



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI



**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



Mieczysław Chalfen  
Roman Dąbrowski  
Jan Jełowicki  
Zbigniew Jurzyk  
Elżbieta Musiał  
Jolanta Szrednicka

# Kurs wyrównawczy z matematyki

dla kierunków studiów:  
Budownictwo oraz Inżynieria Środowiska

Uniwersytet Przyrodniczy we Wrocławiu  
Katedra Matematyki  
Wrocław 2009



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI



**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



---

Kurs wyrównawczy z matematyki  
dla kierunków studiów: Budownictwo oraz Inżynieria Środowiska

Uniwersytet Przyrodniczy we Wrocławiu  
Katedra Matematyki

Materiały do 50-godzinnego kursu wyrównawczego z matematyki  
przeznaczonego dla studentów kierunków przyrodniczych i technicznych  
Uniwersytetu Przyrodniczego we Wrocławiu  
w semestrze zimowym roku akademickiego 2009/2010.

Kurs przeprowadzono w ramach Programu Operacyjnego Kapitał Ludzki  
współfinansowanego ze środków Europejskiego Funduszu Społecznego.

Redakcja merytoryczna: zespół autorów

Skład (L<sup>A</sup>T<sub>E</sub>X): Jan Jełowicki

© Copyright Katedra Matematyki Uniwersytetu Przyrodniczego we Wrocławiu  
Wrocław 2009

---



# Spis treści

Wprowadzenie . . . . .	5
Oprogramowanie do kreślenia wykresów funkcji . . . . .	137





## Wprowadzenie

Przystępujemy do kursu wyrównawczego z matematyki, którego zadaniem jest pomoc studentom rozpoczynającym studia na Uniwersytecie Przyrodniczym we Wrocławiu w pokonywaniu problemów wynikających z wymagań stawianych studentom pierwszego roku studiów. Wobec zauważalnych niedoborów fachowców z wykształceniem technicznym podjęto inicjatywę mającą na celu zwiększenie liczby studentów studiujących na wybranych kierunkach studiów. Jednym z takich działań jest niniejszy kurs wyrównawczy realizowany przez naszą Uczelnię w ramach projektu *Program Operacyjny Kapitał Ludzki*.

Studentom przekazujemy materiały uzupełniające do wykładów i ćwiczeń, które z założenia traktować należy jako powtórkę materiału przerabianego w szkołach średnich w ciągu trzech lat nauki.

Należy zastrzec, że niniejsze zajęcia nie są pełnym, systematycznym kursem matematyki ze szkoły średniej. Dlatego też wiele definicji, twierdzeń, wzorów omówionych jest skrótowo na zasadzie szybkiego przypomnienia materiału znanego z dotychczasowej nauki w szkole średniej.

Zakres materiału obejmuje zagadnienia, które według naszych doświadczeń sprawiają trudności nowo przyjętym studentom, i jednocześnie opanowanie których jest niezbędne w dalszym toku studiów. Zatem kurs nie obejmuje wszystkich działów matematyki szkolnej — np. nie omawiamy stereometrii. Z drugiej strony położono nacisk na te działy matematyki — np. logarytmy, geometrię analityczną — które będą bardzo użyteczne podczas studiów, nie tylko na zajęciach z matematyki wyższej, ale także na wykładach z fizyki, mechaniki, informatyki, statystyki oraz na wielu przedmiotach kierunkowych na wyższych latach studiów.

Materiały niniejsze zostały przygotowane przez pracowników Katedry Matematyki prowadzących zajęcia na kierunkach objętych projektem.

Zajęcia prowadzone w ramach kursu przygotowawczego są obowiązkowe. Otrzymane podczas nich oceny będą uwzględniane przy zaliczaniu zajęć pierwszego semestru.

Kurs rozpoczyna się i kończy obowiązkowym sprawdzianem wiedzy. Prosimy o wypełnienie testu badającego umiejętności matematyczne uczestników kursu.

Życzymy sukcesów podczas studiów na Uniwersytecie Przyrodniczym we Wrocławiu

*autorzy*

Niniejszy dokument zawiera jedynie fragment skryptu omawiający oprogramowanie przydatne przy kreśleniu wykresów funkcji. W tekście zaktualizowano odwołania do numerów wersji oprogramowania zgodnie ze stanem z grudnia 2015 r.

*Jan Jełowicki*  
grudzień 2015 r.





# Oprogramowanie do kreślenia wykresów funkcji

Podczas analizowania zależności funkcyjnych warto posłużyć się automatycznymi systemami rysującymi wykresy funkcji. Nie wyręczą one nas w myśleniu, ale odciążą — przynajmniej częściowo — od wykonywania żmudnych rachunków, i pozwolą skupić się na istocie problemu.

