

Matematyka wyższa z Maximą

Wojciech Młócek

Uniwersytet Rolniczy w Krakowie
Katedra Zastosowań Matematyki
`kzm.ur.krakow.pl`

© Copyright by Wojciech Młócek

Kraków 2006 - 2008

Spis treści

Wprowadzenie	3
1. Podstawowe informacje	4
1.1. Instalacja programu	4
1.2. Pierwsze kroki	4
2. Opis wybranych funkcji	8
2.1. Wyrażenia liczbowe, algebraiczne i trygonometryczne	8
2.2. Funkcje matematyczne	9
2.3. Definiowanie funkcji	10
2.4. Wielomiany, funkcje wymierne	11
2.5. Równania, układy równań	11
2.6. Liczby zespolone	12
2.7. Ciągi, szeregi, granice funkcji	13
2.8. Wykresy	14
2.9. Pochodne i całki	16
2.10. Szereg Taylora	17
2.11. Wektory	18
2.12. Macierze	19
2.13. Równania różniczkowe	20
3. Przykłady	23
3.1. Zadania	23
3.2. Rozwiązania	26
Literatura	43
Skorowidz	44

Wprowadzenie

Maxima jest programem komputerowym typu CAS (Computer Algebra System – system algebry komputerowej), wspomagającym wykonywanie obliczeń matematycznych, zarówno symbolicznych, jak i numerycznych. Zastosowanie rachunku symbolicznego pozwala na rozwiązywanie wielu problemów matematycznych w sposób dokładny.

Na przykład licząc całkę oznaczoną $\int_1^2 \cos x dx$, po wpisaniu `integrate(cos(x),x,1,2)`, otrzymamy rozwiązanie dokładne postaci `sin(2)-sin(1)`.

Program ten wywodzi się z opracowanego w Massachusetts Institute of Technology pod koniec lat 60-tych na zlecenie Departamentu Energii USA programu Macsyma, a umożliwia on m.in.

- wykonywanie obliczeń numerycznych z dowolną dokładnością,
- upraszczanie wyrażeń algebraicznych i trygonometrycznych,
- symboliczne rozwiązywanie równań (w tym różniczkowych),
- symboliczne rozwiązywanie układów równań,
- różniczkowanie i całkowanie symboliczne,
- operacje na macierzach,
- rysowanie wykresów funkcji,
- wykonywanie obliczeń z zakresu rachunku prawdopodobieństwa i statystyki matematycznej,
- definiowanie własnych funkcji przez użytkownika,
- programowanie w Lispie,
- eksport otrzymanych wyników do formatów HTML, LaTeX oraz formatów graficznych PNG, PostScript.

Zajmujący się od 1982 roku Maximą William Schelter uzyskał w roku 1998 od Departamentu Energii zezwolenie na uwolnienie kodu na bezpłatnej licencji GNU (General Public License). Od tego czasu Maxima rozwija się dość intensywnie w wersjach dla systemów Windows oraz Linux, a prace nad programem skupiają się na jego udoskonalaniu i usuwaniu błędów zgłaszanych przez użytkowników.

Od wersji 5.10 program zawiera już zintegrowaną nakładkę wxMaxima (graficzny interfejs), która zapewnia lepsze formatowanie formuł matematycznych oraz ułatwia ich wprowadzanie i modyfikowanie.

Maxima nie posiada własnej grafiki, ale jest dystrybuowana razem z Gnuplotem, popularnym programem do tworzenia wykresów w dwu i trzech wymiarach, z którym współpracuje w sposób automatyczny.

Opis wszystkich funkcji programu znajdziemy w podręcznej pomocy, gdzie zawarto również przykłady ich zastosowań. Wiele cennych informacji znajdziemy też na oficjalnej stronie programu maxima.sourceforge.net.

Opanowanie podstaw programu nie wymaga od nas specjalnie dużej ilości czasu, za to korzyści z tego płynące są czasem nie do przecenienia. Oczywiście należy pamiętać o tym, że Maxima nie uwolni nas od myślenia matematycznego, nie zastąpi nam znajomości teorii matematycznej. Może jednak w wielu przypadkach, poprzez eksperymentowanie, rozważanie wielu alternatyw, wizualizację, ułatwić jej lepsze i szybsze zrozumienie. Pomaga też oszczędzić czas oraz kontrolować i eliminować błędy przy rozwiązywaniu skomplikowanych zadań rachunkowych.

1. Podstawowe informacje

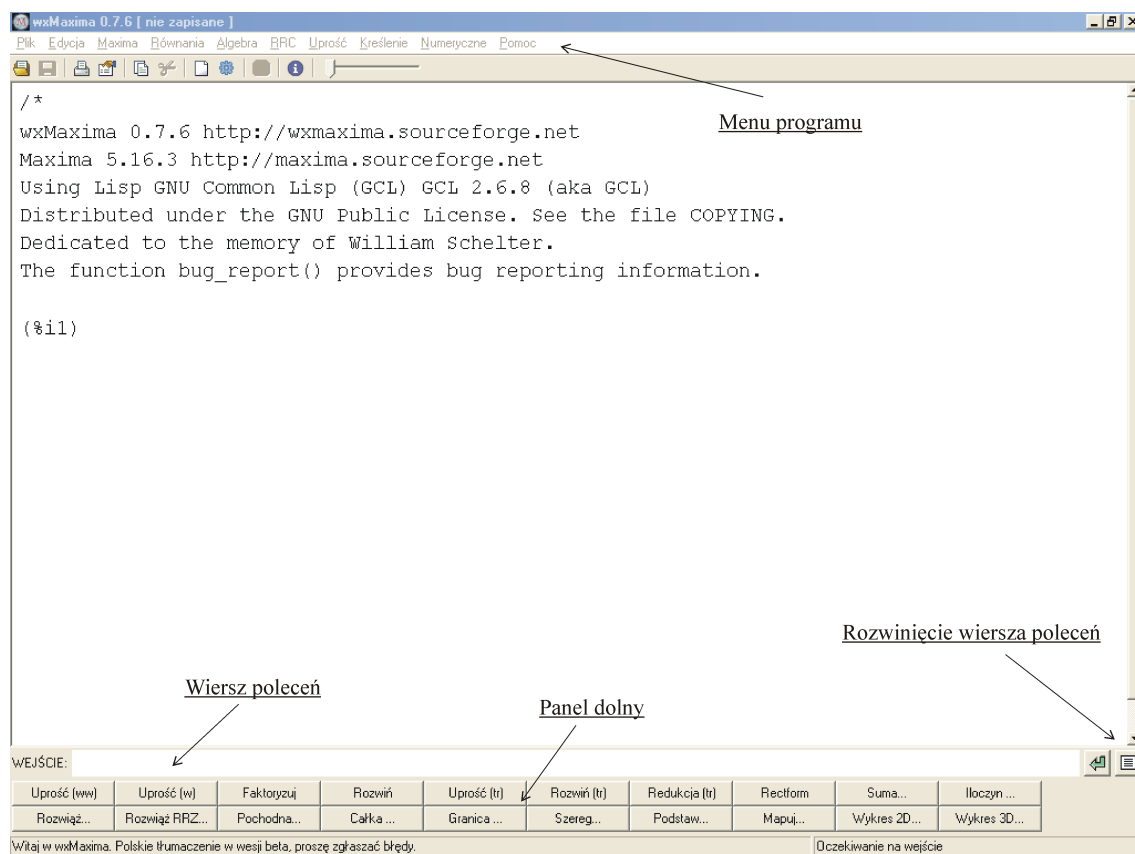
1.1. Instalacja programu

Plik instalacyjny `maxima-5.16.3.exe`¹ (dla systemu Windows lub Linux) pobieramy ze strony maxima.sourceforge.net. Plik ten znajdziemy w dziale Download. Po pobraniu uruchamiamy plik instalacyjny i podążamy za wskazówkami instalatora. Instalacja powinna przebiegać bez żadnych komplikacji. Po zakończeniu instalacji program uruchamiamy wybierając wxMaxima.

1.2. Pierwsze kroki

Przed przystąpieniem do pracy warto rozszerzyć panel dolny do wersji pełnej. W tym celu wybieramy kolejno **Edycja/Preferencje/Dolny panel/Zaawansowany**. By panel został rozszerzony trzeba uruchomić program na nowo.

Okno programu powinno wyglądać teraz następująco:



Polecenia do programu wpisujemy do wiersza poleceń. Wpisaną komendę akceptujemy klawiszem **Enter**. Wiele z dostępnych funkcji programu znajdziemy w menu programu oraz w panelu dolnym. Maxima rozróżnia wielkość liter, stąd wszystkie polecenia wpisujemy zawsze małymi literami. Każde polecenie kończymy znakiem `;` lub `$`. Znak `;` informuje program o tym, aby

¹ najnowsza wersja programu na dzień 26.09.2008 r.

wykonał podaną instrukcję, a wynik wyświetlił na ekranie (znak ten jest wstawiany automatycznie). Natomiast znak \$ oznacza, że instrukcja ma być tylko wykonana, bez wyświetlania rezultatu. Oto przykład:

```
(%i1) 3+4;
(%o1) 7
(%i2) 9-2$
(%i3) 2*8;
(%o3) 16
```

Program akceptuje standardowe operatory arytmetyczne:

operator	znaczenie
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
^ lub **	potęgowanie

Na przykład, aby poznać wartość wyrażenia $\frac{6^4-16}{9^2-1}$ wpiszemy:

```
(%i1) (6^4-16)/(9^2-1);
(%o1) 16
```

Dla porządku podajmy tutaj również spis operatorów porównania oraz operatorów logicznych:

operator	znaczenie
=	jest równe
#	nie jest równe
>	jest większe
<	jest mniejsze
>=	jest większe lub równe
<=	jest mniejsze lub równe
and	logiczne i
or	logiczne lub
not	zaprzeczenie

Jeżeli chcemy się powołać na wyrażenie² użyte wcześniej wpisujemy numer wiersza, w którym się ono znajduje (np %o1). Znak % reprezentuje zawsze ostatni wynik, co obrazuje poniższy przykład:

```
(%i1) 16/2;
(%o1) 8
(%i2) (%)^2;
(%o2) 64
```

Wprowadzane wyrażenia możemy nazywać korzystając z operatora przypisania „:”, oto przykład:

```
(%i1) x:3;
(%o1) 3
(%i2) 4*x;
```

² tak będziemy nazywać zarówno wyrażenia liczbowe, jak i algebraiczne

```
(%o2) 12
```

By zlikwidować znaczenie x korzystamy z funkcji `kill`:

```
(%i3) kill(x);
```

```
(%o3) done
```

```
(%i4) 6*x;
```

```
(%o4) 6x
```

Jeżeli chcemy dokonać jednocześnie kilku przypisań korzystamy z funkcji `block`, oto przykład:

```
(%i1) block(a:2,b:4,c:9)$
```

```
(%i2) b;
```

```
(%o2) 4
```

Likwidację znaczenia wszystkich zmiennych uzyskamy wpisując `kill(all)` lub wybierając z menu Maxima/Wyczyść pamięć.

Należy pamiętać o tym, że jeśli podajemy liczbę w postaci dziesiętnej, miejsce dziesiętne rozdzielane jest kropką, to jest ona traktowana zawsze jako wartość przybliżona i wówczas obliczenia są wykonywane numerycznie. Zilustrujmy to na przykładzie:

```
(%i1) 3/7;
```

```
(%o1)  $\frac{3}{7}$ 
```

```
(%i2) 3.0/7;
```

```
(%o2) 0.42857142857143
```

By otrzymać wartość wyrażenia liczbowego w postaci dziesiętnej wpisujemy `float(wyrażenie)` lub `wyrażenie,numer`, oto przykład:

```
(%i1) 1/3+1/7+1/12;
```

```
(%o1)  $\frac{47}{84}$ 
```

```
(%i2) float(%);
```

```
(%o2) 0.55952380952381
```

```
(%i3) 1/3+1/7+1/12,numer;
```

```
(%o3) 0.55952380952381
```

Komenda `numer:true` sprawi, że wszystkie kolejne obliczenia będą wykonywane numerycznie, oto przykład:

```
(%i1) numer:true$
```

```
(%i2) 2/9+3/14;
```

```
(%o2) 0.43650793650794
```

```
(%i3) 5/7-1/8+6/11;
```

```
(%o3) 1.13474025974026
```

```
(%i4) numer:false$
```

```
(%i5) 2/9+3/14;
```

```
(%o5)  $\frac{55}{126}$ 
```

Obliczenia numeryczne w Maximie możemy wykonywać z dowolnieadaną dokładnością, po komendzie `fpprec:n` będą one wykonywane z dokładnością do n -cyfr znaczących (standardowe ustawienie `fpprec:16`). Możemy też zmieniać dokładność wyświetlanych wyników, poprzez wpisanie `fpprintprec:n`, oto przykład:

```
(%i1) float(23/1121);
```

```
(%o1) 0.020517395182872
```

```
(%i2) fpprintprec:4$
```

```
(%i3) float(23/1121);
```

```
(%o3) 0.021
```

W Maximie zdefiniowano domyślnie kilka stałych, oto one:

stała	wartość
%pi	π
%e	e
%i	i
minf	$-\infty$
inf	$+\infty$

Dowolne stałe, pojawiające się w rozwiązaniu, na przykład równania różniczkowego, poprzedza zawsze znak %.

Warto tutaj powiedzieć jeszcze o apostrofie ', który napisany przed dowolnym operatorem blokuje jego wykonanie, oto przykład:

(%i1) a:4\$

(%i2) a^2;

(%o2) 16

(%i3) 'a^2;

(%o3) a²

Opis innych funkcji programu Maxima znajdziemy w rozdziale następnym.

2. Opis wybranych funkcji

Maxima jest programem bardzo obszernym, stąd nie sposób przedstawić jego wszystkich potencjalnych możliwości. Przedstawiamy tutaj opis wybranych funkcji, których znajomość może być szczególnie użyteczna dla studentów kierunków technicznych.

Przy uruchomieniu programu, ze względu na ich ogromną liczbę, nie wszystkie funkcje ładowane są do pamięci operacyjnej komputera. Rzadziej używane zostały zawarte w tzw. pakietach. Każdy pakiet zawiera zbiór funkcji związany z konkretnym działem matematyki. Pakiety ładujemy do programu wpisując `load(nazwa_pakietu)`.³

W celu uzyskania opisu wybranej funkcji wystarczy wpisać `describe(funkcja)`. By uzyskać przykłady użycia danej funkcji wpisujemy `example(funkcja)`.

