

## Spis treści

Przedmowa.....	5
1. Wprowadzenie.....	6
1.1. Instalacja.....	6
1.2. Menu, pasek narzędzi oraz skróty klawiaturowe.....	7
1.3. Operatory, stałe i funkcje matematyczne .....	14
1.4. Style prezentacji wyników, odwoływanie się do etykiet komórek ...	19
1.5. Korzystanie z pomocy programu.....	20
2. Wykonywanie obliczeń symbolicznych oraz numerycznych.....	23
2.1. Wartości przybliżone, ustalanie precyzji obliczeń .....	23
2.2. Podstawianie oraz deklaracje środowiska lokalnego.....	27
3. Definiowanie funkcji.....	31
4. Przekształcanie oraz upraszczanie wyrażeń algebraicznych .....	37
5. Liczby zespolone.....	47
6. Wykresy .....	52
6.1. Podstawowe polecenia graficzne .....	52
6.2. Pakiet <i>draw</i> .....	62
7. Rozwiązywanie równań i nierówności oraz układów równań .....	77
8. Wektory, macierze i układy równań liniowych .....	85
8.1. Wektory.....	85
8.2. Macierze .....	90
8.3. Układy równań liniowych .....	100
8.4. Macierze i wyznaczniki funkcyjne .....	108
9. Granice i ciągłość funkcji .....	112
10. Rachunek różniczkowy funkcji jednej zmiennej.....	118
11. Ciągi liczbowe i funkcyjne .....	136
12. Szeregi liczbowe i funkcyjne.....	142
13. Rachunek całkowy funkcji jednej zmiennej .....	154

<b>14. Rachunek różniczkowy funkcji wielu zmiennych .....</b>	<b>174</b>
<b>15. Rachunek całkowy funkcji wielu zmiennych .....</b>	<b>187</b>
<b>16. Równania różniczkowe zwyczajne .....</b>	<b>196</b>
<b>17. Równania różnicowe .....</b>	<b>211</b>
<b>18. Elementy programowania .....</b>	<b>221</b>
<b>Literatura .....</b>	<b>223</b>

## **Przedmowa**

Proponowany Czytelnikowi skrypt przedstawia możliwości, jakie daje praca z Systemem Algebry Komputerowej (CAS) Maxima, wspomagającym wykonywanie matematycznych obliczeń symbolicznych oraz numerycznych.

Pozycja ta kierowana jest do studentów, którzy, uczestnicząc w ćwiczeniach laboratoryjnych w pracowni komputerowej, mają możliwość uzupełniania zdobywanej wiedzy matematycznej o doświadczenia związane ze stosowaniem technologii informatycznych oraz do wszystkich, chcących skorzystać z programu Maxima jako narzędzia do rozwiązywania problemów matematycznych i technicznych.

Program Maxima jest pomocnym narzędziem do wizualizacji różnych zagadnień. Umożliwia lepsze i szybsze zrozumienie teorii matematycznych, pozwala kontrolować kolejne etapy obliczeń, rozważać różne sposoby rozwiązywania problemów i interpretować uzyskane wyniki, a także wyeliminować błędy w przypadku skomplikowanych rachunków. Operacje mogą być wykonywane nie tylko na liczbach, także na zmiennych oraz funkcjach.

Jak każde narzędzie informatyczne, nie jest to program pozbawiony wad, więc należy krytycznie podchodzić do uzyskanych wyników i mieć świadomość ograniczeń programu, z którymi możemy się zetknąć podczas zajęć laboratoryjnych. Maxima jest programem bezpłatnym i otwartym, stale rozwijanym przez społeczność użytkowników.

W skrypcie zostały przedstawione rozwiązania zadań przykładowych z wykorzystaniem Maximy w wersji 5.31.2 oraz zadania do samodzielnego rozwiązania wraz z odpowiedziami.

Wśród zadań przykładowych, jak i zadań do samodzielnego rozwiązania, można znaleźć takie, które ilustrują zastosowanie teorii matematycznych w analizie zagadnień optymalizacyjnych, fizycznych i ekonomicznych.

Życzymy przyjemnej pracy z Maximą

Autorki

## 1. Wprowadzenie



Program Maxima wywodzi się z opracowanego pod koniec lat sześćdziesiątych w Massachusetts Institute of Technology programu DOE-Macsyma. W 1998 roku jeden z jego twórców William Schelter uzyskał pozwolenie na udostępnienie kodu źródłowego Macsymy na licencji GPL (Gnu Public License), a dwa lata później zainicjował projekt Maxima na SourceForge, którego celem był dalszy rozwój tego Systemu Algebry

Komputerowej. Od tamtego czasu społeczność użytkowników i deweloperów Maximy stale rośnie.

Maxima jest dostępna zarówno w postaci kodu źródłowego jak i binarnych pakietów instalacyjnych na platformach: MS Windows, Linux oraz MacOSX. Silnik Maximy jest aplikacją konsolową, dostępne są także nakładki graficzne dla poszczególnych systemów: wxMaxima, Xmaxima, iMaxima. W naszym skrypcie będziemy omawiać działanie Maximy wraz z nakładką graficzną wxMaxima dla systemu Windows. Nakładka ta pozwala mniej doświadczonym użytkownikom programu, nie znającym nazw i składni, na swobodną pracę w zakresie podstawowych poleceń. Elementami tego interfejsu są notatnik, menu oraz palety. Wykonane polecenia można zapisać do pliku. Pliki-skrypty wxMaximy mają rozszerzenia wxm albo wxmx. Zapis do formatu wxmx pozwala zachować dodane do skryptu obrazy oraz wyniki obliczeń. Ponadto, skonstruowany dokument, który zawiera tekst, obliczenia i wykresy, możemy dzięki wxMaximie eksportować do formatu html albo tex. Dodatkową zaletą dla początkującego użytkownika wxMaximy jest dostępność polskiego tłumaczenia interfejsu.

W dalszych rozdziałach zaprezentujemy możliwości programu (a czasem też ograniczenia) głównie w zakresie obliczeń symbolicznych dla wybranych tematów realizowanych na studiach I stopnia na Politechnice Łódzkiej. W skrypcie została przyjęta konwencja oznaczeń zgodna z [5].