## Uwaga:

Samodzielne konstruowanie wykresów funkcji w oparciu o własną wiedzę matematyczną, bez posługiwania się narzędziami pomocniczymi, jest zajęciem pożytecznym i pouczającym. Zachęcamy do podejmowania takich działań.

Przed przystąpieniem do technicznych opisów korzystania z oprogramowania trzeba zastrzec, że rysunek otrzymany za pomocą programu graficznego czy też matematycznego nie jest wykresem funkcji. W najlepszym wypadku dostaniemy coś, co jest bardzo podobne do pożądanego przez nas wykresu. Może się jednak zdarzyć, że nawet bardzo dobry program nakreśli rysunek w ogóle nie przypominający wykresu, który pragniemy otrzymać.

Jest tak z kilku powodów. Wykres funkcji  $f$  jest zbiorem punktów postaci

$$\{(x, y) : x \in X, y = f(x)\},$$

gdzie  $X$  jest dziedziną funkcji  $f$  lub jej podzbiorem. Znaczy to, że wykres funkcji określonej na przedziale osi liczbowej składa się z nieskończonej liczby punktów. Żaden system automatyczny nie jest w stanie przygotować ani przetworzyć nieskończonej liczby danych.

Dlatego przy automatycznym tworzeniu wykresu korzysta się z kilku, kilkunastu, kilkadziesiątu bądź kilkuset punktów (więcej nie ma sensu, jeżeli weźmiemy pod uwagę rozdzielczość urządzeń ekranowych), zwanych *punktami węzłowymi*, które następnie łączy się odcinkami prostych lub odcinkami ustalonego typu linii krzywych.

Jeżeli punkty węzłowe zostaną ustalone zbyt rzadko, to otrzymana linia nie będzie przypominać wykresu analizowanej funkcji, i to pomimo, iż wszystkie wartości w punktach węzłowych będą wyznaczone poprawnie.

Na koniec trzeba wspomnieć, że jeżeli nakreślony wykres ma zostać zapamiętany w pliku graficznym, trzeba zadbać, by był to plik w formacie wektorowym. Popularne formaty zapisu grafiki rastrowej: BMP, PNG, GIF, JPEG, itp. *nie nadają się* do przechowywania wykresów.

Oprogramowanie do graficznej prezentacji zależności matematycznych jest kategorią bardzo obszerną, obejmującą m.in. aplikacje typowo dydaktyczne, większość ogólnych narzędzi do graficznej prezentacji danych z arkuszami kalkulacyjnymi włącznie, aż po specjalistyczne środowiska obliczeniowe, z jakich korzystają profesjonaliści.





Wybierając oprogramowanie dla uzupełnienia niniejszych materiałów, kierowano się następującymi przesłankami:

- winno to być oprogramowanie ściśle graficzne, ukierunkowane na konstrukcję wykresów. Z tego powodu nie wzięto pod uwagę np. graficznych modułów arkuszy kalkulacyjnych, których przeznaczeniem jest prezentacja graficzna uprzednio przygotowanych tabel liczbowych;
- winno ono być możliwie proste do użycia, tzn. w typowej sytuacji winno wystarczyć krótkie pojedyncze polecenie lub skorzystanie z jednego formularza;
- powinno ono pozwalać na utrwalenie wyników w podstawowych formatach grafiki wektorowej;
- utworzony wykres winien cechować się wysoką jakością typograficzną;
- oprogramowanie winno być dostępne bez ponoszenia dodatkowych kosztów licencyjnych;
- oprogramowanie winno dać się używać niezależnie od tego, jakim systemem operacyjnym posługuje się czytelnik.

Nawet te dość restrykcyjne warunki spełniane są przez obszerną klasę programów. Ostatecznie wybrano dwie aplikacje: program graficzny o nazwie Gnuplot oraz system obliczeń matematycznych Maxima. Zostaną one krótko przedstawione poniżej.

## Gnuplot

**Charakterystyka.** Gnuplot jest ogólnodostępnym programem przeznaczonym do rysowania wykresów funkcji i prezentacji graficznej danych na dwu- i trójwymiarowych wykresach. Jego pierwsza wersja została opracowana w roku 1986, zaś w czerwcu 2015 r. wydano wersję 5.0.1.

Gnuplot może być używany nieodpłatnie do dowolnych zastosowań.