Pamiętajmy o tym, że przy wprowadzaniu funkcji nie zawsze konieczne jest ręczne wpisywanie jej formuły, bowiem wiele z nich znajdziemy w panelu dolnym oraz w menu programu.

2.1. Wyrażenia liczbowe, algebraiczne i trygonometryczne

Maxima posiada wiele funkcji, które umożliwiają modyfikowanie, upraszczanie oraz rozwijanie wyrażeń. Przedstawmy wybrane z nich:

- `ev(wyrażenie,warunek)` – modyfikuje wyrażenie na podstawie warunku,
- `subst(b,a,wyrażenie)` – po tej komendzie zmienna a w wyrażeniu zostanie zastąpiona przez zmienną b ,
- `expand(wyrażenie)` – rozwija wyrażenie algebraiczne,
- `ratsimp(wyrażenie)` – upraszcza wyrażenie algebraiczne,
- `radcan(wyrażenie)` – upraszcza wyrażenie,
- `factor(wyrażenie)` – rozkłada wyrażenie na czynniki,
- `trigexpand(wyrażenie)` – rozwija wyrażenie trygonometryczne,
- `trigreduce(wyrażenie)`, `trigsimp(wyrażenie)` – upraszcza wyrażenie trygonometryczne (polecenie `halfangles:true` spowoduje, że Maxima będzie wykorzystywać związki między funkcjami kątów połówkowych).

Dla przykładu weźmy wyrażenie $w = (x - a)(x + 2a)$:

```
(%i1) w:(x-a)*(x+2*a);
```

```
(%o1) (x - a)(x + 2a)
```

By rozwinąć to wyrażenie wpiszemy:

```
(%i2) expand(w);
```

```
(%o2) x2 + ax - 2a2
```

jeśli teraz rozłożymy na czynniki ostatni wynik otrzymamy:

```
(%i3) factor(%);
```

```
(%o3) (x - a)(x + 2a)
```

Jeżeli chcemy poznać wartość wyrażenia w dla $a = 1$, $x = 2$ wpiszemy:

```
(%i4) ev(w, a=1, x=2);
```

```
(%o4) 4
```

Aby uprościć wyrażenie $\frac{a-b^2}{\sqrt{a+b}}$ wpiszemy:

```
(%i5) radcan((a-b^2)/(sqrt(a)+b));
```

```
(%o5)  $\sqrt{a} - b$ 
```

³ jeżeli jakaś funkcja będzie wymagała załadowania pakietu, wówczas będziemy to sygnalizować

2.2. Funkcje matematyczne

Wśród wszystkich funkcji programu szczególne miejsce zajmują funkcje matematyczne. W poniższej tabeli zebraliśmy najbardziej popularne z nich:

funkcja	zapis matematyczny
entier(x)	$[x]$
signum(x)	$\operatorname{sgn} x$
sqrt(x)	\sqrt{x}
abs(x)	$ x $
exp(x)	e^x
a^x	a^x
log(x)	$\ln x$
log(x)/log(a)	$\log_a x$
sin(x)	$\sin x$
cos(x)	$\cos x$
tan(x)	$\operatorname{tg} x$
cot(x)	$\operatorname{ctg} x$
asin(x)	$\arcsin x$
acos(x)	$\arccos x$
atan(x)	$\operatorname{arc} \operatorname{tg} x$
acot(x)	$\operatorname{arc} \operatorname{ctg} x$
sinh(x)	$\sinh x$
cosh(x)	$\cosh x$
tanh(x)	$\operatorname{tgh} x$
coth(x)	$\operatorname{ctgh} x$
asinh(x)	$\operatorname{arcsinh} x$
acosh(x)	$\operatorname{arccosh} x$
atanh(x)	$\operatorname{arctgh} x$
acoth(x)	$\operatorname{arcctgh} x$

Powiedzmy jeszcze w tym miejscu o dwóch funkcjach:

- $\max(x_1, \dots, x_n)$ – podaje największą liczbę ze zbioru $\{x_1, \dots, x_n\}$,
- $\min(x_1, \dots, x_n)$ – podaje najmniejszą liczbę ze zbioru $\{x_1, \dots, x_n\}$.

Oto przykład:

```
(%i1) max(0,-1,2,-3,4);
(%o2) 4
(%i2) min(0,-1,2,-3,4);
(%o2) -3
```

2.3. Definiowanie funkcji

Pokażemy teraz na konkretnych przykładach, jak w Maximie zdefiniować funkcję w zależności od jej postaci.

Funkcję

$$f(x) = x^3 + 2$$

definiujemy następująco:

$$f(x) := x^3 + 2$$

Dla funkcji

$$g(x) = \int_1^{x^2} t^3 dt$$

wpisujemy następujące polecenie:

```
define(g(x),integrate(t^3,t,1,x^2))
```

Funkcję

$$h(x) = \begin{cases} x & \text{dla } x < -2, \\ x^2 + 2 & \text{dla } -2 \leq x \leq 1, \\ 3 & \text{dla } x > 1. \end{cases}$$

definiujemy korzystając z wyrażenia warunkowego `if`⁴ w następujący sposób:

```
h(x):=if x<-2 then x else if x>=-2 and x<=1 then x^2+2 else 3
```

Funkcję dwu zmiennych

$$f(x, y) = xy - xe^y$$

możemy zdefiniować poprzez polecenie:

```
f(x,y):=x*y-x*exp(y)
```

Mając zdefiniowaną funkcję możemy obliczyć jej wartość dla dowolnego argumentu z dziedziny, oto przykład:

```
(%i1) f(x,y):=x*y-x*exp(y)$
```

```
(%i2) f(2,1);
```

```
(%o2) 2-2*e
```

Warto powiedzieć jeszcze o funkcji `map(f, [wyrażenie])`, której działanie polega na wykonaniu operacji funkcyjnej na każdym argumencie wyrażenia, oto przykład:

```
(%i1) f(x):=1/x$
```

```
(%i2) map(f, [3,a,1/a]);
```

```
(%o2) [1/3, 1/a, a]
```

Przydatnym poleceniem jest też `depends(y,x)`, po wpisaniu którego y będzie traktowane jako funkcja zmiennej x .

⁴ `if <warunek> then <instrukcja> else <instrukcja>`

2.4. Wielomiany, funkcje wymierne

Dla wielomianów oraz funkcji wymiernych szczególnie przydatne są następujące funkcje:

- `divide(P,Q)` – dzieli wielomian P przez wielomian Q , a wynik podaje w postaci [część całkowita, reszta],
- `realroots(wielomian)` – znajduje pierwiastki rzeczywiste wielomianu,⁵
- `allroots(wielomian)` – znajduje pierwiastki zespolone wielomianu,
- `partfrac(f(x),x)` – rozkłada funkcję wymierną f na sumę wielomianu i ułamków prostych.⁶

Dla przykładu znajdziemy pierwiastki rzeczywiste wielomianu $W(x) = x^4 - 5x^3 + 7x^2 - 5x + 6$:

```
(%i1) W:x^4-5*x^3+7*x^2-5*x+6;
```

```
(%o1) x^4 - 5x^3 + 7x^2 - 5x + 6
```

```
(%i2) realroots(W);
```

```
(%o2) [x=2, x=3]
```

Jeżeli chcemy poznać krotności tych pierwiastków wpiszemy:

```
(%i3) multiplicities;
```

```
(%o3) [1,1]
```

Aby rozłożyć funkcję wymierną $f(x) = \frac{x^3 - 3x^2 + 3x - 5}{x^2 - 5x + 6}$ na sumę wielomianu oraz ułamków prostych wpiszemy:

```
(%i1) f:(x^3-3*x^2+3*x-5)/(x^2-5*x+6);
```

```
(%o1) 
$$\frac{x^3 - 3x^2 + 3x - 5}{x^2 - 5x + 6}$$

```

```
(%i2) partfrac(f,x);
```

```
(%o2) 
$$x + \frac{3}{x-2} + \frac{4}{x-3} + 2$$

```

2.5. Równania, układy równań

Przedstawmy funkcje przydatne do rozwiązywania równań oraz układów równań:

- `lhs(równanie)` – zwraca lewą stronę równania,
- `rhs(równanie)` – zwraca prawą stronę równania,
- `solve(równanie,zmienna)` – rozwiązuje równanie w zbiorze liczb zespolonych,⁷
- `linsolve([równania],[zmienne])` – rozwiązuje liniowy układ równań,
- `algsys([równania],[zmienne])` – rozwiązuje algebraiczny układ równań w zbiorze \mathbb{C}^n , jeżeli chcemy otrzymać rozwiązania w zbiorze \mathbb{R}^n , najpierw wpisujemy `realonly:true`,
- `find_root(f(x),x,a,b)` – znajduje, poprzez aproksymację, miejsca zerowe funkcji f w przedziale $[a,b]$ (przedział tak dobieramy, aby $f(a)$, $f(b)$ były różnych znaków).

Na przykład, aby znaleźć rozwiązanie układu równań

$$\begin{cases} xy - 2x = 0 \\ x^3 - x^2y + x - y = 0 \end{cases}$$

w zbiorze \mathbb{R}^2 wpiszemy:

```
(%i1) realonly:true$
```

```
(%i2) r1:x*y-2*x=0$
```

```
(%i3) r2:x^3-x^2*y+x-y=0$
```

⁵ krotności pierwiastków zwraca funkcja `multiplicities`

⁶ przydatne przy liczeniu całek funkcji wymiernych

⁷ krotności tych rozwiązań, podobnie jak przy funkcji `realroots`, zwraca funkcja `multiplicities`

```
(%i4) algsys([r1,r2],[x,y]);
(%o4) [[x=0,y=0],[x=2,y=2]]
```

Wiemy, że równanie $\left(\frac{1}{2}\right)^x = x$ posiada dokładnie jedno rozwiązanie, co widać po naszkicowaniu wykresów funkcji: $f(x) = \left(\frac{1}{2}\right)^x$, $g(x) = x$. Można jednak podać tylko jego przybliżoną wartość. Korzystając z funkcji `find_root` znajdziemy w Maximie to rozwiązanie. W tym celu przyjmijmy $h(x) = \left(\frac{1}{2}\right)^x - x$:

```
(%i1) h(x):=(1/2)^x-x$
```

Obliczmy teraz wartości funkcji h w punktach $a = 0$, $b = 2$:

```
(%i2) [h(0),h(2)];
```

```
(%o2) [1,-7/4]
```

Zatem $h(0)$, $h(2)$ są różnych znaków, stąd po wpisaniu:

```
(%i3) find_root(h(x),x,0,2);
```

otrzymamy szukane rozwiązanie:

```
(%o3) 0.64118574450499
```

2.6. Liczby zespolone

W zakresie działań na liczbach zespolonych przydatnymi funkcjami w Maximie są:

- `cabs(z)` – oblicza moduł liczby zespolonej z ,
- `realpart(z)` – podaje część rzeczywistą liczby zespolonej z ,
- `imagpart(z)` – podaje część urojoną liczby zespolonej z ,
- `carg(z)` – podaje argument liczby zespolonej z należący do przedziału $(-\pi, \pi]$,
- `conjugate(z)` – podaje sprzężenie liczby zespolonej z ,
- `rectform(z)` – podaje liczbę zespoloną z w postaci $a + ib$,
- `polarform(z)` – podaje liczbę zespoloną z w postaci $|z|e^{i\theta}$, gdzie $\theta \in \arg z$.

Dla przykładu weźmy liczbę zespoloną $z = 1 - i\sqrt{3}$:

```
(%i1) z:1-i*sqrt(3);
```

```
(%o1) 1 - sqrt(3)i
```

```
(%i2) cabs(z);
```

```
(%o2) 2
```

```
(%i3) realpart(z);
```

```
(%o3) 1
```

```
(%i4) imagpart(z);
```

```
(%o4) -sqrt(3)
```

```
(%i5) carg(z);
```

```
(%o5) -pi/3
```

```
(%i6) conjugate(z);
```

```
(%o6) sqrt(3)i + 1
```

```
(%i7) polarform(z);
```

```
(%o7) 2e^(-i*pi/3)
```

Aby zilustrować działanie funkcji `rectform` weźmy $z = \frac{(3+i)^2}{2-i}$:

```
(%i1) z:(3+i)^2/(2-i);
```

```
(%o1) (i+3)^2/(2-i)
```

```
(%i2) rectform(z);
(%o2) 4%i + 2
```

2.7. Ciągi, szeregi, granice funkcji

Poniższa tabela przedstawia wybrane funkcje Maxima dotyczące ciągów, szeregów oraz granic funkcji:

funkcja	zapis matematyczny
<code>limit(a_n,n,inf)</code>	$\lim_{n \rightarrow \infty} a_n$
<code>sum(a_n,n,1,k), simpsum</code>	$\sum_{n=1}^k a_n$
<code>sum(a_n,n,1,inf), simpsum</code>	$\sum_{n=1}^{\infty} a_n$
<code>product(a_n,n,1,k), simpproduct</code>	$\prod_{n=1}^k a_n$
<code>limit(f(x),x,x₀)</code>	$\lim_{x \rightarrow x_0} f(x)$
<code>limit(f(x),x,x₀,minus)</code>	$\lim_{x \rightarrow x_0^-} f(x)$
<code>limit(f(x),x,x₀,plus)</code>	$\lim_{x \rightarrow x_0^+} f(x)$

Jeżeli granica ciągu lub funkcji nie istnieje, wówczas Maxima zwraca `und` lub `ind`. W odpowiedzi otrzymamy `ind`, jeżeli ciąg lub funkcja są ograniczone.

Oto przykład:

```
(%i1) limit(1/x, x, 0);
(%o1) und
(%i2) limit(sin(1/x), x, 0);
(%o2) ind
```

2.8. Wykresy

Wykresy w Maximie tworzone są za pomocą programu Gnuplot⁸, z którym Maxima współpracuje automatycznie. Dokumentację wraz z przykładami do tego programu znajdziemy na stronie www.gnuplot.info.

