для креслення» у діалоговому вікні налаштувань і не встановлюйте для `wxplot_pngcairo` значення `true` з коду `Maxima`.
- `Gnuplot` не вивів коректного файла `.png`.

### 3.5 Спроба накреслити анімацію призводить до повідомлення «помилка: невизначена змінна»

Значення змінної повзунка, типово, буде підставлено до виразу для креслення, лише якщо воно є видимим у цьому виразі. Використання команди `subst`, яка підставляє змінну повзунка до рівняння, яке слід накреслити, має усунути проблему. Наприкінці розділу Вбудовування анімацій до робочого аркуша наведено приклад.

### 3.6 Втрачено вміст комірок і скасування дій не допомагає

У програмі передбачено окремі функції скасування дій для дій з комірками і для дій усередині комірок, тому така подія має бути дуже рідкісною. Якщо таке все ж сталося, існує декілька способів відновити дані:

- У `wxMaxima` є два списки скасування дій: загальний буфер скасування, який є активним, якщо не позначено жодної комірки, та окремі буфери комірок, які є активним, якщо курсор перебуває у відповідній комірці. Варто спробувати скористатися обома варіантами скасування, щоб спробувати дістати потрібного вам попереднього значення.
- Якщо у вас лишився доступ до мітки, яку `Maxima` призначила комірці, просто введіть до комірки цю мітку, і програма покаже її вміст.
- Якщо доступу не лишилося, не панікуйте. У меню «Перегляд» є пункт для показу панелі журналу, на якій буде показано усі команди `Maxima`, які було нещодавно віддано.
- Якщо ніщо інше не допомагає, у `Maxima` є функція для відтворення усіх команд:

```
playback();
```

### 3.7 wxMaxima запускається із повідомленням «Процес Maxima несподівано перервав роботу.»

Однією з можливих причин є те, що Maxima не вдалося знайти у каталозі, який вказано на сторінці «Maxima» діалогового вікна налаштування wxMaxima, а отже, програму взагалі не запущено. Встановлення адреси коректного виконуваного файла Maxima має виправити цю проблему.

### 3.8 Maxima висне при обчислення і не відповідає на команди

Теоретично можливий випадок, коли wxMaxima не може розпізнати момент завершення обчислень у Maxima і тому ніколи не дізнається про те, що може надсилати нові дані до Maxima. У цьому випадку команда виконання обчислення може відновити синхронізацію між двома програмами.

### 3.9 Версії Maxima на основі SBCL не вистачає оперативної пам'яті

Компілятор Lisp SBCL типово позначається із обмеженням на використання пам'яті, яке надає змогу запускати його на слабких комп'ютерах. При компіляції великих пакунків програмного забезпечення, зокрема larsack, або роботі з великими списками або формулами це обмеження може заважати нормальній роботі. Щоб зняти обмеження, SBCL можна передати параметр `--dynamic-space-size`, який вказуватиме SBCL, скільки мегабайтів пам'яті слід зарезервувати. У 32-бітовому Windows-SBCL можна зарезервувати до 999 мегабайтів. У 64-бітовій версії SBCL, яку запущено у Windows, можна наказати програмі використовувати понад близько 1280 мегабайтів, чого має бути достатньо для збирання larsack.

Одним зі способів передати Maxima (а отже, SBCL) параметри командного рядка є поле «Додаткові параметри для Maxima» вікна налаштування wxMaxima.

### 3.10 При введенні даних програма уповільнюється або пропускає символи в Ubuntu

Встановлення пакунка `ibus-gtk` має усунути проблему. Докладніше про це на сторінці (<https://bugs.launchpad.net/ubuntu/+source/wxwidgets3.0/+bug/1421558>).

### 3.11 wxMaxima припиняє роботу, коли Maxima обробляє грецькі символи або умляути

Якщо вашу Maxima засновано на SBCL, слід додати такі рядки до вашого файлу `.sb-clrc`:

```
(setf sb-impl::*default-external-format* :utf-8)
```

Тека, у якій має бути розташовано цей файл залежить від системи та способу встановлення. Втім, будь-яка Maxima на основі SBCL, яка вже виконала обчислення у якійсь комірці протягом поточного сеансу, без проблем повідомить, де його можна знайти, у відповідь на таку команду:

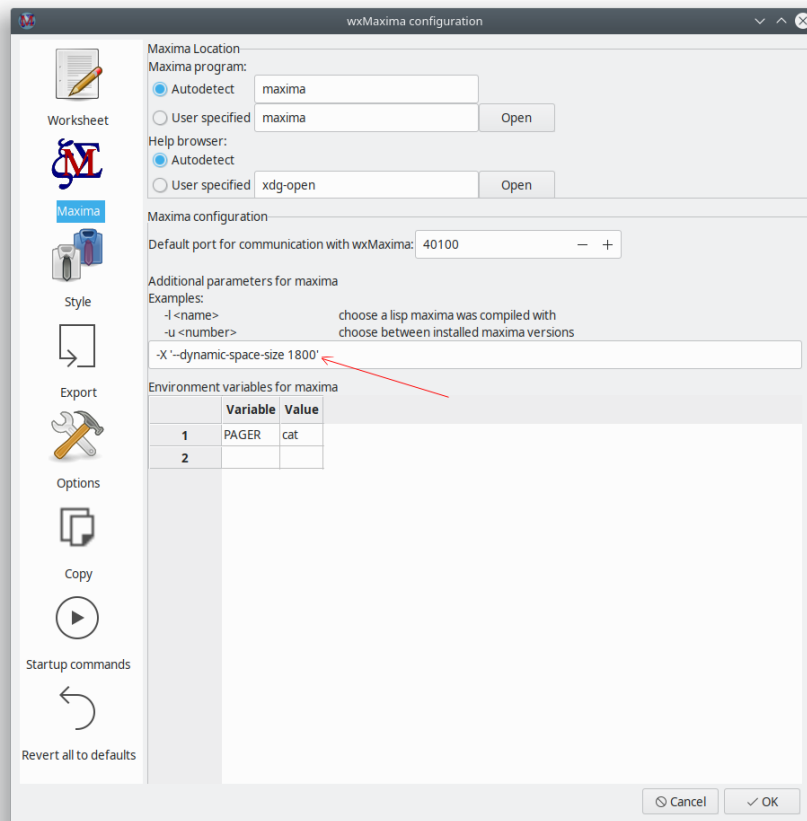


Figure 12: sbcl memory



```
:lisp (sb-impl::userinit-pathname)
```

## 3.12 Креслення

3.12.1 Чи можна зробити так, щоб wxMaxima виводила дані одразу до файлів зображень і до вбудованих до аркуша креслень?

До робочого аркуша вбудовуються файли .png. wxMaxima дозволяє користувачу вказати, де має бути створено ці файли:

```
wxdraw2d(  
    file_name="test",  
    explicit(sin(x),x,1,10)  
);
```

Якщо має бути використано якийсь інший формат зображень, простіше створити зображення окремо, а потім імпортувати їх до робочого аркуша:

```
load("draw");  
pngdraw(name,[contents]):=  
(  
    draw(  
        append(  
            [  
                terminal=pngcairo,  
                dimensions=wxplot_size,  
                file_name=name  
            ],  
            contents  
        )  
    ),  
    show_image(sprintf(false,"~a.png",name))  
);  
pngdraw2d(name,[contents]):=  
    pngdraw(name,gr2d(contents));  
  
pngdraw2d("Test",  
    explicit(sin(x),x,1,10)  
);
```

3.12.2 Чи можна встановити співвідношення розмірів зображення креслення?

Не безпосередньо, за допомогою Maxima. Але для цього є команди `gnuplot`:

```
wxdraw2d(  
    proportional_axis=xy,  
    explicit(sin(x),x,1,10)  
,wxplot_size=[1000,1000];
```

---

## 4 Поширені питання та відповіді на них

### 4.1 Чи є якийсь спосіб умістити більше тексту на сторінку LaTeX?

Так. Скористайтеся пакунком LaTeX «geometry» для визначення розмірів полів.

Ви можете додати такий рядок до преамбули LaTeX (наприклад, за допомогою відповідного поля у діалоговому вікні налаштування («Експорт->Додаткові рядки до преамбули TeX») для встановлення розміру полів у 1 см):

```
\usepackage[left=1cm,right=1cm,top=1cm,bottom=1cm]{geometry}
```

### 4.2 Чи передбачено у програмі «темний» режим?

Якщо бібліотека wxWidgets є достатньо новою, wxMaxima автоматично працюватиме у темному режимі, якщо у ньому працює решта операційної системи. Сам робочий аркуш типово має світле тло. Втім, колір тла можна змінити. Крім того, ви можете скористатися пунктом меню «Перегляд/Інвертувати яскравість робочого аркуша», за допомогою якого можна швидко перетворити робочий аркуш з темного на світлий, і навпаки.

### 4.3 wxMaxima іноді повисає на декілька секунд під час першої хвилини роботи

wxMaxima передає певні складні завдання, зокрема завдання з обробки підручника з Maxima, який складається з понад 1000 сторінок, фоновим завданням, що, зазвичай, відбувається зовсім непомітно. Втім, у момент, коли може знадобитися результат виконання такого завдання, wxMaxima може знадобитися декілька секунд для відновлення нормальної роботи.

---

## 5 Параметри командного рядка

У більшості операційних систем передбачено простіші способи запуску програми, ніж командний рядок, тому навряд ви запускатиме програму з командного рядка. Втім, у wxMaxima передбачено декілька корисних параметрів командного рядка.

- -v або --version: вивести дані щодо версії програми
- -h або --help: вивести короткий довідковий текст
- -o або --open=<рядок>: відкрити файл, який передано як аргумент до цього параметра командного рядка
- -e або --eval: обробити файл після його відкриття
- -b або --batch: якщо ви відкриваєте файл з командного рядка, усі комірки у цьому файлі обчислюються, а потім програма виконує збереження файла.

Це, наприклад, корисно, якщо сеанс, який описано у файлі, призводить до того, що Maxima щось виводить до файлів. Пакетну обробку буде зупинено, якщо wxMaxima виявить, що Maxima вивела повідомлення про помилку, і призупинить обробку, якщо Maxima задасть питання: математика є дещо інтерактивною за природою, тому повністю автономна пакетна обробка можлива далеко не завжди.

- `--logtostdout`: записувати до `stderr` і усі «діагностичні повідомлення» бічної панелі.
- `--pipe`: передавати каналом повідомлення від Maxima до `stdout`.
- `--exit-on-error`: завершити роботу програми за будь-якої помилки maxima.
- `-f` або `--ini=<рядок>`: скористатися файлом `init`, який передано як аргумент цього параметра командного рядка
- `-u, --use-version=<рядок>`: скористатися версією maxima `<рядок>`.
- `-l, --lisp=<рядок>`: скористатися Maxima, яку зібрано за допомогою компілятора Lisp `<рядок>`.
- `-X, --extra-args=<рядок>`: надає змогу вказати додаткові аргументи Maxima
- `-m` або `--maxima=<рядок>`: надає змогу вказати розташування виконуваного файла maxima
- `--enableipc`: дозволяє Maxima керувати wxMaxima шляхом міжпроцесорної взаємодії. Будьте обережні із використанням цього параметра.

Замість символу мінуса у деяких операційних системах перед перемикачами командного рядка використовується тире.