Gnuplota da się uruchomić pod nadzorem systemów operacyjnych: Windows, Linux, Mac OS, Unix/BSD i kilku innych. Pakiet instalacyjny najnowszej wersji Gnuplota można pobrać z witryny <http://www.gnuplot.info>.

Gnuplot ma bogatą dokumentację, w tym opis podręczny (*help*) w języku angielskim.

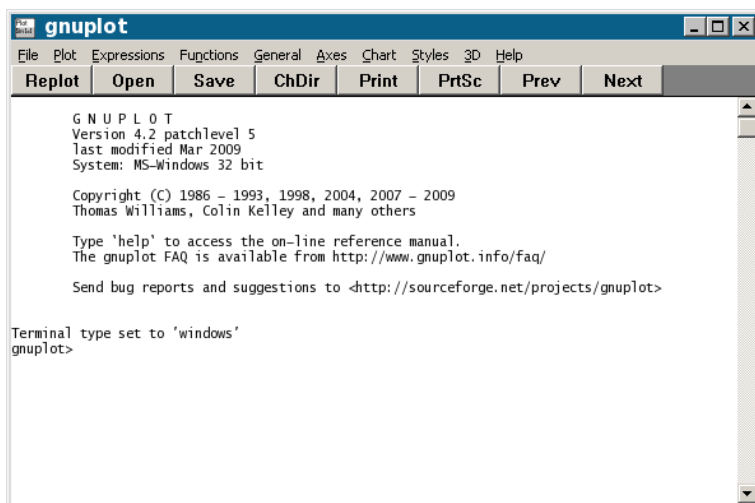
Gnuplot tworzy wykresy funkcji w oparciu o zbiór punktów węzłowych tworzących ciąg arytmetyczny. Liczbę węzłów można zadać dowolnie, jednak program sam jej nie dobiera.

**Opis użycia.** Podstawowy sposób korzystania z Gnuplota polega na pisaniu tekstowych poleceń. Wykonanie polecenia następuje po wciśnięciu klawisza <Enter>. Niektóre wersje (np. dla systemu Windows) są wyposażone w menu i przyciski pomocnicze, pozwalające wybrać czynność do wykonania zamiast wpisywać jej nazwę. Okno robocze Gnuplota przedstawiono na rysunku B.1.

Sesję programu kończy się zamykając okno lub wydając polecenie `quit`.

Podstawowym poleceniem kreślenia wykresu jest `plot`. Jego wynikiem jest osobne okno z wykresem jednej lub wielu funkcji.

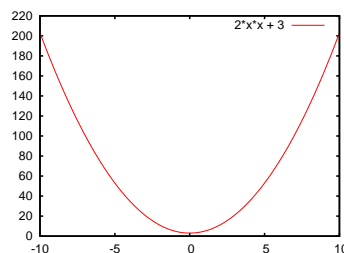
W dalszym toku przedstawiamy kilka przykładów ilustrujących podstawowe użycie Gnuplota do kreślenia wykresów funkcji matematycznych.



Rysunek B.1. Okno robocze programu GnuPlot

**Przykład 1.** Dla narysowania wykresu funkcji zadanej „wzorem matematycznym” wystarczy polecenie `plot` z deklaracją tego wzoru. Dla zmiennej niezależnej domyślnie zarezerwowany jest symbol  $x$ .

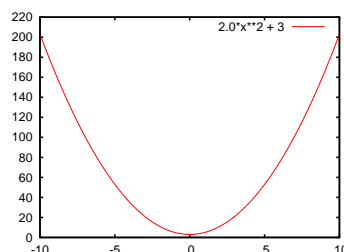
```
plot 2*x*x + 3
```



Cztery podstawowe działania arytmetyczne oznaczamy symbolami  $+$   $-$   $*$   $/$ . Dla potęgowania zarezerwowany jest symbol  $**$ . Kolejność działań jest zgodna z tradycyjną notacją; w razie potrzeby można używać nawiasów (ale tylko okrągłych). W liczbach dziesiętnych część ułamkową oddzielamy od części całkowitej znakiem kropki.