Przedstawmy podstawowe funkcje służące do rysowania wykresów:

- `plot2d([f(x)], [x,a,b])` – rysuje wykres funkcji $f(x)$ w przedziale $[a, b]$,
- `plot2d([f1(x), ..., fn(x)], [x,a,b])` – rysuje wykresy funkcji $f_1(x), \dots, f_n(x)$ w przedziale $[a, b]$ na jednym rysunku,
- `plot2d([parametric,x(t),y(t),[t,c,d]]], [nticks,300])` – rysuje wykres funkcji zadanej parametrycznie: $(x(t), y(t))$, $t \in [c, d]$,
- `plot3d([f(x,y)], [x,a,b], [y,c,d])` – rysuje wykres funkcji $f(x, y)$ dla $(x, y) \in [a, b] \times [c, d]$,
- `plot3d([x(t,s), y(t,s), z(t,s)], [t,a,b], [s,c,d])` – rysuje wykres powierzchni zadanej parametrycznie: $(x(t, s), y(t, s), z(t, s))$, $t \in [a, b]$, $s \in [c, d]$,
- `plot2d([discrete,x,y], [style,[points,l,m,n]])`⁹ lub `plot2d([discrete,xy], [style,[points,l,m,n]])`¹⁰ – tworzy wykres punktowy dla punktów postaci $(x_1, y_1), \dots, (x_n, y_n)$.

Aby otrzymać wykres liniowy należy `[points,l,m,n]` zastąpić przez `[lines,l,m]`.

Parametry $l, m, n \in \mathbb{N}$ odpowiadają odpowiednio za grubość, kolor oraz rodzaj znacznika.

Po wprowadzeniu jednej z powyższych funkcji otrzymamy odpowiedni wykres w nowym oknie. Jeżeli chcemy, by wykres pojawił się w oknie Maximy, wtedy wystarczy zamiast `plot2d`, `plot3d` wpisać odpowiednio `wxplot2d`, `wxplot3d`.

W celu sformatowania wykresu na własny użytek należy do powyższych funkcji wprowadzić dodatkowe polecenia. Przedstawmy niektóre z nich:

- `[gnuplot_preamble, "set zeroaxis"]` – dodaje osie układu współrzędnych,
- `[xlabel, "opis"], [ylabel, "opis"]` – zmienia opis osi układu współrzędnych,
- `[gnuplot_preamble, "set nokey"]` – usuwa legendę,
- `[legend, "tytuł"]` lub `[gnuplot_curve_titles, ["title 'tytuł'"]]` – zmienia legendę,
- `[style,[lines,l,m]]` lub `[gnuplot_curve_styles, ["with lines n"]]` – zmienia grubość oraz kolor linii ($l, m \in \mathbb{N}$),
- `[gnuplot_term, ps], [gnuplot_out_file, "plot.eps"]` – zapisuje wykres do pliku graficznego z rozszerzeniem eps¹¹ pod nazwą `plot.eps`,
- `[gnuplot_term, png], [gnuplot_out_file, "plot.png"]` – zapisuje wykres do pliku graficznego z rozszerzeniem png pod nazwą `plot.png`.

W Maximie możemy również za pomocą funkcji `implicit_plot`¹² rysować wykresy krzywych o równaniach uwikłanych.

W celu narysowania krzywej uwikłanej $F(x, y) = 0$ dla $(x, y) \in (a, b) \times (c, d)$ wpiszemy:

```
implicit_plot(F(x,y)=0, [x,a,b], [y,c,d])
```

To samo polecenie `implicit_plot` może posłużyć do otrzymania wykresów kilku krzywych uwikłanych na tym samym obrazku:

```
implicit_plot([F1(x,y)=0, F2(x,y)=0, ..., Fn(x,y)=0], [x,a,b], [y,c,d])
```

⁸ jest on instalowany razem z Maximą, plik startowy znajdziemy w katalogu `../Maxima/bin/`

⁹ $x : [x_1, x_2, \dots, x_n]$, $y : [y_1, y_2, \dots, y_n]$

¹⁰ $xy : [[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]]$

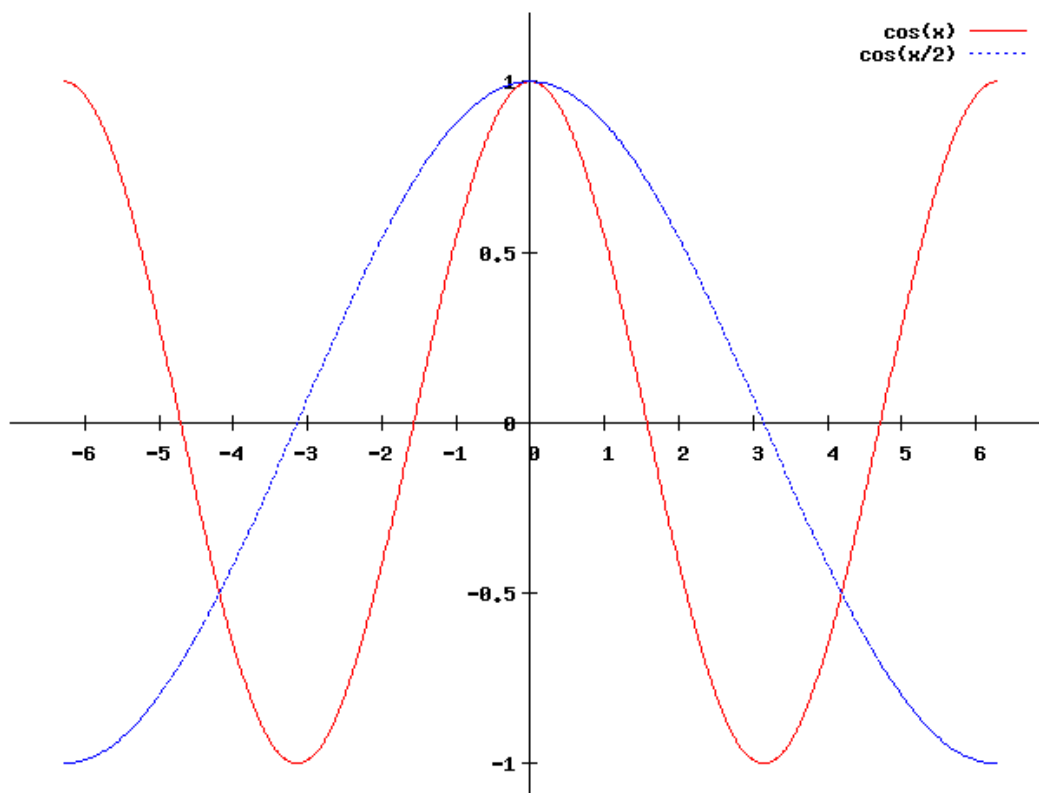
¹¹ pliki z tym rozszerzeniem można przeglądać m.in. w przeglądarce GSview

¹² działa po załadowaniu pakietu `implicit_plot`

Rozszerzeniem podstawowych funkcji związanych z tworzeniem wykresów w Maximize jest pakiet `draw`. Opis tego pakietu wraz z przykładami Czytelnik znajdzie w pomocy programu.

Dla przykładu, korzystając z pakietu `draw`, narysujemy na jednym obrazku wykresy funkcji $\cos x$ i $\cos \frac{x}{2}$ w przedziale $[-2\pi, 2\pi]$:

```
(%i1) load(draw)$
(%i2) wxdraw2d(xrange=[-7,7], yrange=[-1.1,1.2],
  axis_top=false, axis_bottom=false,
  axis_left=false, axis_right=false, xaxis=true, yaxis=true,
  xtics_axis=true, ytics_axis=true, xtics=[-6,1,6], ytics=[-1,1/2,1],
  xaxis_type=solid, yaxis_type=solid, nticks=100,
  color=red, line_type=solid, key="cos(x)", explicit(cos(x),x,-2*%pi,2*%pi),
  color=blue, line_type=dots, key="cos(x/2)",
  explicit(cos(x/2),x,-2*%pi,2*%pi))$
(%t2)
```



2.9. Pochodne i całki

Poniżej przedstawiamy wybrane funkcje Maximy dotyczące rachunku różniczkowego i całkowego:

funkcja	zapis matematyczny
<code>diff(f(x),x)</code>	$f'(x)$
<code>diff(f(x),x,n)</code>	$f^{(n)}(x)$
<code>integrate(f(x),x)</code>	$\int f(x)dx$
<code>integrate(f(x),x,a,b)</code> <code>romberg(f(x),x,a,b)^{a)}</code>	$\int_a^b f(x)dx$
<code>diff(f(x,y),x)</code>	$\frac{\partial f}{\partial x}(x,y)$
<code>diff(f(x,y),y)</code>	$\frac{\partial f}{\partial y}(x,y)$
<code>diff(f(x,y),x,2)</code>	$\frac{\partial^2 f}{\partial x^2}(x,y)$
<code>diff(f(x,y),x,1,y,1)</code>	$\frac{\partial^2 f}{\partial x \partial y}(x,y)$
<code>diff(f(x,y),y,2)</code>	$\frac{\partial^2 f}{\partial y^2}(x,y)$
<code>diff(f(x,y),x,k,y,n)</code>	$\frac{\partial^{k+n} f}{\partial x^k \partial y^n}(x,y)$

^{a)} oblicza całkę numerycznie metodą Romberga

Na przykład weźmy funkcję $f(x) = x^2 \sin(3x)$ i obliczmy jej pierwszą oraz drugą pochodną:

```
(%i1) f(x):=x^2*sin(3*x)$
(%i2) diff(f(x), x);
(%o2) 2xsin(3x) + 3x^2cos(3x)
(%i3) diff(f(x), x, 2);
(%o3) -9x^2sin(3x) + 2sin(3x) + 12xcos(3x)
```

W przypadku, gdy nie jest określona zmienna względem której liczymy pochodną funkcji, wówczas jest ona reprezentowana przez funkcję `del`, oto przykład:

```
(%i1) diff(log(f));
(%o1)  $\frac{\text{del}(f)}{f}$ 
```

Dzięki funkcji `changevar` możemy dokonywać podstawienia pod całką, jak również obliczać całki przez podstawienie.

Dla przykładu weźmy całkę $\int_1^4 \sqrt{x-1} dx$:

– by ją obliczyć wpiszemy:

```
(%i1) integrate(sqrt(x-1), x, 1, 4);
(%o1)  $2\sqrt{3}$ 
```

– by dokonać podstawienia $t = x - 1$ wpiszemy:

```
(%i1) changevar('integrate(sqrt(x-1), x, 1, 4), x-1-t, t, x);
(%o1)  $\int_0^3 \sqrt{t} dt$ 
```

– by ją obliczyć przez podstawienie $t = x - 1$ wpiszemy:

```
(%i1) changevar(integrate(sqrt(x-1), x, 1, 4), x-1-t, t, x);
(%o1)  $2\sqrt{3}$ 
```


2.10. Szereg Taylora

Z kursu analizy matematycznej wiemy, że przy pewnych założeniach wartości funkcji możemy przybliżać wielomianami. Przybliżenie to uzyskujemy po rozwinięciu funkcji zgodnie ze wzorem Taylora. Dla funkcji jednej zmiennej pierwszych k wyrazów tego rozwinięcia wokół punktu x_0 przedstawia się następująco:

$$f(x) \approx \sum_{n=0}^k \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n$$

By uzyskać to rozwinięcie w Maximie wpisujemy:

```
taylor(f(x), x, x0, k)
```

Na przykład znajdziemy pierwsze cztery wyrazy wzoru Taylora dla funkcji $f(x) = x^x$ w otoczeniu punktu $x_0 = 1$:

```
(%i1) taylor(x^x, x, 1, 4);
```

```
(%o1) 1 + (x - 1) + (x - 1)^2 + (x - 1)^3/2 + (x - 1)^4/3 + ...
```

Pełne rozwinięcie tworzy szereg Taylora¹³

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n,$$

który w Maximie otrzymamy po wpisaniu:

```
niceindices(powerseries(f(x), x, x0))14
```

Na przykład znajdziemy szereg Taylora dla funkcji $f(x) = e^x$ w punkcie $x_0 = 0$:

```
(%i1) niceindices(powerseries(exp(x), x, 0));
```

```
(%o1) sum_{i=0}^inf x^i/i!
```

Rozwinięcie funkcji dwóch zmiennych zgodnie ze wzorem Taylora do wyrazów rzędu $(p - 1)$ włącznie wokół punktu (x_0, y_0) przedstawia się następująco:

$$f(x_0 + h, y_0 + k) \approx f(x_0, y_0) + \frac{1}{1!} d^1(x_0, y_0) + \frac{1}{2!} d^2(x_0, y_0) + \dots + \frac{1}{(p-1)!} d^{p-1}(x_0, y_0).$$

W Maximie uzyskamy to rozwinięcie po wprowadzeniu polecenia:

```
taylor(f(x,y), [x,y], [x0,y0], p-1)$ ev(%, [x=h+x0,y=k+y0])
```

Na przykład rozwiniemy zgodnie ze wzorem Taylora do wyrazów rzędu szóstego włącznie funkcję $f(x, y) = x \sin y$ w otoczeniu punktu $(1, \pi)$:

```
(%i1) taylor(x*sin(y), [x,y], [1,%pi], 6)$ ev(%, [x=h+1,y=k+%pi]);
```

```
(%i2)
```

```
(%o2) -(h+1)k^5 + (-20h-20)k^3 + (120h+120)k/120
```

```
(%i3) expand(%)
```

```
(%o3) -hk^5/120 - k^5/120 + hk^3/6 + k^3/6 - hk - k
```

¹³ szereg Taylora zawsze możemy napisać, jednak nie zawsze funkcja jest rozwijalna w szereg Taylora

¹⁴ niceindices pozwala uzyskać indeksy postaci i, j, k, l, m, n

2.11. Wektory

W Maximie możemy również wykonywać operacje na wektorach. Przyjmijmy, że $\vec{x} = [x_1, x_2, x_3]$, $\vec{y} = [y_1, y_2, y_3]$, $\vec{z} = [z_1, z_2, z_3]$, funkcjami przydatnymi są:

funkcja	zapis matematyczny
<code>x: [x₁, x₂, x₃]</code>	\vec{x}
<code>x ± y</code>	$\vec{x} \pm \vec{y}$
<code>sqrt(x.x)</code>	$ \vec{x} $
<code>uvect(x)^{a)}</code>	$\frac{\vec{x}}{ \vec{x} }$
<code>x.y</code> lub <code>inprod(x,y)^{a)}</code>	$\vec{x} \circ \vec{y}$ (iloczyn skalarny)
<code>express(x~y)^{b)}</code>	$\vec{x} \times \vec{y}$ (iloczyn wektorowy)
<code>express((x~y).z)^{b)}</code>	$(\vec{x} \times \vec{y}) \circ \vec{z}$ (iloczyn mieszany)