### 1.1. Instalacja

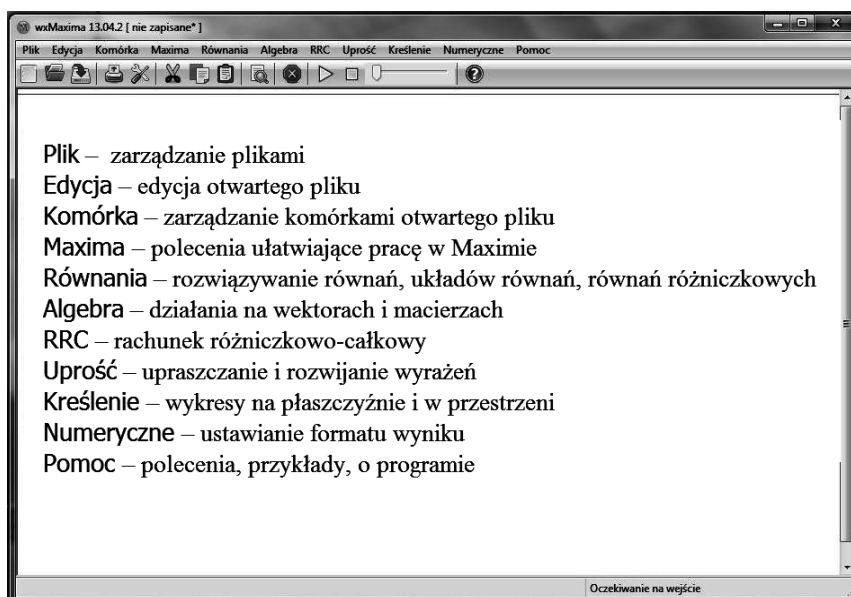
Najnowsza wersja dostępna w trakcie składania tego skryptu, to wersja 5.34.1 (8.09.2014). Należy jednak wspomnieć, że nie wszystkie wersje są dystrybuowane jako pakiety instalacyjne dla środowiska Windows, stąd obecnie najnowsza (oficjalna) dla systemu Windows wersja to: 5.31.2 (8.10.2013).

Pakiety instalacyjne są dostępne na stronie:

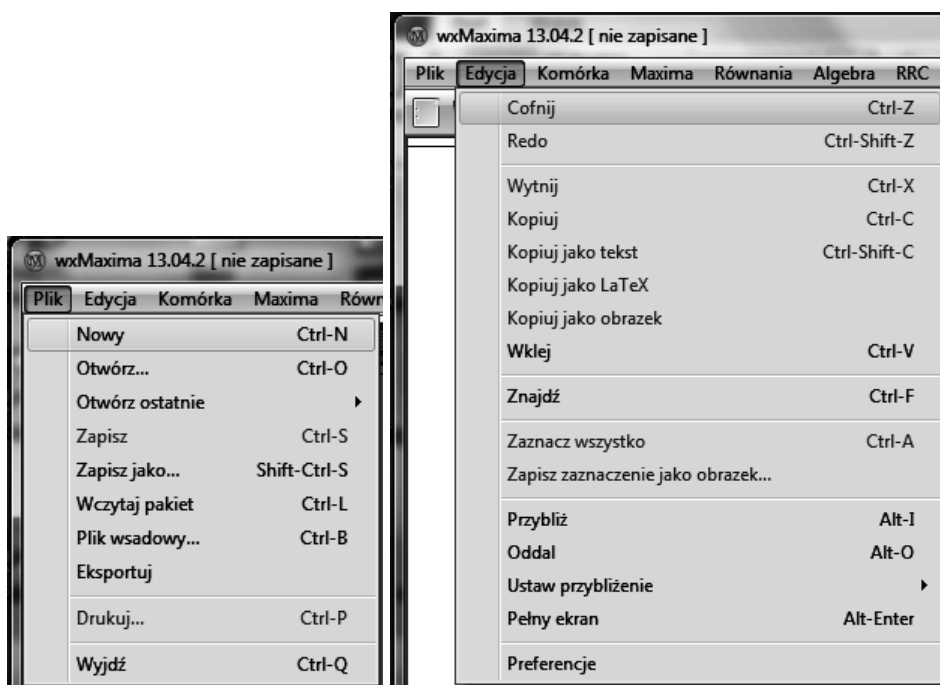
<http://sourceforge.net/projects/maxima/files/>.

## 1.2. Menu, pasek narzędzi oraz skróty klawiaturowe

Najpierw zapoznajmy się z graficznym interfejsem wxMaximy, który pozwala użytkownikowi w prosty sposób łączyć w jednym dokumencie tekst, obliczenia oraz wykresy.



Rysunek 1.1. Menu



Rysunek 1.2. Menu-Plik

Rysunek 1.3. Menu-Edycja

## Menu-Plik/Wczytaj pakiet

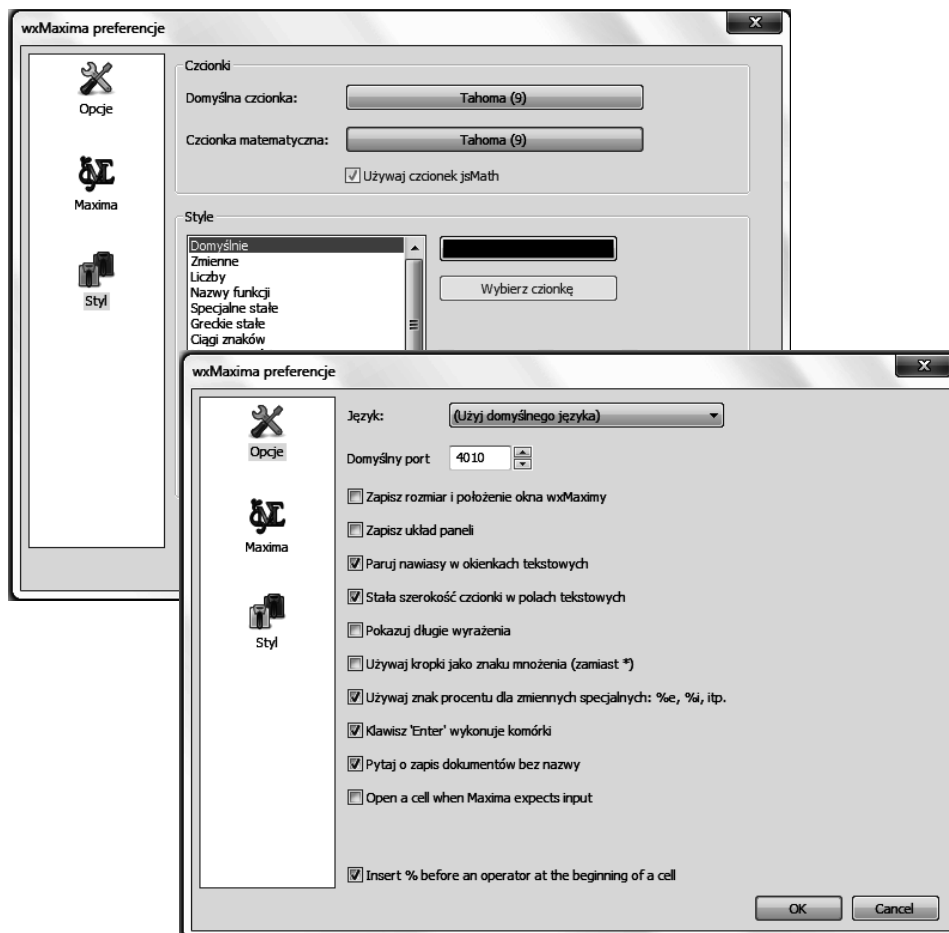
Oprócz korzystania z funkcji menu, paska narzędzi, paneli i udogodnień myszki, wiele możliwości zyskujemy, wczytując pakiety Maximy dostępne jako pliki \*.mac (czasem \*.lisp) w katalogu Maxima-5.31.2/share/maxima/5.31.2/share.