Tak więc ten sam przykład można przepisać w postaci

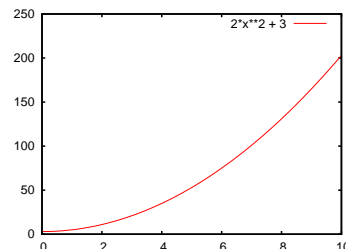
```
plot 2.0*x**2 + 3
```





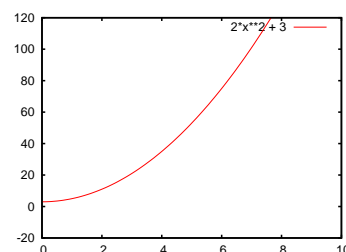
**Przykład 2.** Domyślnie oś OX obejmuje zakres od  $-10$  do  $10$ . Inny zakres wartości określamy następująco:

```
plot [x=0:10] 2*x**2 + 3
```



Podobnie wymusza się zakresy wartości dla obu osi:

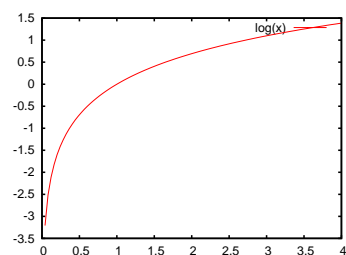
```
plot [x=0:10] [-20:120] 2*x**2 + 3
```



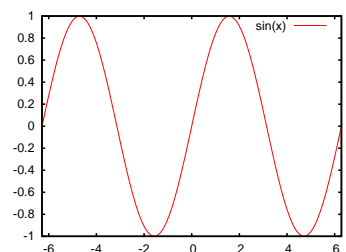
Jeżeli mamy potrzebę nadania zmiennej niezależnej innej nazwy, możemy to zrobić w następujący sposób: `plot [t=0:10] [-20:120] 2*t**2 + 3`

**Przykład 3.** Przybliżenie liczby  $\pi$  jest dostępne pod nazwą `pi`. Mamy możliwość używania najważniejszych funkcji elementarnych (uwaga: pierwiastek to `sqrt`, logarytm dziesiętny to `log10`, zaś `log` oznacza logarytm naturalny) i trygonometrycznych (uwaga: tangens nosi nazwę `tan`). Argumenty funkcji obowiązkowo umieszczamy w nawiasach okrągłych.

```
plot [x=0:4] log(x)
```



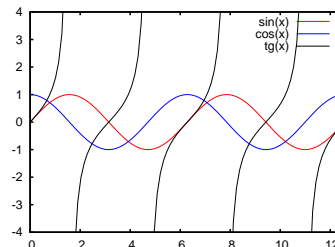
```
plot [x=-2*pi:2*pi] sin(x)
```





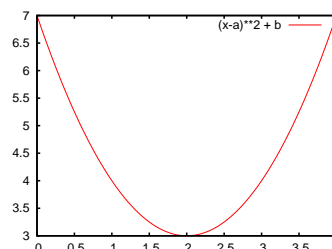
**Przykład 4.** Na jednym wykresie da się przedstawić wiele funkcji. Ich deklaracje należy oddzielać przecinkami. Opcja `title` ustala niestandardowy wpis w legendzie.

```
plot [x=0:4*pi] [-4:4] sin(x), cos(x), tan(x) title "tg(x)"
```



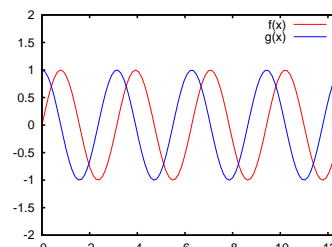
**Przykład 5.** Rysowana funkcja może zależeć od zadeklarowanego uprzednio parametru lub parametrów. Deklaracja stałej ma postać `nazwa = wartość`

```
a = 2
b = 3
plot [x=0:4] (x-a)**2 + b
```



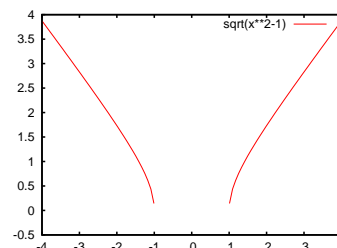
**Przykład 6.** Możemy też nadać funkcji nazwę za pomocą deklaracji `nazwa(argument) = wzór`

```
f(x) = sin(a*x)
g(x) = sin(a*x + b)
a = 2
b = pi/2
plot [x=0:4*pi] [-2:2] f(x), g(x)
```



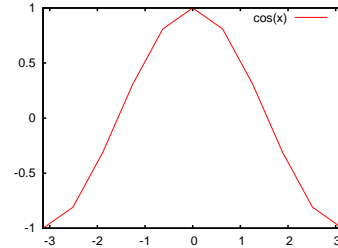
**Przykład 7.** Miłą właściwością Gnupłota jest, że w sytuacji, kiedy oś OX obejmuje obszar poza dziedziny funkcji, jest on pomijany przy rysowaniu (niektóre inne programy łączą linią punkty z obu stron takiego obszaru, zgłaszają błąd, a nawet wstawiają na wykres wartość 0).

```
plot [x=-4:4] [-0.5:4] sqrt(x**2-1)
```