^{a)} działa po załadowaniu pakietu `eigen`

^{b)} działa po załadowaniu pakietu `vect`

Niech na przykład $\vec{x} = [-1, 2, 1]$, $\vec{y} = [3, 0, 2]$, $\vec{z} = [4, 1, -2]$:

```
(%i1) x: [-1, 2, 1]$
```

```
(%i2) y: [3, 0, 2]$
```

```
(%i3) z: [4, 1, -2]$
```

– obliczmy długość wektora \vec{x} :

```
(%i4) sqrt(x.x);
```

```
(%o4)  $\sqrt{6}$ 
```

– obliczmy iloczyn skalarny $\vec{x} \circ \vec{y}$:

```
(%i5) x.y;
```

```
(%o5) -1
```

– obliczmy iloczyn wektorowy $\vec{y} \times \vec{z}$:

```
(%i6) load(vect)$
```

```
(%i7) express(y~z);
```

```
(%o7) [-2, 14, 3]
```

– obliczmy iloczyn mieszany $(\vec{x} \times \vec{y}) \circ \vec{z}$:

```
(%i8) express((x~y).z);
```

```
(%o8) 33
```

Mając dany wektor możemy w łatwy sposób powoływać się na jego składowe, oto przykład:

```
(%i1) x: [-2, 1, 4, 3]$
```

```
(%i2) x[3];
```

```
(%o2) 4
```

Przydatnymi są również funkcje `lmin`, `lmax`, które wśród składowych wektora znajdują wartość najmniejszą oraz największą odpowiednio, oto przykład:

```
(%i1) x: [3, -1, 0, 2]$
```

```
(%i2) lmin(x);
```

```
(%o2) -1
```

```
(%i3) lmax(x);
```

```
(%o3) 3
```

2.12. Macierze

W Maximie dostępnych jest szereg funkcji związanych z macierzami. Macierze wprowadzamy do programu korzystając z funkcji `matrix`¹⁵, oto przykład:

```
(%i1) matrix([2,-1], [1,3]);
```

```
(%o1)  $\begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$ 
```

Możemy również użyć funkcji `entermatrix`, której działanie obrazuje poniższy przykład:

```
(%i1) entermatrix(2,2);
```

Is the matrix 1. Diagonal 2. Symmetric 3. Antisymmetric 4. General

Answer 1, 2, 3 or 4 : 4;

Row 1 Column 1: 2;

Row 1 Column 2: -1;

Row 2 Column 1: 1;

Row 2 Column 2: 3;

Matrix entered.

```
(%o1)  $\begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$ 
```

Aby wprowadzić macierz identycznościową stopnia n wystarczy wpisać `ident(n)`, oto przykład:

```
(%i2) ident(2);
```

```
(%o2)  $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ 
```

Poniższa tabela przedstawia podstawowe działania na macierzach:

funkcja	zapis matematyczny
$A \pm B$	$A \pm B$
$\alpha * A$	$\alpha \cdot A$
$A.B$	$A \cdot B$

Niech $A = \begin{bmatrix} 1 & 2 \\ 3 & -1 \end{bmatrix}$, $B = \begin{bmatrix} 4 & 2 \\ 1 & 0 \end{bmatrix}$. Znajdźmy macierz $A \cdot B - 2A$:

```
(%i1) A:matrix([1,2], [3,-1])$
```

```
(%i2) B:matrix([4,2], [1,0])$
```

```
(%i3) A.B-2*A;
```

```
(%o3)  $\begin{bmatrix} 4 & -2 \\ 5 & 8 \end{bmatrix}$ 
```

Ponadto funkcjami przydatnymi są:

- `A[i,j]` – zwraca element a_{ij} macierzy A ,
- `col(A,k)` – zwraca k -tą kolumnę macierzy A ,
- `row(A,k)` – zwraca k -ty wiersz macierzy A ,
- `transpose(A)` – transponuje macierz A ,
- `determinant(A)` – oblicza wyznacznik macierzy kwadratowej A ,
- `rank(A)` – wyznacza rząd macierzy A ,
- `invert(A)` – znajduje macierz odwrotną do macierzy kwadratowej A ,
- `newdet(A,n)` – oblicza minor główny rzędu n macierzy kwadratowej A ,

¹⁵ dostępna jest ona w menu, wybieramy **Algebra/Wprowadź macierz**

- `ncharpoly(A,x)`¹⁶ – znajduje wielomian charakterystyczny macierzy kwadratowej A ,
- `eigenvalues(A)` – znajduje wartości własne macierzy kwadratowej A wraz z ich krotnościami, a wynik podaje w postaci `[[wartości własne],[krotności]]`,
- `eigenvectors(A)` – znajduje wartości własne macierzy kwadratowej A wraz z ich krotnościami oraz odpowiadające im wektory własne, a wynik podaje w postaci `[[[wartości własne],[krotności]],wektory własne]`,
- `wronskian([f1(x),...,fn(x)],x)`¹⁷ – zwraca macierz Wrońskiego dla funkcji $f_1(x), \dots, f_n(x)$.

2.13. Równania różniczkowe

W zakresie równań różniczkowych Maxima potrafi między innymi:

- rozwiązywać równania różniczkowe zwyczajne rzędu pierwszego;

$$F(x, y(x), y'(x)) = 0$$

- rozwiązywać równania różniczkowe zwyczajne rzędu drugiego;

$$F(x, y(x), y'(x), y''(x)) = 0,$$

- rozwiązywać układy dwóch¹⁸ równań różniczkowych rzędu pierwszego;

$$\frac{dy}{dx} = f(x, y, z), \quad \frac{dz}{dx} = g(x, y, z).$$

Oto lista przydatnych funkcji (oznaczenia jw.):

- `ode2(równanie,y,x)` – podaje rozwiązanie ogólne równania różniczkowego rzędu pierwszego lub drugiego,
- `ic1(rozwiazanie ogólne,x=x0,y=y0)` – podaje rozwiązanie szczególne równania różniczkowego rzędu pierwszego spełniające warunek $y(x_0) = y_0$,
- `ic2(rozwiazanie ogólne,x=x0,y=y0,diff(y,x)=y1)` – podaje rozwiązanie szczególne równania różniczkowego rzędu drugiego spełniające warunki: $y(x_0) = y_0$, $y'(x_0) = y_1$,
- `bc2(rozwiazanie ogólne,x=x1,y=y1,x=x2,y=y2)` – podaje rozwiązanie szczególne równania różniczkowego rzędu drugiego spełniające warunki: $y(x_1) = y_1$, $y(x_2) = y_2$,
- `desolve([równanie1,równanie2],[y(x),z(x)])` – podaje rozwiązanie ogólne układu dwóch równań różniczkowych.

W celu uzyskania rozwiązania szczególnego, spełniającego warunki $y(x_0) = y_0$, $z(x_0) = z_0$, należy najpierw wpisać: `atvalue(y(x),x=x0,y0)` oraz `atvalue(z(x),x=x0,z0)`, a dopiero później skorzystać z funkcji `desolve`.

Dla przykładu weźmy równanie różniczkowe:

$$\frac{dy}{dx} - y = 2xe^x.$$

Pokażemy najpierw, jak wprowadzić to równanie do Maximy. Możemy to zrobić na trzy sposoby:

- pierwszy sposób;

```
(%i1) 'diff(y,x)-y=2*x*exp(x);
```

```
(%o1)  $\frac{d}{dx}y - y = 2x e^x$ 
```

¹⁶ działa po załadowaniu pakietu `nchrpl`

¹⁷ działa po załadowaniu pakietu `functs`

¹⁸ lub więcej

- drugi sposób;


```
(%i1) depends(y,x)$
(%i2) diff(y,x)-y=2*x*exp(x);
(%o2)  $\frac{d}{dx}y - y = 2x e^x$ 
```
- trzeci sposób;


```
(%i1) diff(y(x),x)-y(x)=2*x*exp(x);
(%o1)  $\frac{d}{dx}y(x) - y(x) = 2x e^x$ 
```

Rozwiążemy teraz to równanie:

```
(%i1) r:'diff(y,x)-y=2*x*exp(x)$
(%i2) ode2(r, y, x);
(%o2) y=(x^2+%c)%e^x
```

Jeżeli Maxima potrafi rozwiązać równanie różniczkowe funkcja `method` podaje nam metodę¹⁹, którą program wykorzystał do rozwiązania tego równania:

```
(%i3) method;
(%o3) linear
```

Metody wykorzystywane w Maximie przy rozwiązywaniu równań różniczkowych:

- pierwszego rzędu:
 - ✓ `linear` – liniowe,
 - ✓ `separable` – o rozdzielonych zmiennych,
 - ✓ `exact` – zupełne, jeżeli jest to równanie zupełne z czynnikiem całkującym, to po wpisaniu `intfactor` otrzymamy postać czynnika całkującego,
 - ✓ `homogeneous` – jednorodne,
 - ✓ `bernoulli` – Bernoulliego, po wpisaniu `odeindex` otrzymamy rząd tego równania,
- drugiego rzędu:
 - ✓ `constcoeff` – liniowe o stałych współczynnikach,
 - ✓ `variationofparameters` – liniowe, jeżeli jest to równanie liniowe niejednorodne, to po wpisaniu `yp` otrzymamy rozwiązanie szczególne,
 - ✓ `freeofy` – niezależące jawnie od y ,
 - ✓ `freeofx` – niezależące jawnie od x ,
 - ✓ `exact` – zupełne,
 - ✓ `euler` – Eulera,
 - ✓ `bessel` – Bessel’a.

Dla przykładu rozwiążemy równanie:

$$y^2 + (yx - 1)\frac{dy}{dx} = 0, \quad y(1) = 1$$

```
(%i1) r:y^2+(y*x-1)*'diff(y,x)=0;
(%o1) (xy-1) $\left(\frac{d}{dx}y\right)$ +y^2=0
(%i2) rozw:ode2(r,y,x);
(%o2) xy-log(y)=%c
(%i3) ic1(rozw,x=1,y=1);
(%o3) xy-log(y)=1
(%i4) method;
(%o4) exact
```

¹⁹ metoda pokrywa się z typem równania różniczkowego

```
(%i5) intfactor;
```

```
(%o5) 1/y
```

oraz układ równań:

$$\frac{dy}{dx} = z + 1, \quad \frac{dz}{dx} = y + x, \quad y(0) = z(0) = 1$$

```
(%i1) r1:diff(y(x),x)=z(x)+1;
```

```
(%o1) dy/dx = z(x) + 1
```

```
(%i2) r2:diff(z(x),x)=y(x)+x;
```

```
(%o2) dz/dx = y(x) + x
```

```
(%i3) atvalue(y(x),x=0,1)$
```

```
(%i4) atvalue(z(x),x=0,1)$
```

```
(%i5) desolve([r1,r2],[y(x),z(x)]);
```

```
(%o5) [y(x) = 2%e^x - %e^-x - x, z(x) = 2%e^x + %e^-x - 2]
```

Jeżeli Maxima nie potrafi rozwiązać równania różniczkowego, jako wynik zwraca **false**, oto przykład:

```
(%i1) 'diff(y,x)=sin(x-y);
```

```
(%o1) d/dx y = -sin(y - x)
```

```
(%i2) ode2(%, y, x);
```

```
(%o2) false
```

Rozszerzeniem funkcji `ode2` jest pakiet `contrib_ode`, w którym zawarto bardziej zaawansowane metody wykorzystywane przy rozwiązywaniu równań różniczkowych.

3. Przykłady

Rozdział ten zawiera przykłady zastosowań Maximy do rozwiązywania zadań z zakresu podstawowego kursu matematyki prowadzonego na uczelniach technicznych (analiza, algebra, równania różniczkowe). Przy rozwiązywaniu zadań zakładamy znajomość teorii matematycznej, obejmującej ten właśnie kurs oraz znajomość funkcji programu opisanych w poprzednim rozdziale. Komentarze dotyczące rozwiązań zawarliśmy tylko tam, gdzie jest to konieczne.

3.1. Zadania

1. Obliczyć $(2 - \sqrt{3})^4$.
2. Sprowadzić do najprostszej postaci wyrażenie $(a^{-1} - b^{-1}) \frac{\sqrt{a} + \sqrt{b}}{\sqrt{b} - \sqrt{a}} - \frac{2}{\sqrt{ab}}$.
3. Rozwiązać w zbiorze liczb rzeczywistych równanie $x^5 - 2x^4 - 4x^3 + 4x^2 - 5x + 6 = 0$.
4. Rozwiązać w zbiorze liczb zespolonych równanie $x^5 - 2x^4 - 4x^3 + 4x^2 - 5x + 6 = 0$.
5. Znaleźć wszystkie pierwiastki wielomianu $W(x) = x^5 + x^2 + 1$.
6. Sprowadzić do postaci iloczynowej wielomian $W(x) = 2x^5 + 3x^4 - x^3 + x^2 - 3x - 2$.
7. Wykonać dzielenie wielomianu $P(x) = x^4 - 3x^2 + 2$ przez wielomian $Q(x) = x^3 + 2x^2 - x - 4$.
8. Rozłożyć funkcję wymierną
$$f(x) = \frac{x^3 - x^2 - x - 2}{x^4 + 4x^3 - x^2 - 16x - 12}$$
na sumę ułamków prostych.
9. Rozwiązać układ równań
$$\begin{cases} 2x^2 + xy - 4 = 0 \\ 2y^2 + xy - 10 = 0. \end{cases}$$
10. Rozwiązać równanie $2^{2x} - 5 \cdot 2^x + 4 = 0$.
11. Rozwiązać równanie $e^x = x + 2$.
12. Obliczyć granicę ciągu $a_n = \left(\frac{n+2}{n-3}\right)^{2n}$.
13. Wyznaczyć sumę szeregu $\sum_{n=1}^{\infty} \frac{1}{n^2}$.
14. Wykonać działania $\frac{(4-i)^2}{2+3i}$.
15. Znaleźć postać trygonometryczną liczby zespolonej $z = 1 - i\sqrt{3}$.
16. Obliczyć $(1 - \sqrt{3}i)^{13}$.
17. Obliczyć granicę $\lim_{x \rightarrow 0^-} (1 - e^{3x}) \operatorname{ctg} x$.
18. Obliczyć granicę $\lim_{x \rightarrow +\infty} \frac{x^2 + 2}{e^{2x} - 1}$.
19. Wyznaczyć granicę $\lim_{x \rightarrow 2} \frac{1}{x-2}$.
20. Wyznaczyć granicę $\lim_{x \rightarrow 1} x \sin \frac{1}{x-1}$.
21. Znaleźć funkcję odwrotną do funkcji $f(x) = \frac{x+3}{x-5} + 2$.
22. Narysować wykres funkcji $f(x) = 3 \sin^2 x - 2 \sin x - 2$ w przedziale $[-6, 6]$.
23. Obliczyć pochodną funkcji $f(x) = \ln(\operatorname{arc} \operatorname{tg}(x^2 - 1))$.
24. Niech $f(x) = \sin x \cos^3(2x)$. Obliczyć $f'(\frac{\pi}{3})$.
25. Niech $f(x) = \sin(x^3)$. Obliczyć f' , f'' , f''' .