Możemy też zbudować własny plik wsadowy z rozszerzeniem **mac**, który wczytuje się przez **Menu-Plik/Plik wsadowy** (polecenie **batch**(".....")) albo pakiet (moduł z definicjami funkcji) z rozszerzeniem **mac** wczytywany przez **Menu-Plik/Wczytaj pakiet** lub polecenie **load**(".....").

```
(%i1) load(distrib);
(%o1) C:/PROGRA~2/MAXIMA~1.2/share/maxima/5.31.2/share/distrib/distrib.mac
```

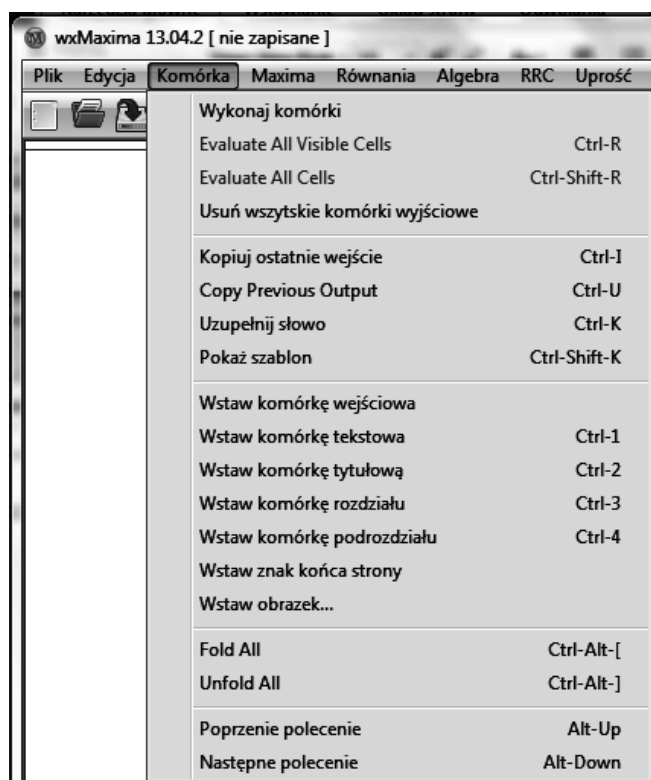
## Menu-Edycja/Preferencje

Pojawiające się okno pozwala na ustawienie własnych preferencji.



Rysunek 1.4. Preferencje

Omawiając pracę z notatnikiem wxMaxima, używa się określenia **komórka**. Komórka to podstawowy element konstrukcyjny dokumentu. W zależności od typu (np. komórka wejścia, komórka tekstowa, komórka rozdziału) może ona służyć do obliczeń, wprowadzania komentarzy albo logicznego podziału dokumentu. Komórki wejścia oraz komórki tekstowe są spięte nawiasem, co pokazuje ich początek i koniec oraz pozwala na zaznaczanie, kopiowanie, usuwanie, wklejanie oraz ukrywanie ich zawartości (kliknięcie na trójkąt w lewym górnym rogu). Komórki podziału dokumentu (tytułowa, rozdziału, podrozdziału) również można zwinąć/rozwinąć, klikając na kwadrat (patrz rysunek 1.7).



Rysunek 1.5. Menu-Komórka

### Skróty klawiaturowe

**Ctrl-G** – przerywa bieżące zadanie Maximy

**Ctrl-R** – wykonuje wszystkie nieukryte komórki

**Ctrl-Shift-R** – wykonuje wszystkie komórki

**Ctrl-I** – kopiuje ostatnie wejście

**Ctrl-U** – kopiuje wyjście

**Ctrl-K** – przedstawia warianty uzupełnienia rozpoczętego polecenia (działa już od pierwszego znaku, zarówno dla funkcji wbudowanych jak i zdefiniowanych przez użytkownika lub wczytanych za pomocą pliku wsadowego)

**Ctrl-Shift-K** – podobnie jw., z dodatkowym uzupełnianiem składni polecenia

**Ctrl-1** – wstawia komórkę tekstową

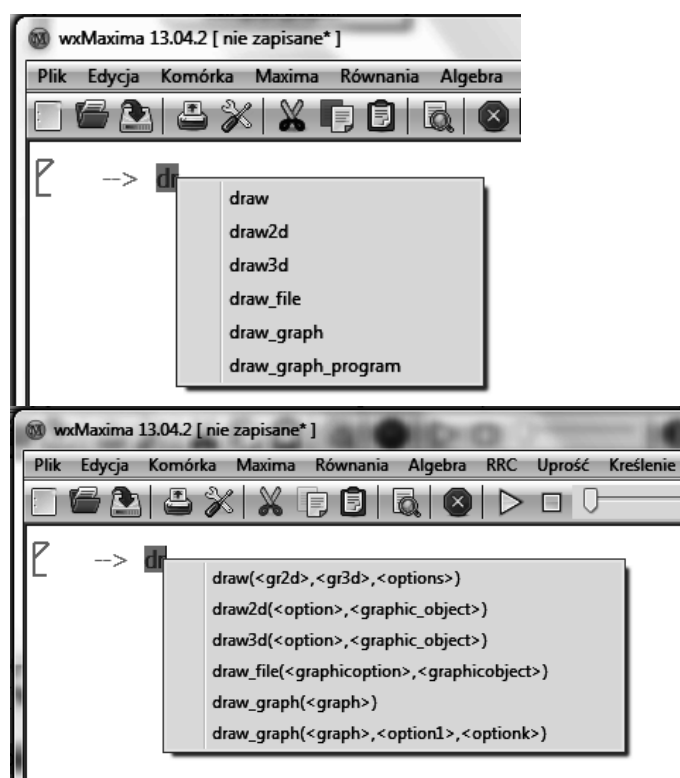
**Ctrl-2** – wstawia komórkę tytułową

**Ctrl-3** – wstawia komórkę rozdziału

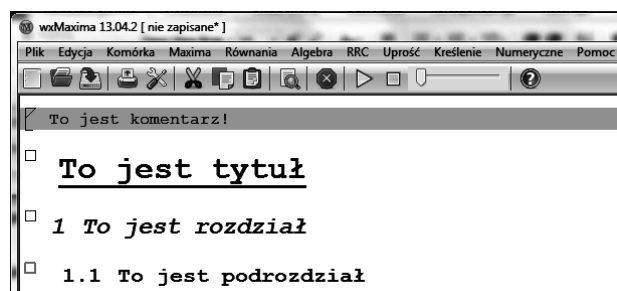
**Ctrl-4** – wstawia komórkę podrozdziału