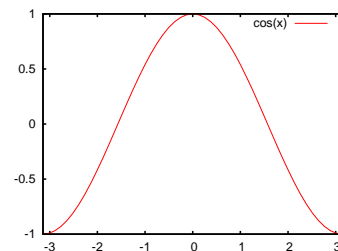


**Przykład 8.** Jeżeli punkty węzłowe są rozmieszczone zbyt rzadko, możemy wymusić ich liczbę poleceniem `set samples`.

```
set samples 11  
plot [x=-pi:pi] cos(x)
```

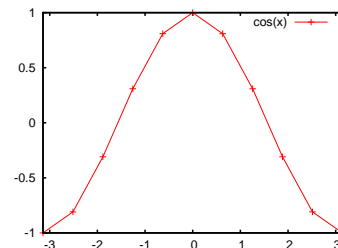


```
set samples 51  
plot [x=-pi:pi] cos(x)
```



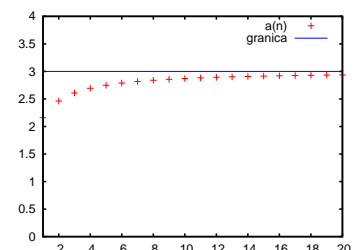
Położenie punktów węzłowych zobaczymy podając opcję `with linespoints`:

```
set samples 11  
plot [x=-pi:pi] cos(x) with linespoints
```



**Przykład 9.** Otrzymanie przy użyciu Gnupłota wykresu ciągu jest proste: wymaga doboru punktów węzłowych tak, by przypadają w kolejnych liczbach naturalnych, oraz usunięcia linii łączących węzły na wykresie za pomocą opcji `with points`.

```
m=20  
set samples m  
a(n) = (3 - 1.8/(n+1)**1.1)  
plot [n=1:m] [0:4] a(n) with points, 3 title "granica"
```



Przedstawione przykłady dotyczą jedynie najprostszych wykresów. Gnuplot dobrze się sprawdza nie tylko przy kreśleniu wykresów funkcji, ale i przy prezentacji graficznej zbiorów da-



nych — także w sytuacjach, w których moduły graficzne arkuszy kalkulacyjnych okazują się niewystarczające.

## Maxima

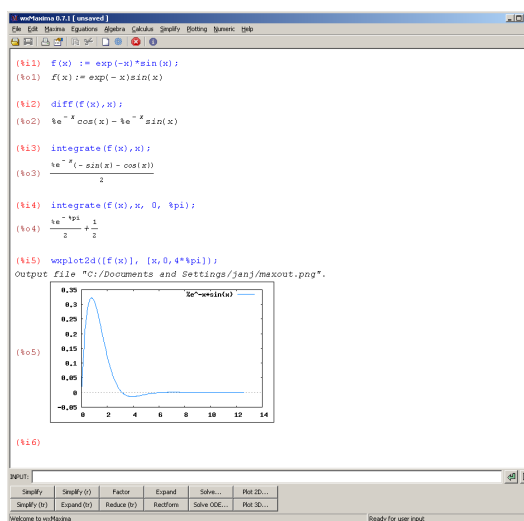
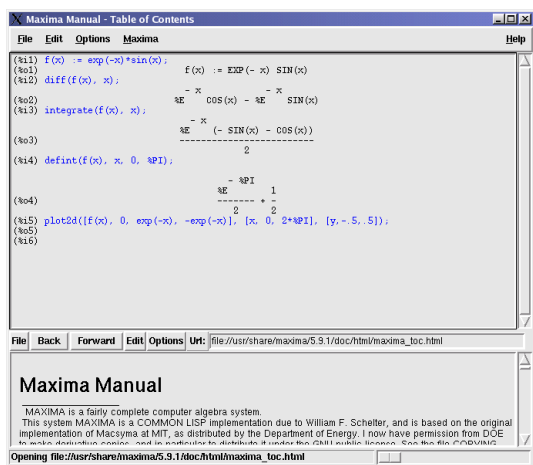
**Charakterystyka.** Maxima jest kontynuacją eksperymentalnego systemu obliczeń symbolicznych MACSYMA, tworzonych w latach 1968–1982 w Massachusetts Institute of Technology. W 1998 r. jego kod źródłowy został opublikowany na licencji otwartej pod nazwą Maxima. Wersja opatrzona numerem 5.37.3 pochodzi z listopada 2015 r.