26. Niech $f(x) = x \sin x$. Narysować na jednym rysunku wykresy funkcji f , f' , f'' w przedziale $[-7, 7]$.
27. Rozwinąć funkcję $f(x) = \ln(1+x)$ w szereg Taylora w otoczeniu punktu $x_0 = 0$.
28. Znaleźć pięć pierwszych wyrazów wzoru Taylora funkcji $f(x) = x\sqrt{x}$ w punkcie $x_0 = 1$.
29. Znaleźć równanie stycznej do wykresu funkcji $f(x) = \sqrt{2x} \cos(x-2)$ w punkcie $x_0 = 2$.
30. Wyznaczyć ekstrema lokalne i zbadać monotoniczność funkcji $f(x) = xe^{-x^2}$.
31. Wyznaczyć punkty przegięcia i zbadać wklęsłość/wypukłość funkcji $f(x) = xe^{-x^2}$.
32. Znaleźć wartość najmniejszą i największą funkcji $f(x) = (x^3 - 2x^2 + 2x - 2)e^x$ w przedziale $[-3, 3]$.
33. Obliczyć całkę nieoznaczoną $\int x^2 \arctg x dx$.
34. Obliczyć całkę oznaczoną $\int_0^{\frac{\pi}{2}} \cos^3 x dx$.
35. Obliczyć całkę oznaczoną $\int_0^4 \arctg^2 x dx$.
36. Obliczyć całkę niewłaściwą $\int_{-\infty}^{+\infty} \frac{dx}{x^2+16}$.
37. Obliczyć całkę niewłaściwą $\int_1^e \frac{dx}{x \ln x}$.
38. Obliczyć pole obszaru ograniczonego wykresami funkcji: $f(x) = 2x^2 - 12x + 10$, $g(x) = 2x - 2$.
39. Narysować wykres krzywej (rozety ośmiolistnej) danej równaniami:
 $x(t) = 2 \sin(4t) \cos t$, $y(t) = 2 \sin(4t) \sin t$, $t \in [0, 2\pi]$.
40. Obliczyć długość krzywej $y = \sqrt{x+2}$, $-2 \leq x \leq 0$.
41. Narysować wykres krzywej (spirali Archimedesesa) zadanej biegunowo: $r = t$, $t \in [0, 4\pi]$.
42. Narysować wykres krzywej danej równaniami:

$$x = \sin t \left(e^{\cos t} - 2 \cos(4t) + \sin^5\left(\frac{t}{12}\right) \right), \quad y = \cos t \left(e^{\cos t} - 2 \cos(4t) + \sin^5\left(\frac{t}{12}\right) \right)$$

dla $t \in [0, 24\pi]$.

43. Niech $A = \begin{bmatrix} 1 & 4 & -1 & 3 \\ 3 & -1 & 2 & 5 \\ -1 & 3 & 1 & 0 \\ -3 & 5 & 2 & 1 \end{bmatrix}$. Obliczyć wyznacznik macierzy A .
44. Niech $A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 2 & 2 & 4 & 6 \\ 1 & 1 & 2 & 3 \end{bmatrix}$. Wyznaczyć rząd macierzy A .
45. Niech $A = \begin{bmatrix} 1 & 0 & 0 \\ 3 & -1 & 0 \\ -1 & 2 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 2 & 0 \\ 0 & -1 & 3 \\ 0 & 1 & 1 \end{bmatrix}$. Znaleźć macierz $A^{-1} \cdot B^T$.
46. Niech $A = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix}$, $B = \begin{bmatrix} 2 & -1 \\ 0 & 3 \\ 4 & 1 \end{bmatrix}$, $C = \begin{bmatrix} 4 & 2 & 0 \\ 1 & -1 & 1 \end{bmatrix}$.

Znaleźć macierz $2A^{-1} + 3C \cdot B - 4A^T$.

47. Rozwiązać poniższy układ równań:

$$\begin{cases} 5x_1 + 2x_2 - x_3 = 4 \\ 3x_1 + x_2 - 2x_3 = 1 \\ 2x_1 + 3x_2 + 3x_3 = 5 \end{cases}.$$

48. Rozwiązać poniższy układ równań:

$$\begin{cases} x + 2y + z + t = 2 \\ -2x - 3y + z - t = 0 \\ -x - y + 2z = 2 \end{cases}.$$

49. Rozwiązać poniższy układ równań:

$$\begin{cases} x_1 + x_2 - x_3 = 2 \\ 2x_1 + x_2 + x_3 = 0 \\ -x_1 - x_2 + x_3 = 1 \end{cases}.$$

50. Posługując się definicją liniowej niezależności wektorów, sprawdzić, czy wektory $u_1 = (1, 3, 0, -2)$, $u_2 = (2, 1, 2, 3)$, $u_3 = (-1, 0, 4, 1)$, $u_4 = (0, 2, -2, 1)$ są liniowo niezależne.
51. Narysować wykres funkcji $f(x, y) = \sin(x^2 + y^2)$ dla $(x, y) \in [-3, 3] \times [-3, 3]$.
52. Narysować wykres krzywej (linii śrubowej) danej równaniami:
 $x(t) = \cos t$, $y(t) = \sin t$, $z(t) = t$, $t \in [0, 5\pi]$.
53. Narysować elipsoidę daną równaniami:
 $x = 2 \cos a \cos b$, $y = \sin a \cos b$, $z = \sin b$, $a \in [0, 2\pi]$, $b \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.
54. Znaleźć pochodne cząstkowe rzędu drugiego funkcji $f(x, y) = x \sin(xy)$.
55. Rozwinąć zgodnie ze wzorem Taylora do wyrazów rzędu trzeciego włącznie funkcję $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x, y) = e^y \ln x$ w otoczeniu punktu $(1, 0)$.
56. Narysować wykres krzywej danej równaniem uwikłanym $y^3 + 8xy - 8x^2 - 4y^2 - 8y = 0$.
57. Znaleźć ekstrema lokalne funkcji $f(x, y) = 3x^3 + 3x^2y - y^3 - 15x$.
58. Wyznaczyć płaszczyzny trójszcianu Freneta krzywej $x = t \cos t$, $y = t \sin t$, $z = t$ dla $t = 0$.
59. Rozwiązać równanie różniczkowe $y' - xy = xe^{x^2}$.
60. Rozwiązać równanie różniczkowe $x^3y'' + x^2y' = 1$, $y(1) = 2$, $y'(1) = 3$.
61. Rozwiązać równanie różniczkowe $y'' - 2y' + 2y = e^{2x} + x$.

3.2. Rozwiązania

W podrozdziale tym, przed rozwiązaniem każdego z zadań przytaczamy raz jeszcze jego treść, co z pewnością ułatwi Czytelnikowi zrozumienie wprowadzanych poleceń do programu.

- Obliczyć $(2 - \sqrt{3})^4$.

```
(%i1) (2-sqrt(3))^4;
(%o1) (2-√3)4
(%i2) expand(%);
(%o2) 97-56√3
```
- Sprowadzić do najprostszej postaci wyrażenie $(a^{-1} - b^{-1})\frac{\sqrt{a}+\sqrt{b}}{\sqrt{b}-\sqrt{a}} - \frac{2}{\sqrt{ab}}$.

```
(%i1) (a^(-1)-b^(-1))*(sqrt(a)+sqrt(b))/(sqrt(b)-sqrt(a))-2/sqrt(a*b);
(%o1) (1/a - 1/b) (√b + √a) / (√b - √a) - 2/√ab
(%i2) radcan(%);
(%o2) (b+a)/ab
```
- Rozwiązać w zbiorze liczb rzeczywistych równanie $x^5 - 2x^4 - 4x^3 + 4x^2 - 5x + 6 = 0$.

```
(%i1) x^5-2*x^4-4*x^3+4*x^2-5*x+6=0;
(%o1) x5 - 2x4 - 4x3 + 4x2 - 5x + 6 = 0
(%i2) realroots(%);
(%o2) [x=-2, x=1, x=3]
```
- Rozwiązać w zbiorze liczb zespolonych równanie $x^5 - 2x^4 - 4x^3 + 4x^2 - 5x + 6 = 0$.

```
(%i1) x^5-2*x^4-4*x^3+4*x^2-5*x+6=0;
(%o1) x5 - 2x4 - 4x3 + 4x2 - 5x + 6 = 0
(%i2) solve(%,x);
(%o2) [x = 3, x = -2, x = -%i, x = %i, x = 1]
```
- Znaleźć wszystkie pierwiastki wielomianu $W(x) = x^5 + x^2 + 1$.

```
(%i1) W:x^5+x^2+1;
(%o1) x5 + x2 + 1
(%i2) solve(W=0,x);
(%o2) [0 = x5 + x2 + 1]
(%i3) fpprintprec:3$
(%i4) allroots(W);
(%o4) [x = 0.8%i + 0.8, x = 0.8 - 0.8%i, x = 0.8%i - 0.2, x = -0.8%i - 0.2, x = -1.19]
```
- Sprowadzić do postaci iloczynowej wielomian $W(x) = 2x^5 + 3x^4 - x^3 + x^2 - 3x - 2$.

```
(%i1) W:2*x^5+3*x^4-x^3+x^2-3*x-2;
(%o1) 2x5 + 3x4 - x3 + x2 - 3x - 2
(%i2) factor(W);
(%o2) (x-1)(x+2)(2x+1)(x2+1)
```
- Wykonać dzielenie wielomianu $P(x) = x^4 - 3x^2 + 2$ przez wielomian $Q(x) = x^3 + 2x^2 - x - 4$.

```
(%i1) P:x^4-3*x^2+2;
(%o1) x4 - 3x2 + 2
(%i2) Q:x^3+2*x^2-x-4;
(%o2) x3 + 2x2 - x - 4
```

```
(%i3) divide(P,Q);
(%o3) [x - 2, 2x^2 + 2x - 6]
```

Zatem $\frac{P(x)}{Q(x)} = x - 2 + \frac{2x^2+2x-6}{x^3+2x^2-x-4}$.

8. Rozłożyć funkcję wymierną

$$f(x) = \frac{x^3 - x^2 - x - 2}{x^4 + 4x^3 - x^2 - 16x - 12}$$

na sumę ułamków prostych.

```
(%i1) f:(x^3-x^2-x-2)/(x^4+4*x^3-x^2-16*x-12);
```

```
(%o1) 
$$\frac{x^3 - x^2 - x - 2}{x^4 + 4x^3 - x^2 - 16x - 12}$$

```

```
(%i2) partfrac(f,x);
```

```
(%o2) 
$$\frac{7}{2(x+3)} - \frac{3}{x+2} + \frac{1}{2(x+1)}$$

```

9. Rozwiązać układ równań $\begin{cases} 2x^2 + xy - 4 = 0 \\ 2y^2 + xy - 10 = 0. \end{cases}$

```
(%i1) r1:2*x^2+x*y-4=0;
```

```
(%o1)  $xy + 2x^2 - 4 = 0$ 
```

```
(%i2) r2:2*y^2+x*y-10=0;
```

```
(%o2)  $2y^2 + xy - 10 = 0$ 
```

```
(%i3) algsys([r1,r2],[x,y]);
```

```
(%o3) [ [x =  $\frac{4}{\sqrt{3}}$ , y =  $-\frac{5}{\sqrt{3}}$ ], [x =  $-\frac{4}{\sqrt{3}}$ , y =  $\frac{5}{\sqrt{3}}$ ], [x = -1, y = -2], [x = 1, y = 2] ]
```

10. Rozwiązać równanie $2^{2x} - 5 \cdot 2^x + 4 = 0$.

```
(%i1) 2^(2*x)-5*2^x+4=0;
```

```
(%o1)  $2^{2x} - 5 \cdot 2^x + 4 = 0$ 
```

```
(%i2) solve(%);
```

```
(%o2) [x = 0, x =  $\frac{\log(4)}{\log(2)}$ ]
```

```
(%i3) radcan(%);
```

```
(%o3) [x = 0, x = 2]
```

11. Rozwiązać równanie $e^x = x + 2$.

```
(%i1) f(x):=exp(x)-x-2$
```

```
(%i2) [f(-3),f(0),f(2)];
```

```
(%o2) [ $e^{-3} + 1$ , -1,  $e^2 - 4$ ]
```

```
(%i3) find_root(f(x),x,-3,0);
```

```
(%o3) -1.841405660436961
```

```
(%i4) find_root(f(x),x,0,2);
```

```
(%o4) 1.146193220620583
```

12. Obliczyć granicę ciągu $a_n = \left(\frac{n+2}{n-3}\right)^{2n}$.

```
(%i1) a:((n+2)/(n-3))^(2*n);
```

```
(%o1)  $\left(\frac{n+2}{n-3}\right)^{2n}$ 
```

```
(%i2) limit(a,n,inf);
```

```
(%o2)  $e^{10}$ 
```

13. Wyznaczyć sumę szeregu $\sum_{n=1}^{\infty} \frac{1}{n^2}$.

```
(%i1) sum(1/n^2,n,1,inf), simpsum;
```

```
(%o1) %pi^2
      6
```

14. Wykonać działania $\frac{(4-i)^2}{2+3i}$.

```
(%i1) (4-%i)^2/(2+3*i);
```

```
(%o1) (4 - %i)^2
      3%i + 2
```

```
(%i2) rectform(%);
```

```
(%o2) 6      61%i
      13     13
```

15. Znaleźć postać trygonometryczną liczby zespolonej $z = 1 - i\sqrt{3}$.

```
(%i1) z:1-%i*sqrt(3);
```

```
(%o1) 1 - sqrt(3)*i
```

```
(%i2) cabs(z)*(Cos(carg(z))+%i*Sin(carg(z)));
```

```
(%o2) 2 * ( %iSin( - %pi/3 ) + Cos( - %pi/3 ) )
```

16. Obliczyć $(1 - \sqrt{3}i)^{13}$.

```
(%i1) (1-sqrt(3)*%i)^13;
```

```
(%o1) (1 - sqrt(3)*i)^13
```

```
(%i2) rectform(%);
```

```
(%o2) 4096 - 4096*sqrt(3)*i
```

17. Obliczyć granicę $\lim_{x \rightarrow 0^-} (1 - e^{3x}) \operatorname{ctg} x$.

```
(%i1) (1-exp(3*x))*cot(x);
```

```
(%o1) (1 - %e^3x)cot(x)
```

```
(%i2) limit(%,x,0,min);
```

```
(%o2) -3
```

18. Obliczyć granicę $\lim_{x \rightarrow +\infty} \frac{x^2+2}{e^{2x}-1}$.

```
(%i1) (x^2+2)/(exp(2*x)-1);
```

```
(%o1) x^2 + 2
      %e^2x - 1
```

```
(%i2) limit(%,x,inf);
```

```
(%o2) 0
```

19. Wyznaczyć granicę $\lim_{x \rightarrow 2} \frac{1}{x-2}$.

```
(%i1) limit(1/(x-2), x, 2);
```

```
(%o1) und
```

Zatem $\lim_{x \rightarrow 2} \frac{1}{x-2}$ nie istnieje.