**Ctrl-Alt-[ ( Ctrl-Alt-)]** – zwija (rozwija) wszystkie rozdziały i podrozdziały

**Alt-Up, Alt-Down** – pozwalają przeglądać historię wprowadzanych poleceń



Rysunek 1.6. Uzupełnianie polecenia i składni



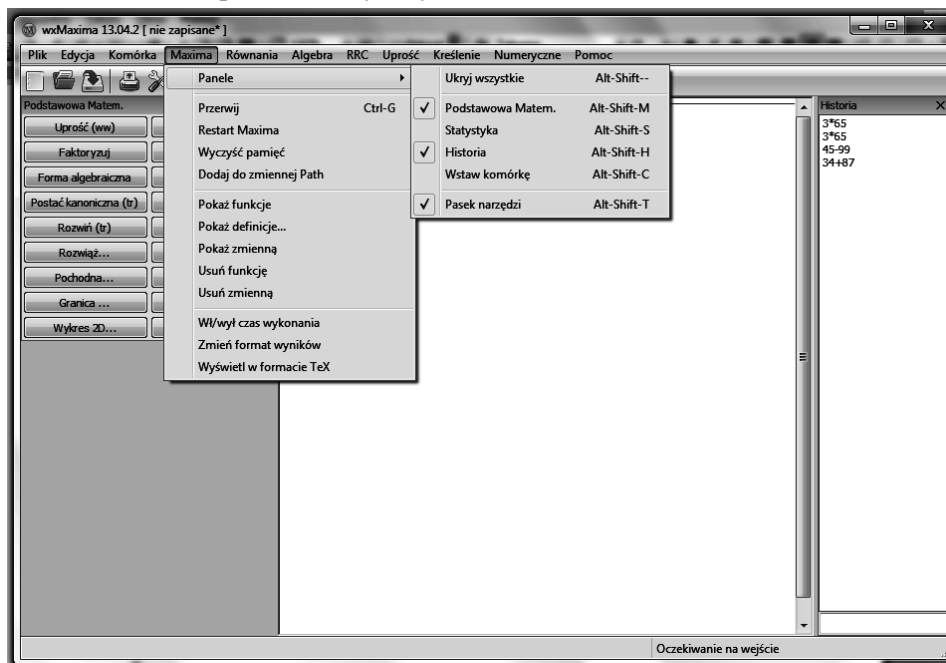
Rysunek 1.7. Struktura dokumentu

### Uwaga 1

Aktywne komórki są zaznaczone kolorem czerwonym.

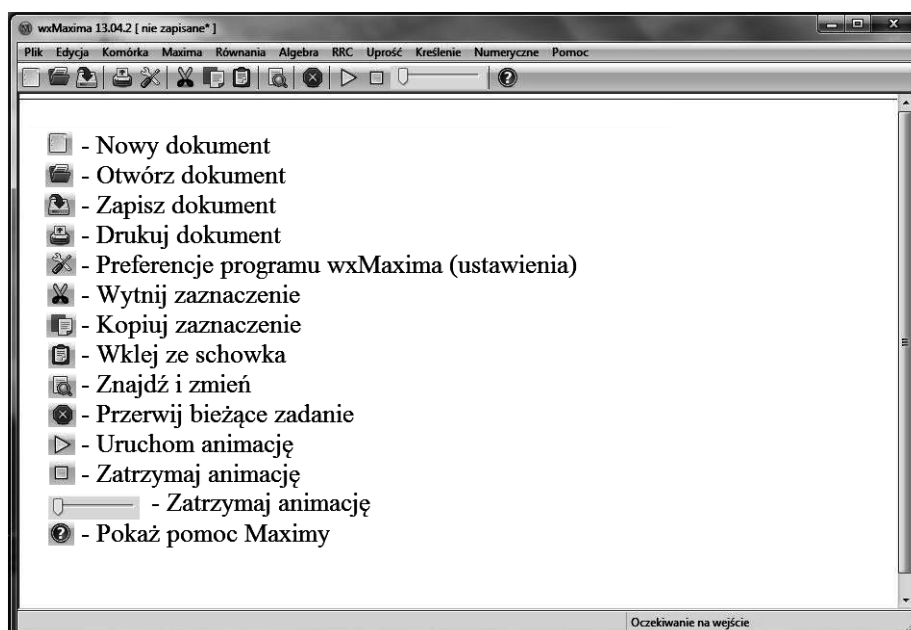


W celu usprawnienia pracy istnieje możliwość wykorzystania paska narzędzi, okna historii oraz paneli tematycznych.

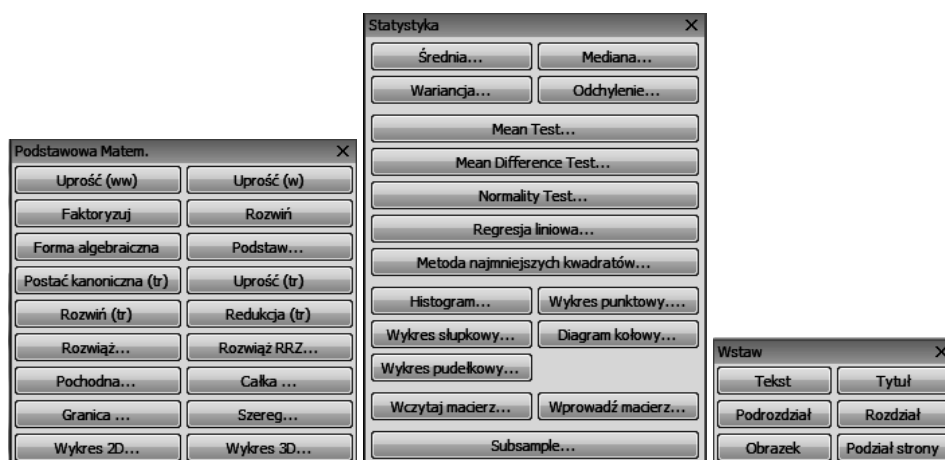


Rysunek 1.8. Menu-Maxima/Panele

Warto zapamiętać skróty klawiaturowe dla używanych paneli lub w **Preferencjach** zaznaczyć opcję: **Zapisz układ paneli**.