Maxima jest systemem do przekształcania wyrażeń matematycznych. Odróżnia ją to od przeważającej większości oprogramowania inżynierskiego i naukowego, które potrafi jedynie obliczać wartości wyrażeń liczbowych. Maxima ma wbudowane moduły graficzne, które pozwalają m.in. na kreślenie wykresów funkcji jednej zmiennej.

Maximę można uruchomić pod nadzorem systemów: Windows, Linux, Mac OS X oraz Unix/BSD. Pakiety instalacyjne można pobrać z witryny <http://maxima.sourceforge.net>.

**Opis użycia.** Maxima posiada kilka interfejsów użytkowych. Ich okna są przedstawione na rysunku B.2. Najbardziej przyjazny dla użytkownika jest interfejs wxMaxima.

```
Maxima 5.9.1 http://maxima.sourceforge.net
Using Lisp CMU Common Lisp 19a
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.
(%i1) f(x) := exp(-x)*sin(x);
(%o1) f(x) := EXP(- x) SIN(x)
(%i2) diff(f(x), x);
(%o2) %E^-x COS(x) - %E^-x SIN(x)
(%i3) integrate(f(x), x);
(%o3) %E^-x (- SIN(x) - COS(x))
(%i4) defint(f(x), x, 0, %PI);
(%o4) %E^-%PI 1
      2 2
(%i5) plot2d([f(x), 0, exp(-x), -exp(-x)], [x, 0, 2*%PI], [y, -5, 5]);
(%o5)
(%i6) quit();
```



Rysunek B.2. Okna robocze interfejsów Maximy: tekstowego, xmaxima, wxMaxima

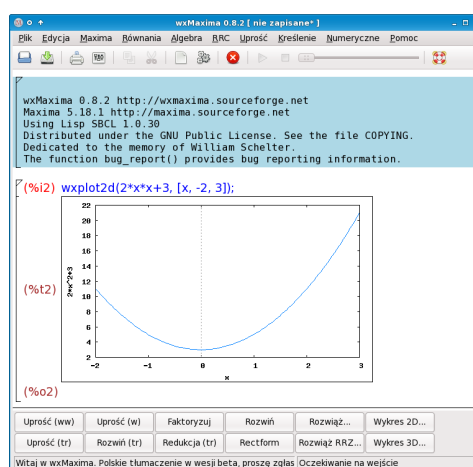
Korzystanie z Maximy polega na pisaniu poleceń. Polecenia kończymy znakiem ; (wxMaxima nie wymaga go). Wykonanie polecenia następuje po wciśnięciu klawisza <Enter> (w najnowszych wersjach programu wxMaxima <Ctrl+Enter>). Sesję kończy polecenie quit (); lub zamknięcie okna programu.

Podstawowym poleceniem kreślenia wykresu w interfejsie wxMaxima jest wxplot2d. Jego wynikiem jest wykres wstawiony do głównego okna roboczego. Podobne polecenie plot2d, dostępne także w innych interfejsach, generuje wykres w osobnym oknie.

Ciekawostką jest, że do generowania grafiki Maxima używa... programu Gnuplot (lub alternatywnie kilku innych programów).

**Przykład 10.** W najprostszym przypadku dotyczącym nakreślenia wykresu funkcji zadanej „wzorem” użyjemy polecenia

```
wxplot2d(2*x*x + 3, [x, -2, 3]);
```

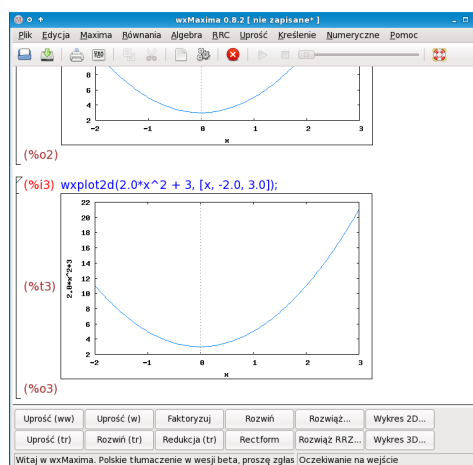


Pierwsze wyrażenie w nawiasie opisuje funkcję do narysowania, drugie — zakres osi OX. Nazwę zmiennej niezależnej można ustalić dowolnie.

Cztery podstawowe działania arytmetyczne oznaczamy symbolami + - \* /. Dla potęgowania zarezerwowany jest symbol ^ . Kolejność działań jest zgodna z tradycyjną notacją; w razie potrzeby można używać nawiasów (ale tylko okrągłych). W liczbach dziesiętnych część ułamkową oddzielamy od części całkowitej znakiem kropki.