20. Wyznaczyć granicę $\lim_{x \rightarrow 1} x \sin \frac{1}{x-1}$.

```
(%i1) limit(x*sin(1/(x-1)), x, 1);
```

```
(%o1) ind
```

Zatem $\lim_{x \rightarrow 1} x \sin \frac{1}{x-1}$ nie istnieje.

21. Znaleźć funkcję odwrotną do funkcji $f(x) = \frac{x+3}{x-5} + 2$.

```
(%i1) f: (x+3)/(x-5)+2;
```

```
(%o1)  $\frac{x+3}{x-5} + 2$ 
```

```
(%i2) solve(f=y,x);
```

```
(%o2)  $[x = \frac{5y-7}{y-3}]$ 
```

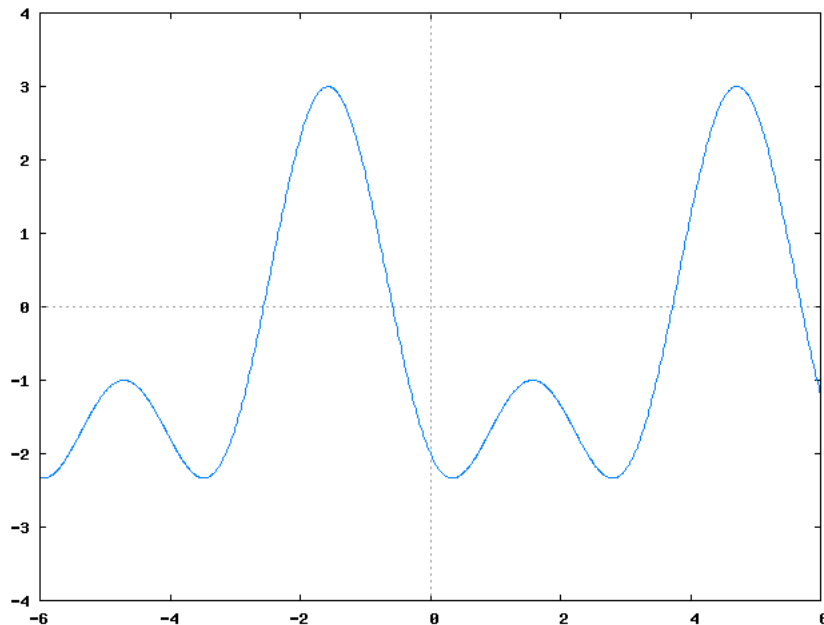
22. Narysować wykres funkcji $f(x) = 3 \sin^2 x - 2 \sin x - 2$ w przedziale $[-6, 6]$.

```
(%i1) f(x):=3*sin(x)^2-2*sin(x)-2;
```

```
(%o1)  $f(x) := 3\sin(x)^2 - 2\sin(x) - 2$ 
```

```
(%i2) wxplot2d([f(x)], [x,-6,6], [y,-4,4], [plot_format, gnuplot],  
[gnuplot_preamble, "set zeroaxis"], [nticks,100],[xlabel,""],  
[ylabel,""])$
```

```
(%t2)
```



23. Obliczyć pochodną funkcji $f(x) = \ln(\arctan(x^2 - 1))$.

```
(%i1) f:log(atan(x^2-1));
```

```
(%o1)  $\log(\arctan(x^2 - 1))$ 
```

```
(%i2) diff(f,x);
```

```
(%o2)  $\frac{2x}{((x^2 - 1)^2 + 1) \arctan(x^2 - 1)}$ 
```

24. Niech $f(x) = \sin x \cos^3(2x)$. Obliczyć $f'(\frac{\pi}{3})$.

```
(%i1) f:sin(x)*cos(2*x)^3;
```

```
(%o1)  $\sin(x)\cos(2x)^3$ 
```

```
(%i2) define(f1(x),diff(f,x));
```

```
(%o2)  $f1(x) := \cos(x)\cos(2x)^3 - 6\sin(x)\cos(2x)^2\sin(2x)$ 
```

```
(%i3) f1(%pi/3);
```

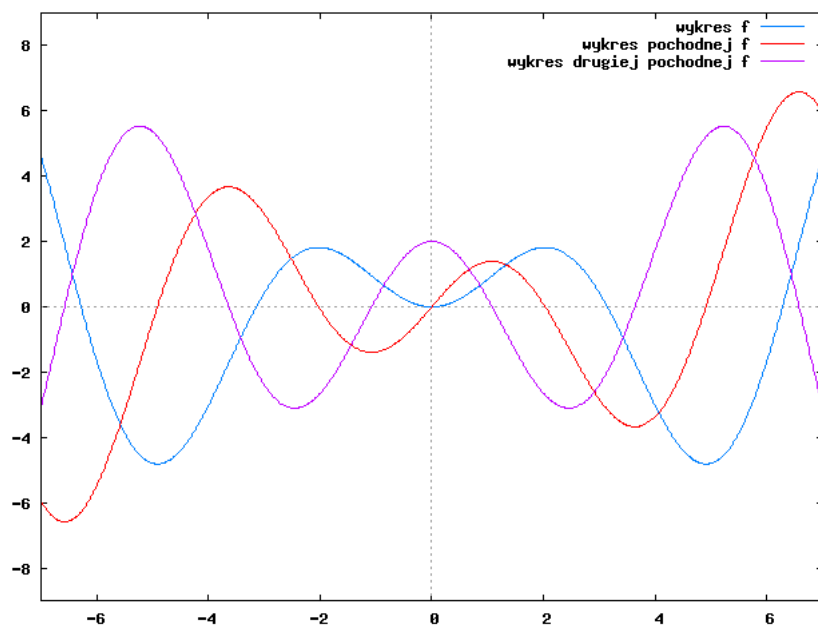
```
(%o3)  $-\frac{19}{16}$ 
```

25. Niech $f(x) = \sin(x^3)$. Obliczyć f' , f'' , f''' .

```
(%i1) f:sin(x^3);
(%o1) sin(x^3)
(%i2) diff(f,x);
(%o2) 3x^2cos(x^3)
(%i3) diff(f,x,2);
(%o3) 6xcos(x^3) - 9x^4sin(x^3)
(%i4) diff(f,x,3);
(%o4) -54x^3sin(x^3) - 27x^6cos(x^3) + 6cos(x^3)
```

26. Niech $f(x) = x \sin x$. Narysować na jednym rysunku wykresy funkcji f , f' , f'' w przedziale $[-7, 7]$.

```
(%i1) f(x):=x*sin(x)$
(%i2) f1:diff(f(x),x)$
(%i3) f2:diff(f(x),x,2)$
(%i4) wxplot2d([f,f1,f2], [x,-7,7], [y,-9,9], [gnuplot_preamble,
"set zeroaxis;"], [nticks,100], [xlabel,""], [legend, "wykres f",
"wykres pochodnej f", "wykres drugiej pochodnej f"])$
(%t4)
```



27. Rozwinąć funkcję $f(x) = \ln(1+x)$ w szereg Taylora w otoczeniu punktu $x_0 = 0$.

```
(%i1) niceindices(powerseries(log(1+x),x,0));
(%o1) - \sum_{i=1}^{\infty} \frac{(-1)^i x^i}{i}
```

28. Znaleźć pięć pierwszych wyrazów wzoru Taylora funkcji $f(x) = x\sqrt{x}$ w punkcie $x_0 = 1$.

```
(%i1) taylor(x*sqrt(x),x,1,5);
(%o1) 1 + \frac{3(x-1)}{2} + \frac{3(x-1)^2}{8} - \frac{(x-1)^3}{16} + \frac{3(x-1)^4}{128} - \frac{3(x-1)^5}{256} + \dots
```

29. Znaleźć równanie stycznej do wykresu funkcji $f(x) = \sqrt{2x} \cos(x-2)$ w punkcie $x_0 = 2$.

```
(%i1) f(x):=sqrt(2*x)*cos(x-2)$
```

```
(%i2) define(f1(x),diff(f(x),x))$
```

```
(%i3) y=f1(2)*(x-2)+f(2);
```

```
(%o3)  $y = \frac{x-2}{2} + 2$ 
```

30. Wyznaczyć ekstrema lokalne i zbadać monotoniczność funkcji $f(x) = xe^{-x^2}$

```
(%i1) f(x):=x*exp(-x^2);
```

```
(%o1)  $f(x) := x \exp(-x^2)$ 
```

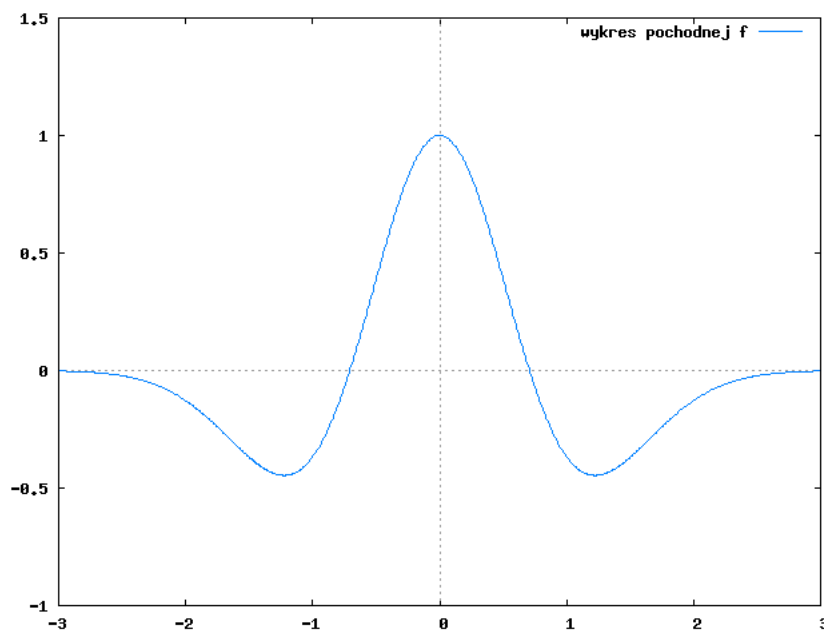
```
(%i2) define(f1(x),diff(f(x),x));
```

```
(%o2)  $f1(x) := e^{-x^2} - 2x^2 e^{-x^2}$ 
```

```
(%i3) a:solve(f1(x)=0,x);
```

```
(%o3)  $[x = -\frac{1}{\sqrt{2}}, x = \frac{1}{\sqrt{2}}]$ 
```

```
(%i4) plot2d([f1], [x,-3,3], [y,-1,1.5], [gnuplot_preamble, "set zeroaxis;"],  
[nticks,100], [xlabel,""], [ylabel,""], [legend, "wykres pochodnej f"])$  
(%t4)
```



```
(%i5) ev(f(x),a[1]);
```

```
(%o5)  $-\frac{1}{\sqrt{2}\sqrt{e}}$ 
```

```
(%i6) ev(f(x),a[2]);
```

```
(%o6)  $\frac{1}{\sqrt{2}\sqrt{e}}$ 
```

Dla $x \in (-\infty, -\frac{1}{\sqrt{2}})$ funkcja f jest malejąca. W punkcie $x = -\frac{1}{\sqrt{2}}$ ma minimum lokalne równe $f(-\frac{1}{\sqrt{2}}) = -\frac{1}{\sqrt{2}e}$. Dla $x \in (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ funkcja f jest rosnąca. W punkcie $x = \frac{1}{\sqrt{2}}$ ma maksimum lokalne równe $f(\frac{1}{\sqrt{2}}) = \frac{1}{\sqrt{2}e}$. Dla $x \in (\frac{1}{\sqrt{2}}, +\infty)$ funkcja f jest malejąca.

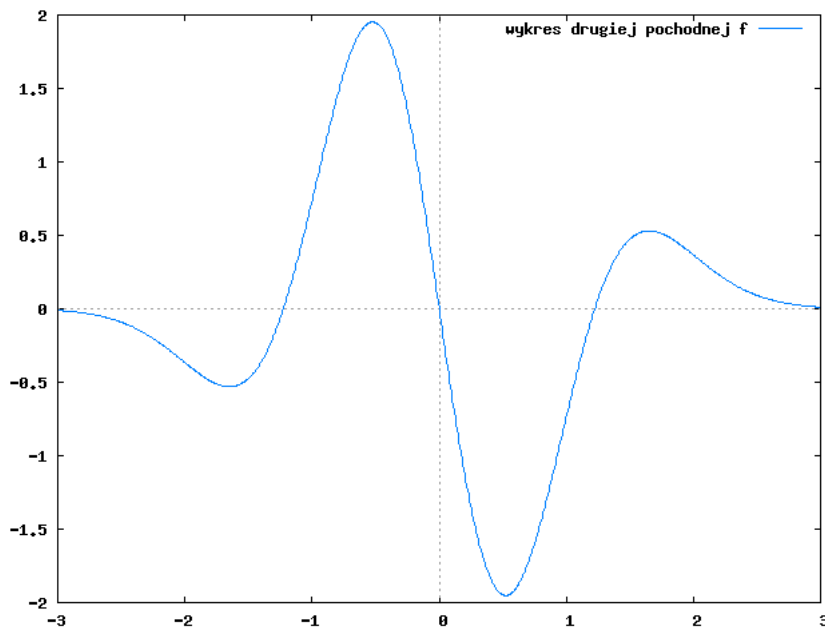
31. Wyznaczyć punkty przegięcia i zbadać wklęsłość/wypukłość funkcji $f(x) = xe^{-x^2}$

```
(%i1) f(x):=x*exp(-x^2);
```

```
(%o1)  $f(x) := x \exp(-x^2)$ 
```

```
(%i2) define(f2(x),diff(f(x),x,2));
```

```
(%o2) f2(x) := 4x^3*e^-x^2 - 6x*e^-x^2
(%i3) solve(f2(x)=0,x);
(%o3) [x = -sqrt(3)/2, x = sqrt(3)/2, x = 0]
(%i4) plot2d([f2], [x,-3,3], [y,-2,2], [gnuplot_preamble, "set zeroaxis;"],
  [nticks,100], [xlabel,""], [ylabel,""], [legend, "wykres drugiej pochodnej
  f"])$
(%t4)
```



Dla $x \in (-\infty, -\sqrt{\frac{3}{2}})$ funkcja f jest wklęsła. W punkcie $x = -\sqrt{\frac{3}{2}}$ ma punkt przegięcia. Dla $x \in (-\sqrt{\frac{3}{2}}, 0)$ funkcja f jest wypukła. W punkcie $x = 0$ ma punkt przegięcia. Dla $x \in (0, \sqrt{\frac{3}{2}})$ funkcja f jest wklęsła. W punkcie $x = \sqrt{\frac{3}{2}}$ ma punkt przegięcia. Dla $x \in (\sqrt{\frac{3}{2}}, +\infty)$ funkcja f jest wypukła.