Rysunek 1.9. Narzędzia



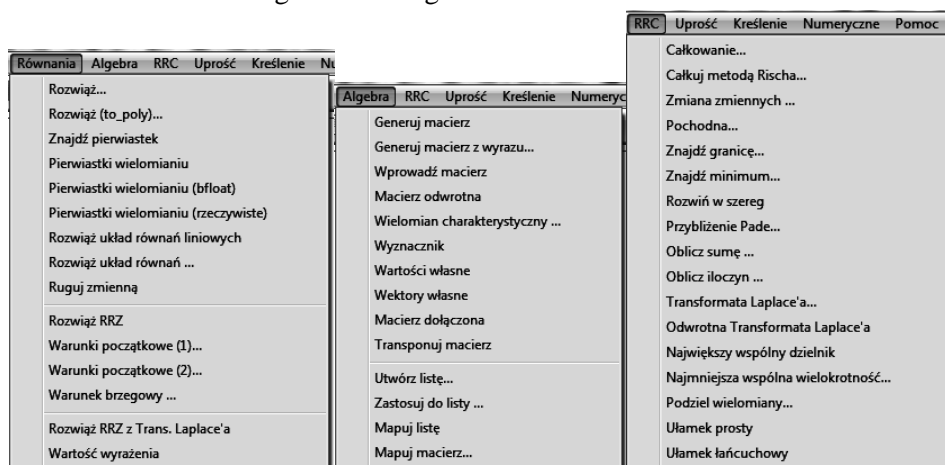
Rysunek 1.10. Panele tematyczne

W **Menu-Maxima** znajdziemy również polecenie **Restart Maxima**.

Zaczyna ono sesję Maximy od nowa, tzn. resetuje ustawienia, numerację etykiet, usuwa wartości, funkcje, tablice, własności.

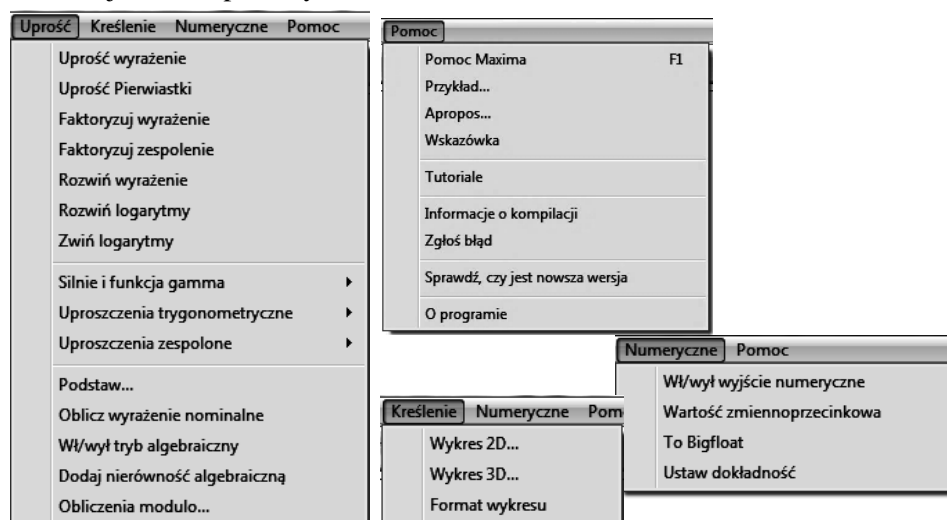
Natomiast wybór **Maxima/Przerwij** (skrót klawiaturowy Ctrl-G) przerywa jedynie bieżące obliczenia.

Poniżej są przedstawione rozwinięcia Menu zawierające polecenia dotyczące: rozwiązywania równań oraz układów równań, rachunku macierzowego oraz rachunku różniczkowego i całkowego.



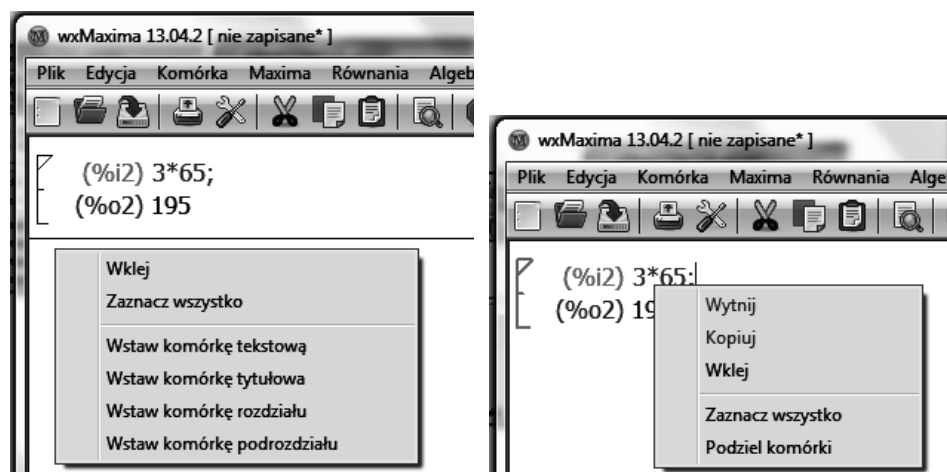
Rysunek 1.11. Menu-Równania, Menu-Algebra, Menu-RRC

W poniższych rozwinięciach Menu znajdziemy polecenia upraszczające, polecenia służące do rysowania wykresów, ustalania formatu wyniku numerycznego oraz znajdowania pomocy.



Rysunek 1.12. Menu-Uprość, Menu-Kreślenie, Menu-Numeryczne, Menu-Pomoc

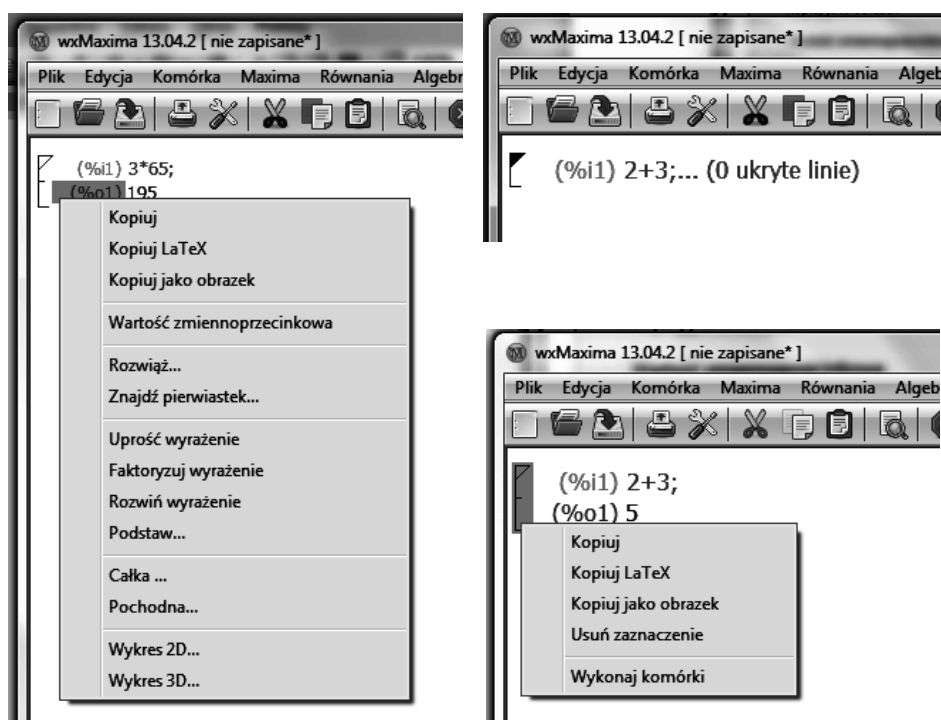
Dostęp do poleceń Maximy możemy też uzyskać, klikając **prawym przyciskiem myszy**. W zależności od miejsca kliknięcia: w wolnym polu, na linii wprowadzania input – (% i nr komórki), na linii wyniku output – (% o nr komórki) lub na zaznaczeniu komórki otwierają się odpowiednio różne menu kontekstowe:



Rysunek 1.13

Na powyższym rysunku widoczny jest też **poziomy kursor**, który pojawia się po kliknięciu w miejscu poza komórkami. Rozpoczęcie wprowadzania tekstu tworzy nową komórkę wejścia.

Warunkiem wykonania polecenia jest zakończenie linii **znakiem średnika ( ; )**. W wxMaximie znak ten jest dodawany automatycznie.



Rysunek 1.14

### 1.3. Operatory, stałe i funkcje matematyczne

Tabela 1.1

OPERATORY MATEMATYCZNE	OPIS
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
^	potęgowanie
**	potęgowanie – operator równoważny ^
<b>sqrt()</b>	pierwiastek kwadratowy
!	silnia
!!	podwójna silnia

```
(%i1) 5+3*4-8/2+3!;
(%o1) 19
```

```
(%i2) 2^3+2**5+4!!+sqrt(2);
(%o2)  $\sqrt{2} + 48$ 
```

Tabela 1.2

PRZYPISANIE/ DEKLARACJE/ CZYSZCZENIE	OPIS
<b>z:a</b>	przypisanie zmiennej z wartości $a$
<b>:=</b>	definiowanie funkcji (patrz tabela 3.1)
<b>=</b>	definiowanie równania
<b>values</b>	podaje listę zmiennych z przypisaniem
<b>remvalue(z)</b>	usuwa przypisanie dla zmiennej $z$
lub <b>remvalue(all)</b>	lub wszystkich zmiennych
<b>functions, remfunction(f)</b>	podobnie jw., tylko dla funkcji
<b>declare(a,w)</b>	deklaruje własność $w$ dla $a$ <sup>1)</sup>
<b>assume(w)</b>	ustala warunek z relacją np.: $>$ , $<$ , ...
<b>facts(w)</b>	podaje fakty związane z $w$
<b>forget(w)</b>	usuwa fakty związane z $w$
<b>remove(a,w)</b>	usuwa własność $w$ dla $a$ <sup>1)</sup>
<b>reset()</b>	resetuje zmienne i opcje globalne do ich domyślnych ustawień
<b>kill(labels)</b>	resetuje numerację etykiet
<b>kill(a)</b>	usuwa wartości, funkcje, tablice oraz własności związane z $a$
<b>kill(all)</b>	usuwa wartości, funkcje, tablice, własności, resetuje etykiety
<b>\$</b>	na końcu polecenia (zamiast $;$ ) powoduje, że nie wyświetla się output

<sup>1)</sup>  $a$  może być funkcją lub zmienną.

Poniżej przykłady zastosowań poleceń z tabeli 1.2.

$\left[ \begin{array}{l} (\%i1) a:3; \\ (\%o1) 3 \end{array} \right.$	$\left[ \begin{array}{l} (\%i2) \text{assume}(a>0,a<1); \\ (\%o2) [a>0, a<1] \end{array} \right.$	$\left[ \begin{array}{l} (\%i7) \text{assume}(b<=0); \\ (\%o7) [b<=0] \end{array} \right.$
$\left[ \begin{array}{l} (\%i2) f(x):=x^2; \\ (\%o2) f(x):=x^2 \end{array} \right.$	$\left[ \begin{array}{l} (\%i3) \text{limit}(a^n, n, \text{inf}); \\ (\%o3) 0 \end{array} \right.$	$\left[ \begin{array}{l} (\%i8) \text{abs}(b); \\ (\%o8) -b \end{array} \right.$
$\left[ \begin{array}{l} (\%i3) f(5)+a; \\ (\%o3) 28 \end{array} \right.$	$\left[ \begin{array}{l} (\%i4) \text{forget}(a>0,a<1); \\ (\%o4) [a>0, a<1] \end{array} \right.$	$\left[ \begin{array}{l} (\%i9) \text{facts}(); \\ (\%o9) [a>1, 0>=b] \end{array} \right.$
$\left[ \begin{array}{l} (\%i4) \text{kill}(\text{all}); \\ (\%o0) \text{done} \end{array} \right.$	$\left[ \begin{array}{l} (\%i5) \text{assume}(a>1); \\ (\%o5) [a>1] \end{array} \right.$	$\left[ \begin{array}{l} (\%i10) \text{forget}(b<=0); \\ (\%o10) [b<=0] \end{array} \right.$
$\left[ \begin{array}{l} (\%i1) a; \\ (\%o1) a \end{array} \right.$	$\left[ \begin{array}{l} (\%i6) \text{limit}(a^n, n, \text{inf}); \\ (\%o6) \infty \end{array} \right.$	$\left[ \begin{array}{l} (\%i11) \text{facts}(); \\ (\%o11) [a>1] \end{array} \right.$

Warto zauważyć, że polecenie **kill** ma szerszy zakres działania niż polecenia **remvalue**, **remfunction**, **remove**. Czyści ono nie tylko wartości oraz własności danego obiektu (zmiennej, funkcji), ale również etykiety i inne ustawienia.

Przy stosowaniu pewnych deklaracji lub poleceń czyszczących w wyniku pojawia się **done**. Stawiając \$ zamiast ; unikniemy wyświetlenia tego komunikatu.

<pre>(%i1) a:3\$ b:4\$       a+b;       values; (%o3) 7 (%o4) [a, b]</pre>	<pre>(%i5) remvalue(a)\$       a+b;       values; (%o6) a + 4 (%o7) [b]</pre>
--	---

Poniżej przykład deklaracji dla funkcji: **oddfun** (funkcja nieparzysta) oraz **evenfun** (funkcja parzysta).