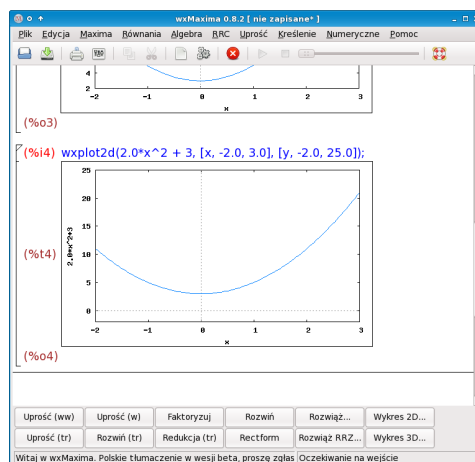
Tak więc ten sam przykład można przepisać w postaci

```
wxplot2d(2.0*x^2 + 3, [x, -2.0, 3.0]);
```



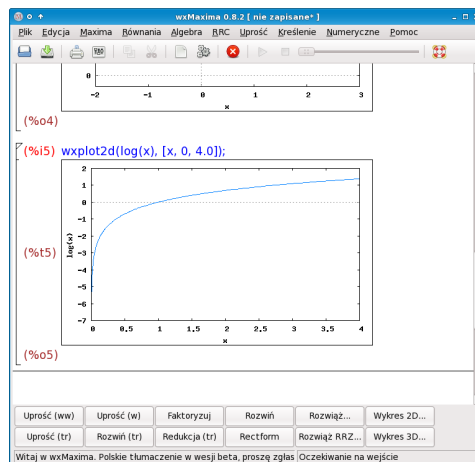
**Przykład 11.** Możemy wymusić inne niż domyślne skalowanie osi OY.

```
wxplot2d(2.0*x^2 + 3, [x, -2.0, 3.0], [y, -2.0, 25.0]);
```

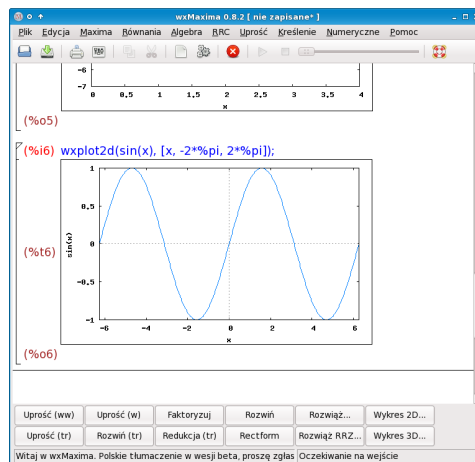


**Przykład 12.** Liczba  $\pi$  jest dostępna pod nazwą %pi. Możemy korzystać z wielu funkcji elementarnych (uwaga: pierwiastek to sqrt, log oznacza logarytm naturalny) i trygonometrycznych (uwaga: tangens nosi nazwę tan, cotangens to cot).

```
wxplot2d(log(x), [x, 0, 4]);
```



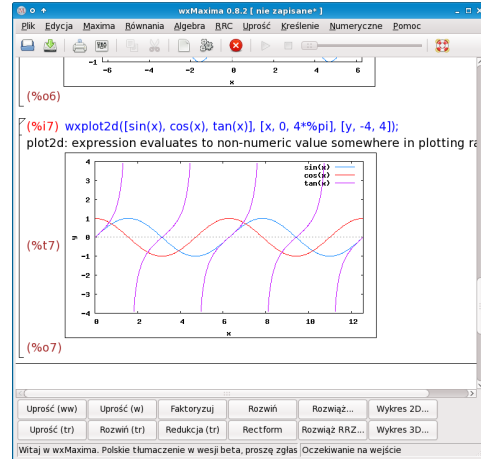
```
wxplot2d(sin(x), [x, -2*pi, 2*pi]);
```





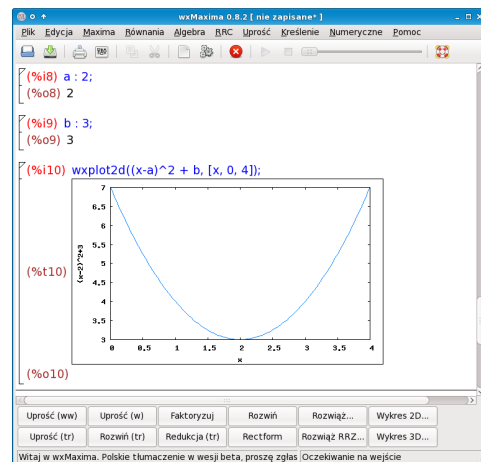
**Przykład 13.** Na jednym wykresie da się przedstawić wiele funkcji. Ich deklaracje oddzielone przecinkami trzeba objąć nawiasem kwadratowym.

```
wxplot2d([sin(x), cos(x), tan(x)], [x, 0, 4*pi], [y, -4, 4]);
```