- 32.** Znaleźć wartość najmniejszą i największą funkcji $f(x) = (x^3 - 2x^2 + 2x - 2)e^x$ w przedziale $[-3, 3]$.

```
(%i1) f(x):=(x^3-2*x^2+2*x-2)*%e^x$
(%i2) 'solve(diff(f(x),x)=0,x);
(%o2) [x = 1, x = -2, x = 0, %e^x = 0]
(%i3) a:[f(-3), f(-2), f(0), f(1), f(3)];
(%o3) [-53%e^-3, -22%e^-2, -2, -%e, 13%e^3]
(%i4) lmin(a);
(%o4) -22%e^-2
(%i5) lmax(a);
(%o5) 13%e^3
```

Funkcja f w przedziale $[-3, 3]$ przyjmuje wartość najmniejszą w punkcie $x = -2$ i jest ona równa $f(-2) = -22e^{-2}$, natomiast wartość największą w punkcie $x = 3$ i jest ona równa $f(3) = 13e^3$.

33. Obliczyć całkę nieoznaczoną $\int x^2 \arctan x dx$.

```
(%i1) integrate(x^2*atan(x),x);
```

```
(%o1)  $\frac{x^3 \arctan(x)}{3} - \frac{\frac{x^2}{2} - \frac{\log(x^2+1)}{2}}{3}$ 
```

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

```
(%o2)  $\frac{\log(x^2+1) + 2x^3 \arctan(x) - x^2}{6}$ 
```

34. Obliczyć całkę oznaczoną $\int_0^{\frac{\pi}{2}} \cos^3 x dx$.

```
(%i1) integrate(cos(x)^3, x, 0, %pi/2);
```

```
(%o1)  $\frac{2}{3}$ 
```

35. Obliczyć całkę oznaczoną $\int_0^4 \arctan^2 x dx$.

```
(%i1) integrate(atan(x)^2, x, 0, 4);
```

```
(%o1)  $\int_0^4 \arctan(x)^2 dx$ 
```

```
(%i2) romberg(atan(x)^2, x, 0, 4);
```

```
(%o2) 4.271673744976122
```

36. Obliczyć całkę niewłaściwą $\int_{-\infty}^{+\infty} \frac{dx}{x^2+16}$.

```
(%i1) integrate(1/(x^2+16), x, minf, inf);
```

```
(%o1)  $\frac{\pi}{4}$ 
```

37. Obliczyć całkę niewłaściwą $\int_1^e \frac{dx}{x \ln x}$.

```
(%i1) integrate(1/(x*log(x)), x, 1, %e);
```

```
Is x+%e positive, negative, or zero?positive;
```

```
Is x+%e-1 positive, negative, or zero?positive;
```

```
Integral is divergent
```

```
-- an error. Quitting. To debug this try debugmode(true);
```

Zatem całka $\int_1^e \frac{dx}{x \ln x}$ jest rozbieżna.

38. Obliczyć pole obszaru ograniczonego wykresami funkcji: $f(x) = 2x^2 - 12x + 10$, $g(x) = 2x - 2$.

```
(%i1) f:2*x^2-12*x+10$
```

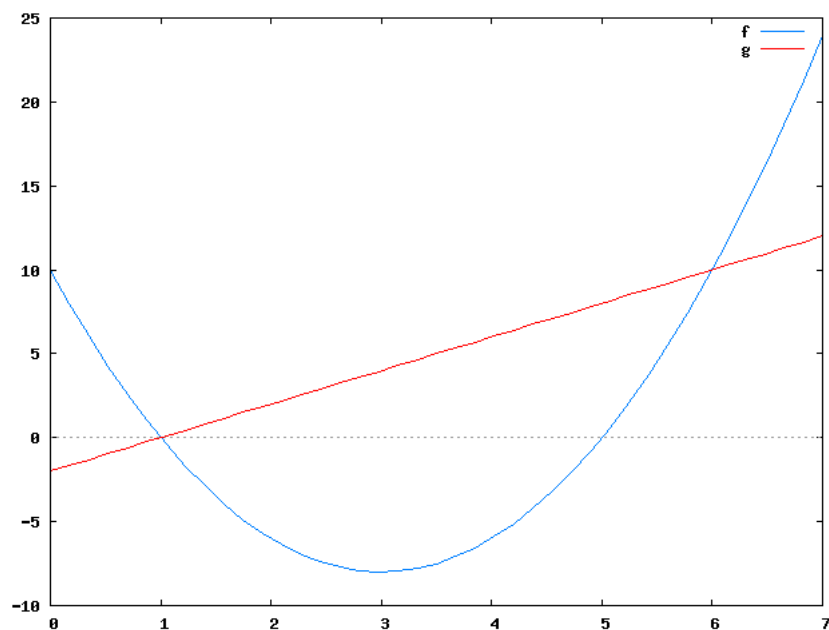
```
(%i2) g:2*x-2$
```

```
(%i3) solve(f-g=0,x);
```

```
(%o3) [x = 6, x = 1]
```

```
(%i4) wxplot2d([f,g], [x,0,7], [gnuplot_preamble, "set zeroaxis;"],  
[xlabel,""])$
```

```
(%t4)
```



```
(%i5) integrate(g-f, x, 1, 6);
```

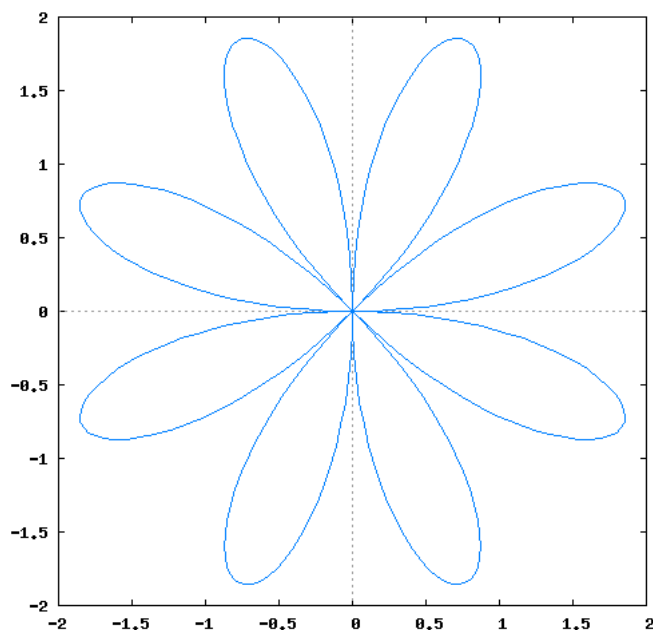
```
(%o5) 125  
      3
```

39. Narysować wykres krzywej (rozety ośmiolistnej) danej równaniami:

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

```
(%i1) wxplot2d([[parametric, 2*sin(4*t)*cos(t), 2*sin(4*t)*sin(t),  
[t, 0, 2*%pi]]], [gnuplot_preamble, "set size ratio 1; set zeroaxis;"],  
[nticks,300], [xlabel,""], [ylabel,""])]$
```

```
(%t1)
```



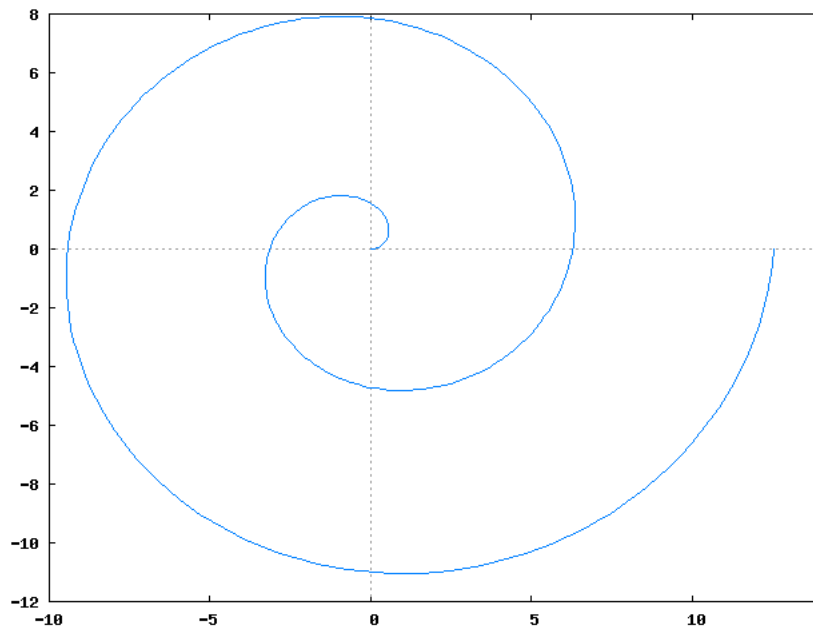
40. Obliczyć długość krzywej $y = \sqrt{x+2}$, $-2 \leq x \leq 0$.

```
(%i1) y:sqrt(x+2)$
```

```
(%i2) y1:diff(y,x)$
(%i3) integrate(sqrt(1+y1^2), x, -2, 0);
(%o3)  $\frac{\log(12\sqrt{2}+17)}{8} + \frac{3\sqrt{2}}{2}$ 
```

41. Narysować wykres krzywej (spirali Archimedesesa) zadanej biegunowo: $r = t$, $t \in [0, 4\pi]$.

```
(%i1) wxplot2d(['parametric, t*cos(t), t*sin(t), [t, 0, 4*%pi],
[nticks, 300]]], [x,-10,14], [gnuplot_preamble, "set zeroaxis;"],
[xlabel,""], [ylabel,""])$
(%t1)
```

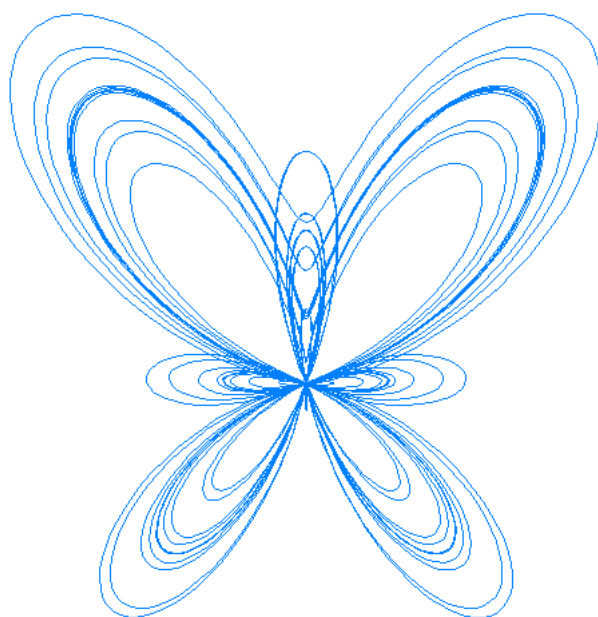


42. Narysować wykres krzywej danej równaniami:

$$x = \sin t \left(e^{\cos t} - 2 \cos(4t) + \sin^5\left(\frac{t}{12}\right) \right), \quad y = \cos t \left(e^{\cos t} - 2 \cos(4t) + \sin^5\left(\frac{t}{12}\right) \right)$$

dla $t \in [0, 24\pi]$.

```
(%i1) x(t):=sin(t)*(exp(cos(t))-2*cos(4*t)+sin(t/12)^5)$
(%i2) y(t):=cos(t)*(exp(cos(t))-2*cos(4*t)+sin(t/12)^5)$
(%i3) wxplot2d(['parametric, x, y, [t, 0, 24*%pi], [nticks, 3000]]],
[x,-5,5], [box,false], [xlabel,""], [ylabel,""])$
(%t3)
```



43. Niech $A = \begin{bmatrix} 1 & 4 & -1 & 3 \\ 3 & -1 & 2 & 5 \\ -1 & 3 & 1 & 0 \\ -3 & 5 & 2 & 1 \end{bmatrix}$. Obliczyć wyznacznik macierzy A .
- ```
(%i1) A:matrix([1,4,-1,3], [3,-1,2,5], [-1,3,1,0], [-3,5,2,1])$
(%i2) determinant(A);
(%o2) -106
```
44. Niech  $A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 2 & 2 & 4 & 6 \\ 1 & 1 & 2 & 3 \end{bmatrix}$ . Wyznaczyć rząd macierzy  $A$ .
- ```
(%i1) A:matrix([0,1,1,1], [2,2,4,6], [1,1,2,3])$
(%i2) rank(A);
(%o2) 2
```
45. Niech $A = \begin{bmatrix} 1 & 0 & 0 \\ 3 & -1 & 0 \\ -1 & 2 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 2 & 0 \\ 0 & -1 & 3 \\ 0 & 1 & 1 \end{bmatrix}$. Znaleźć macierz $A^{-1} \cdot B^T$.
- ```
(%i1) A:matrix([1,0,0], [3,-1,0], [-1,2,1])$
(%i2) B:matrix([1,2,0], [0,-1,3], [0,1,1])$
(%i3) invert(A).transpose(B);
(%o3) $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & -1 \\ -1 & 1 & 3 \end{bmatrix}$
```
46. Niech  $A = \begin{bmatrix} 2 & 1 \\ 3 & -1 \end{bmatrix}$ ,  $B = \begin{bmatrix} 2 & -1 \\ 0 & 3 \\ 4 & 1 \end{bmatrix}$ ,  $C = \begin{bmatrix} 4 & 2 & 0 \\ 1 & -1 & 1 \end{bmatrix}$ .  
Znaleźć macierz  $2A^{-1} + 3C \cdot B - 4A^T$ .
- ```
(%i1) A:matrix([2,1], [3,-1])$
(%i2) B:matrix([2,-1], [0,3], [4,1])$
(%i3) C:matrix([4,2,0], [1,-1,1])$
```

```
(%i4) 2*invert(A)+3*C.B-4*transpose(A);
```

```
(%o4) 
$$\begin{bmatrix} \frac{82}{5} & -\frac{28}{5} \\ \frac{76}{5} & -\frac{29}{5} \end{bmatrix}$$

```

47. Rozwiązać poniższy układ równań:

$$\begin{cases} 5x_1 + 2x_2 - x_3 = 4 \\ 3x_1 + x_2 - 2x_3 = 1 \\ 2x_1 + 3x_2 + 3x_3 = 5 \end{cases}$$

```
(%i1) linsolve([5*x1+2*x2-x3=4, 3*x1+x2-2*x3=1, 2*x1+3*x2+3*x3=5],
[x1,x2,x3]);
```

```
(%o1) [x1 = 1, x2 = 0, x3 = 1]
```

48. Rozwiązać poniższy układ równań:

$$\begin{cases} x + 2y + z + t = 2 \\ -2x - 3y + z - t = 0 \\ -x - y + 2z = 2 \end{cases}$$

```
(%i1) linsolve([x+2*y+z+t=2, -2*x-3*y+z-t=0, -x-y+2*z=2], [x,y,z,t]);
Dependent equations eliminated: (3)
```

```
(%o1) [x = %r1, y = 2*r2 - %r1 - 2, z = %r2, t = -5*r2 + %r1 + 6]
```

Układ równań posiada nieskończenie wiele rozwiązań zależnych od dwóch parametrów.