<pre>(%i1) kill(f)\$</pre>	<pre>(%i4) kill(f)\$</pre>
<pre>(%i2) declare(f,oddfun); (%o2) done</pre>	<pre>(%i5) declare(f,evenfun); (%o5) done</pre>
<pre>(%i3) f(-x)+f(x); (%o3) 0</pre>	<pre>(%i6) f(-x)+f(x); (%o6) 2 f(x)</pre>

Tabela 1.3

OPERATORY PORÓWNANIA	OPIS
=	równy
#	różny (negacja równości)
> ( >= )	wiekszy (wiekszy lub równy)
< ( <= )	mniejszy (mniejszy lub równy)

Tabela 1.4

OPERATORY LOGICZNE	OPIS
<b>not</b>	operator logiczny – negacja
<b>and</b>	operator logiczny – koniunkcja
<b>or</b>	operator logiczny – alternatywa
<b>is(p) lub p, pred</b>	sprawdza wartość logiczną zdania <i>p</i>

<pre>(%i1) not 2&gt;4; (%o1) true</pre>	<pre>(%i1) is(5=4); (%o1) false</pre>
<pre>(%i2) 2&gt;4 or 2&lt;4; (%o2) true</pre>	<pre>(%i2) 5=4,pred; (%o2) false</pre>
<pre>(%i3) 2&gt;4 and 2&lt;4; (%o3) false</pre>	<pre>(%i3) is (5#4); (%o3) true</pre>
	<pre>(%i4) is (5&lt;=4); (%o4) false</pre>

Tabela 1.5

STAŁE I SYMBOLE	OPIS
<b>%pi</b>	liczba $\pi$
<b>%e</b>	liczba Eulera, podstawa logarytmu naturalnego
<b>%phi</b>	„złota proporcja” $\frac{(1+\sqrt{5})}{2}$
<b>%i</b>	jednostka urojona $i$
<b>inf</b>	$+\infty$
<b>minf</b>	$-\infty$

[ (%i1) %e, numer;  
 (%o1) 2.718281828459045

Więcej na temat wyznaczania wartości przybliżonych znajduje się w rozdziale 2.

[ (%i2) %pi, numer;  
 (%o2) 3.141592653589793

[ (%i3) %phi, numer;  
 (%o3) 1.618033988749895

[ (%i4) %i^2;  
 (%o4) -1

### Przypomnijmy definicje wybranych funkcji:

- funkcja secans:  $\sec x = \frac{1}{\cos x}$ ;
- funkcja cosecans:  $\csc x = \frac{1}{\sin x}$ ;
- funkcja sufit:  $[x]$  – najmniejsza liczba całkowita nie mniejsza od  $x$ ;
- funkcja podłoga:  $[x]$  – największa liczba całkowita nie większa od  $x$ ;
- funkcja znak:  $\operatorname{sgn} x = \begin{cases} -1, & x < 0 \\ 0, & x = 0; \\ 1, & x > 0 \end{cases}$
- funkcja sinus hiperboliczny:  $\sinh x = \frac{e^x - e^{-x}}{2}$ ;
- funkcja cosinus hiperboliczny:  $\cosh x = \frac{e^x + e^{-x}}{2}$ ;
- funkcja tangens hiperboliczny:  $\tanh x = \frac{\sinh x}{\cosh x}$ ;
- funkcja cotangens hiperboliczny:  $\coth x = \frac{\cosh x}{\sinh x}$ .

#### Uwaga 1

Maxima rozróżnia duże i małe litery. Argumenty funkcji są umieszczane w nawiasach okrągłych. Nazwy funkcji wbudowanych rozpoczynają się małą literą.

Tabela 1.6

FUNKCJE MATEMATYCZNE	OPIS
<b>abs(x)</b>	$ x $ (wartość bezwzględna)
<b>min(x<sub>1</sub>, ..., x<sub>n</sub>)</b>	najmniejsza z liczb $x_1, \dots, x_n$
<b>max(x<sub>1</sub>, ..., x<sub>n</sub>)</b>	największa z liczb $x_1, \dots, x_n$
<b>sqrt(x)</b>	$\sqrt{x}$ (pierwiastek kwadratowy)
<b>ceiling(x)</b>	$\lceil x \rceil$ (sufit)
<b>floor(x)</b>	$\lfloor x \rfloor$ (podłoga)
<b>round(x)</b>	przybliżenie całkowite liczby $x$
<b>signum(x)</b>	$\text{sgn}(x)$ (znak)
<b>%e^x</b> albo <b>exp(x)</b>	$e^x$ (exponens)
<b>a^x</b>	$a^x$
<b>log(x)</b>	$\ln(x)$ (logarytm naturalny)
<b>log(x)/log(a)</b>	$\log_a(x)$
<b>sin(x)</b>	$\sin(x)$
<b>cos(x)</b>	$\cos(x)$
<b>tan(x)</b>	$\text{tg}(x)$
<b>cot(x)</b>	$\text{ctg}(x)$
<b>sec(x)</b>	$\text{sec}(x)$ (secans)
<b>csc(x)</b>	$\text{csc}(x)$ (cosecans)
<b>asin(x)</b>	$\arcsin(x)$ (arcus sinus)
<b>acos(x)</b>	$\arccos(x)$ (arcus cosinus)
<b>atan(x)</b>	$\text{arctg}(x)$ (arcus tangens)
<b>acot(x)</b>	$\text{arcctg}(x)$ (arcus cotangens)
<b>sinh(x)</b>	$\sinh(x)$ (sinus hiperboliczny)
<b>cosh(x)</b>	$\cosh(x)$ (cosinus hiperboliczny)
<b>tanh(x)</b>	$\text{tgh}(x)$ (tangens hiperboliczny)
<b>coth(x)</b>	$\text{ctgh}(x)$ (cotangens hiperboliczny)
<b>random(x)</b>	wyznacza wartość pseudolosową z przedziału $[0, x)$ tego samego typu co $x$ , w szczególności np. dla liczb naturalnych liczbę ze zbioru $\{0, 1, 2, \dots, n - 1\}$
<b>binomial(n,k)</b>	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$

```
[ (%i1) abs(ceiling (%pi)-floor (%pi));
  (%o1) 1
```

```
[ (%i3) min(signum(%pi),floor(%pi));
  (%o3) 1
```

```
[ (%i2) max(signum(%pi),floor(%pi));
  (%o2) 3
```

```
[ (%i4) round (sqrt(2));
  (%o4) 1
```