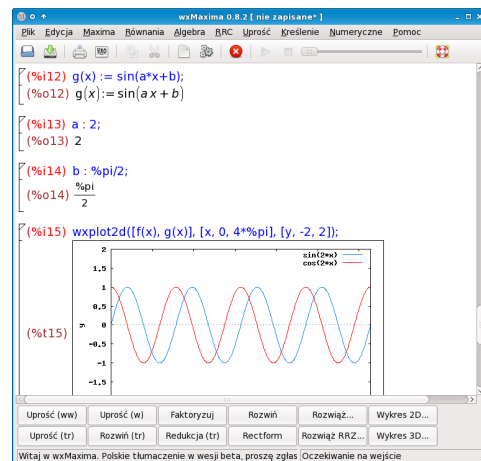
**Przykład 14.** Rysowana funkcja może zależeć od zadeklarowanego uprzednio parametru lub parametrów. Deklaracja stałej ma postać nazwa : wartość ;

```
a : 2;  
b : 3;  
wxplot2d((x-a)^2 + b, [x, 0, 4]);
```



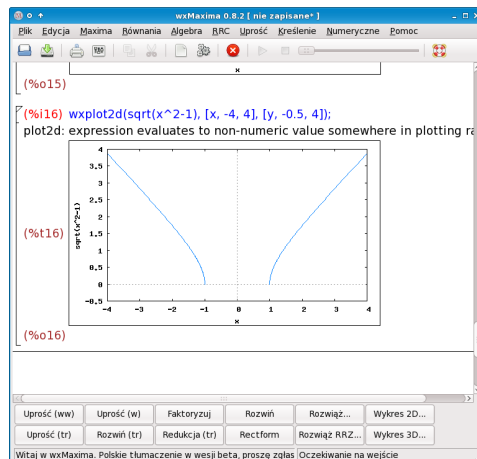
**Przykład 15.** Możemy też zadeklarować funkcję nadając jej nazwę. Deklaracja funkcji danej „wzorem” ma postać nazwa(argument) := wzór ;

```
f(x) := sin(a*x);  
g(x) := sin(a*x + b);  
a : 2;  
b : %pi/2;  
wxplot2d([f(x), g(x)], [x, 0, 4*pi], [y, -2, 2]);
```



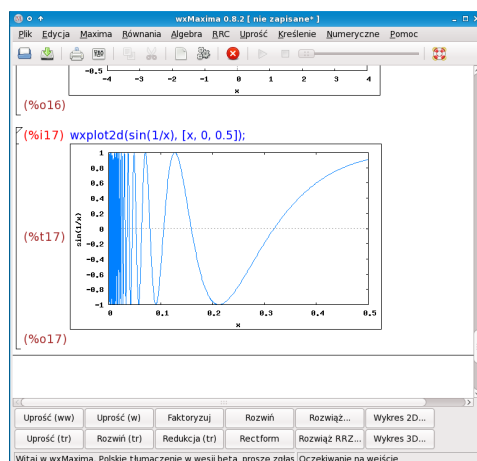
**Przykład 16.** Podobnie jak Gnuplot, także Maxima automatycznie pomija węzły przypadające poza dziedziną funkcji. Nigdy też nie łączy węzłów rozdzielonych punktem lub przedziałem nie należącym do dziedziny.

```
wxplot2d(sqrt(x^2-1), [x, -4, 4], [y, -0.5, 4]);
```



**Przykład 17.** W przeciwieństwie do Gnuplota, Maxima automatycznie wyznacza położenie punktów węzłowych, tak by zapewnić odpowiednią jakość wykresu. Odległości pomiędzy węzłami na ogół nie są jednakowe.

```
wxplot2d(sin(1/x), [x, 0, 0.5]);
```



Otrzymanie przy użyciu Maximy wykresu ciągu jest możliwe, lecz wymaga deklaracji wykraczających poza elementarny charakter tego opracowania.

Uzupełniający sposób otrzymywania wykresów polega na użyciu opcji Kreślenie/Wykres 2d... dostępnej w menu wxMaximy i wypełnieniu formularza.

Przedstawione przykłady dotyczą jedynie najprostszyc przypadków. Maxima jest narzędziem przydatnym nie tylko do kreślenia wykresów, ale przede wszystkim do przekształcania wyrażeń matematycznych.