[	(%i5) $\exp(3)+\log(\%e);$
[	(%o5) $\%e^3 + 1$
[	(%i6) $\sec(\%pi/6)+\csc(\%pi/6);$
[	(%o6) $\frac{2}{\sqrt{3}} + 2$
[	(%i7) $(\exp(1)-\exp(-1))/2, \text{ numer};$
[	(%o7) 1.175201193643801
[	(%i8) $\sinh(1), \text{ numer};$
[	(%o8) 1.175201193643801
[	(%i9) $\text{is}(\text{float}((\exp(1)+\exp(-1))/2)=\text{float}(\cosh(1)));$
[	(%o9) true
[	(%i10) $\sin(\%pi/4)+\cos(\%pi)+\tan(0)+\cot(\%pi/2);$
[	(%o10) $\frac{1}{\sqrt{2}} - 1$
[	(%i11) $\text{asin}(1)+\text{acos}(0)+\text{atan}(\text{sqrt}(3))+\text{acot}(1);$
[	(%o11) $\frac{19\pi}{12}$

#### 1.4. Style prezentacji wyników, odwoływanie się do etykiet komórek

Tabela 1.7

POLECENIA	OPIS
'	wprowadzony przed znakiem operatora wstrzymuje wykonanie operacji
"	wprowadzony przed znakiem operatora nakazuje wykonanie operacji
<b>display(w)</b>	wyświetla nazwę i wartość wyrażenia w
<b>print(w)</b> albo	wykonuje i wyświetla wyrażenie jako łańcuch
<b>print(„t1”,w1,...)</b>	znakowy, pozwala też łączyć teksty i zmienne
<b>sprint(w)</b>	podobnie jw. tylko nie łamie linii w długich wyrażeniach
<b>%</b>	przechowuje ostatni output, można go przywołać
<b>%th(k)</b>	przywołuje k-ty wstecz output
<b>%i7</b>	przywołuje input o numerze 7
<b>%o7</b>	przywołuje output o numerze 7

```

[ (%i1) 2+3;
  (%o1) 5

[ (%i2) a:%;
  (%o2) 5

[ (%i3) a;
  (%o3) 5

[ (%i1) diff(a*x^2, x);
  (%o1) 2 a x

[ (%i2) a:b;b:3;
  (%o2) b
  (%o3) 3

[ (%i6) 'integrate(x^2, x);
  (%o6)  $\int x^2 dx$ 

[ (%i7) display(integrate(x^2, x))$
   $\int x^2 dx = \frac{x^3}{3}$ 

[ (%i4) b:3$

[ (%i5) x^2=b;
  (%o5)  $x^2 = 3$ 

[ (%i6) solve([%], [x]);
  (%o6)  $[x = -\sqrt{3}, x = \sqrt{3}]$ 

[ (%i4) diff(a*x^2, x);
  (%o4) 2 b x

[ (%i5) "%;
  (%o5) 6 x

[ (%i8) sprint("Funkcją pierwotną funkcji f(x)=2 jest F(x)=",integrate(2, x),".")$
  Funkcją pierwotną funkcji f(x)=2 jest F(x)= 2*x .

```

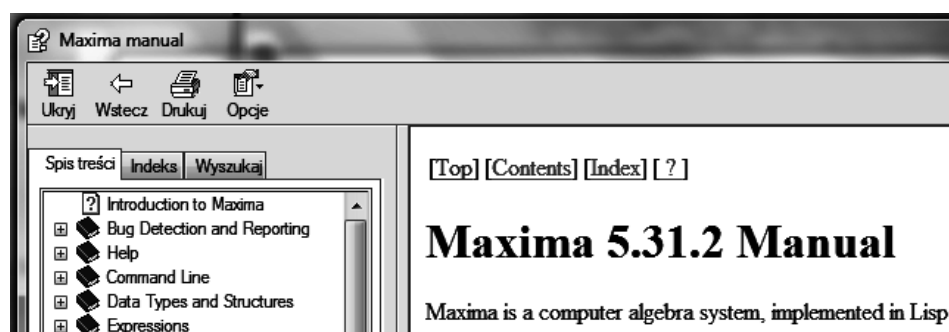
## 1.5. Korzystanie z pomocy programu

### Klawisz F1

Wciśnięcie klawisza F1 otwiera okno pomocy (Maxima Manual). Jeżeli przed naciśnięciem F1 kursor znajduje się w obrębie jakiegoś polecenia czy też opcji, wywołana zostanie pomoc kontekstowa, która otworzy podręcznik na stronie opisującej dane polecenie.

Okno pomocy możemy też otworzyć przez **Menu-Pomoc/Pomoc Maxima**.

Aby odnaleźć potrzebne słowo kluczowe, można je wyszukać tematycznie (zakładka *Spis treści*), wg nazwy (zakładka *Indeks*) lub przeszukać cały Manual (zakładka *Wyszukaj*).



Rysunek 1.15. Okno Pomocy

Tabela 1.8. Różne zapytania z poziomu notatnika Maximy

POLECENIA-POMOC	OPIS
<b>? nazwa</b>	przedstawia opis tematu o podanej nazwie
<b>?? nazwa</b>	przedstawia listę tematów zawierających podaną nazwę
<b>apropos</b> (temat)	wyświetla nazwy Maximy zawierające dany łańcuch znaków
<b>demo</b> (temat)	wyszukuje pliki demonstracyjne
<b>describe</b> (temat)	przedstawia objaśnienie wpisanego łańcucha znaków
<b>example</b> (temat)	przedstawia przykłady do danego tematu
<b>example</b> ()	wyświetla listę dostępnych tematów

Po wprowadzeniu komendy **?? nazwa** pojawia się lista tematów dotyczących tego zagadnienia.

Poniżej przykład zapytania o **package**. Należy wybrać liczbę z podanego zakresu, **a** (all) lub **n** (none).

```
(%i1) ?? package;
0: Package absimp
1: Package facexp
2: Package functs
3: Package ineq
4: Package rducon
5: Package scifac
6: Package sqdnst
7: current_let_rule_package (Functions and Variables for Rules and Patterns)
8: default_let_rule_package (Functions and Variables for Rules and Patterns)
9: let_rule_packages (Functions and Variables for Rules and Patterns)
10: packagefile (Functions and Variables for Miscellaneous Options)
Enter space-separated numbers, `all' or `none': n;
(%o1) true
```

Poniżej przykładowe zapytanie o ustawienie (opcję) **logabs**. Opcja logabs pojawi się jeszcze w rozdziale 13, przy okazji obliczania całek funkcji wymiernych.

```
(%i1) ? logabs;  
-- Option variable: logabs  
   Default value: 'false'  
   When doing indefinite integration where logs are generated, e.g.  
   'integrate(1/x,x)', the answer is given in terms of 'log(abs(...))'  
   if 'logabs' is 'true', but in terms of 'log(...)' if 'logabs' is  
   'false'. For definite integration, the 'logabs:true' setting is  
   used, because here "evaluation" of the indefinite integral at the  
   endpoints is often needed.  
(%o1) true
```