49. Rozwiązać poniższy układ równań:

$$\begin{cases} x_1 + x_2 - x_3 = 2 \\ 2x_1 + x_2 + x_3 = 0 \\ -x_1 - x_2 + x_3 = 1 \end{cases}$$

```
(%i1) linsolve([x1+x2-x3=2, 2*x1+x2+x3=0, -x1-x2+x3=1], [x1,x2,x3]);
```

```
Inconsistent equations: (3)
```

```
-- an error. Quitting. To debug this try debugmode(true);
```

Układ równań nie posiada rozwiązań.

50. Posługując się definicją liniowej niezależności wektorów, sprawdzić, czy wektory $u_1 = (1, 3, 0, -2)$, $u_2 = (2, 1, 2, 3)$, $u_3 = (-1, 0, 4, 1)$, $u_4 = (0, 2, -2, 1)$ są liniowo niezależne.

```
(%i1) u1:[1,3,0,-2]$
```

```
(%i2) u2:[2,1,2,3]$
```

```
(%i3) u3:[-1,0,4,1]$
```

```
(%i4) u4:[0,2,-2,1]$
```

```
(%i5) r:a*u1+b*u2+c*u3+d*u4;
```

```
(%o5) [-c + 2b + a, 2d + b + 3a, -2d + 4c + 2b, d + c + 3b - 2a]
```

```
(%i6) linsolve([r[1]=0, r[2]=0, r[3]=0, r[4]=0], [a,b,c,d]);
```

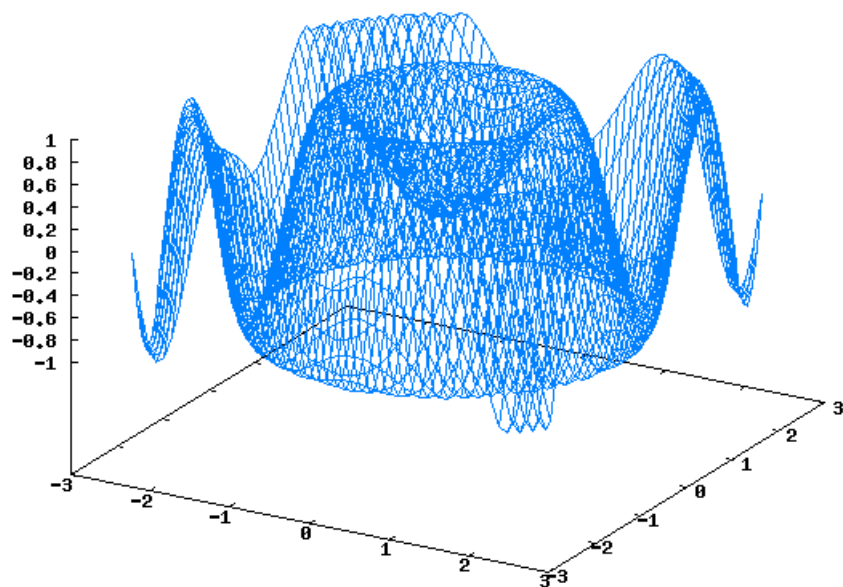
```
(%o6) [a = 0, b = 0, c = 0, d = 0]
```

Wektory u_1 , u_2 , u_3 , u_4 są liniowo niezależne.

51. Narysować wykres funkcji $f(x, y) = \sin(x^2 + y^2)$ dla $(x, y) \in [-3, 3] \times [-3, 3]$.

```
(%i1) plot3d(sin(x^2+y^2), [x,-3,3], [y,-3,3], [plot_format,gnuplot],
[grid,50,50], [gnuplot_pm3d,false], [gnuplot_curve_titles, ["title ''"]])$
```

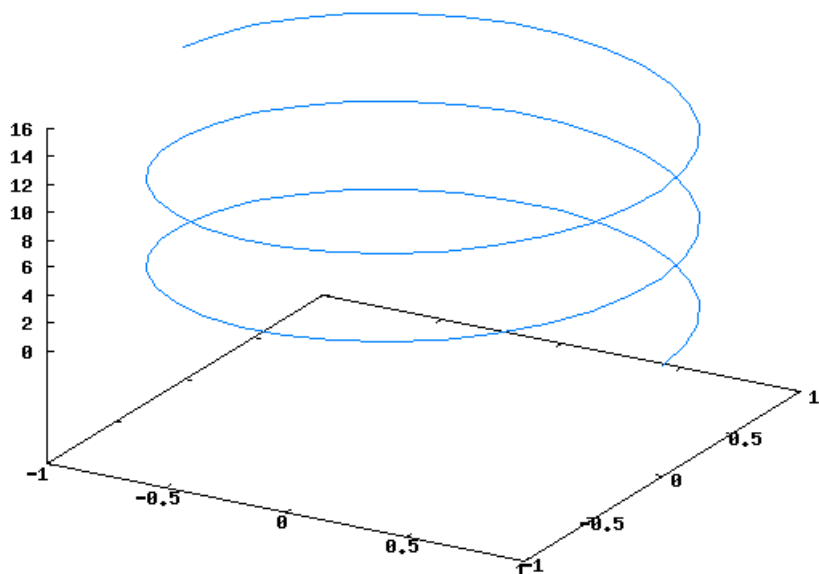
(%t1)



52. Narysować wykres krzywej (linii śrubowej) danej równaniami:

$$x(t) = \cos t, y(t) = \sin t, z(t) = t, t \in [0, 5\pi].$$

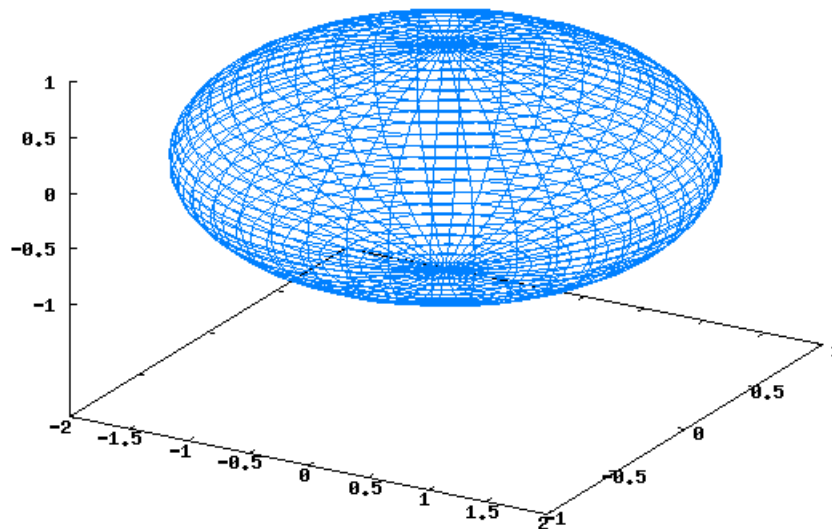
```
(%i1) plot3d([cos(t),sin(t),t], [t,0,5*%pi], [y,-1,1], [grid,80,80],
  [gnuplot_curve_titles, ["title '"]], [gnuplot_pm3d,false])$
(%t1)
```



53. Narysować elipsoide daną równaniami:

$$x = 2 \cos a \cos b, y = \sin a \cos b, z = \sin b, a \in [0, 2\pi], b \in [-\frac{\pi}{2}, \frac{\pi}{2}].$$

```
(%i1) plot3d([2*cos(a)*cos(b),sin(a)*cos(b),sin(b)], [a,0,2*%pi],
    [b,-%pi/2,%pi/2], [gnuplot_pm3d,false], [grid,40,40],
    [gnuplot_preamble, "set nokey"])$
(%t1)
```



54. Znaleźć pochodne cząstkowe rzędu drugiego funkcji $f(x, y) = x \sin(xy)$.

```
(%i1) f:x*sin(x*y)$
(%i2) diff(f,x,2);
(%o2) 2ycos(xy) - xy^2sin(xy)
(%i3) diff(f,y,2);
(%o3) -x^3sin(xy)
(%i4) diff(f,x,1,y,1);
(%o4) 2xcos(xy) - x^2ysin(xy)
```

55. Rozwinąć zgodnie ze wzorem Taylora do wyrazów rzędu trzeciego włącznie funkcję $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x, y) = e^y \ln x$ w otoczeniu punktu $(1, 0)$.

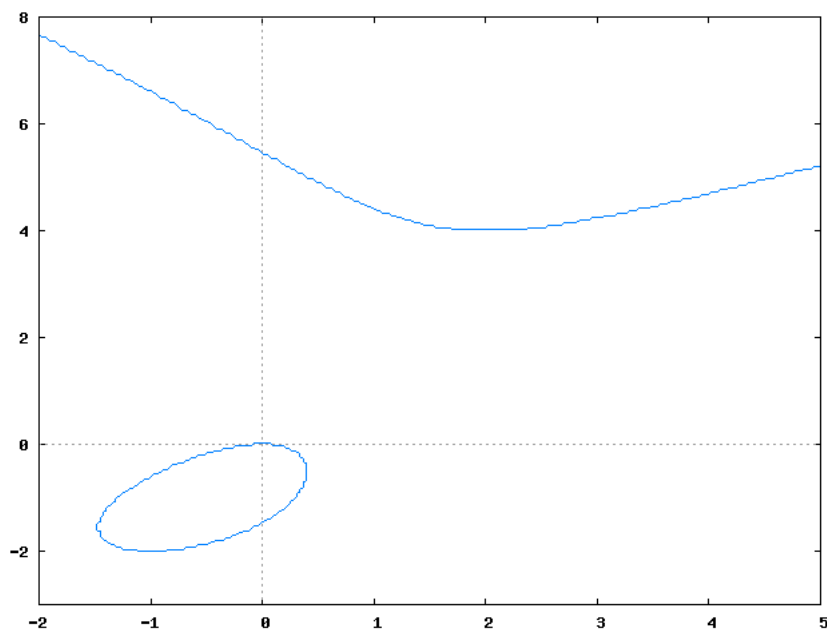
```
(%i1) taylor(exp(y)*log(x), [x,y], [1,0], 3)$ ev(%, [x=h+1,y=k]);
(%i2)
(%o2) 
$$\frac{3hk^2 + (-3h^2 + 6h)k + 2h^3 - 3h^2 + 6h}{6}$$

(%i3) expand(%);
(%o3) 
$$\frac{hk^2}{2} - \frac{h^2k}{2} + hk + \frac{h^3}{3} - \frac{h^2}{2} + h$$

```

56. Narysować wykres krzywej danej równaniem uwikłanym $y^3 + 8xy - 8x^2 - 4y^2 - 8y = 0$.

```
(%i1) load(implicit_plot)$
(%i2) F(x,y):=y^3+8*x*y-8*x^2-4*y^2-8*y;
(%o2) F(x,y):=y^3+8*x*y+(-8)*x^2+(-4)*y^2+(-8)*y
(%i3) implicit_plot(F(x,y)=0,[x,-2,5],[y,-3,8], [gnuplot_preamble,
"set zeroaxis"], [gnuplot_curve_titles, ["title ''"]])$
(%t3)
```



57. Znaleźć ekstrema lokalne funkcji $f(x, y) = 3x^3 + 3x^2y - y^3 - 15x$

```
(%i1) f(x,y):=3*x^3+3*x^2*y-y^3-15*x;
(%o1) f(x,y) := 3x^3 + 3x^2y - y^3 + (-15)x
(%i2) f_x:diff(f(x,y),x);
(%o2) 6xy + 9x^2 - 15
(%i3) f_y:diff(f(x,y),y);
(%o3) 3x^2 - 3y^2
(%i4) realonly:true$
(%i5) p:algsys([f_x=0,f_y=0],[x,y]);
(%o5) [[x = -sqrt(5), y = sqrt(5)], [x = sqrt(5), y = -sqrt(5)], [x = -1, y = -1], [x = 1, y = 1]]
(%i6) A:matrix([diff(f(x,y),x,2),diff(f(x,y),x,1,y,1)],
[diff(f(x,y),y,1,x,1),diff(f(x,y),y,2)]);
(%o6) [ 6y + 18x  6x
        6x      -6y ]
(%i7) A1:ev(A,p[1]);
(%o7) [ -12sqrt(5) -6sqrt(5)
        -6sqrt(5) -6sqrt(5) ]
(%i8) minory_glowne:[newdet(A1,1),newdet(A1,2)];
(%o8) [-12sqrt(5),36sqrt(5)^2]
(%i9) ev(f(x,y),p[1]);
(%o9) 10sqrt(5)
(%i10) A2:ev(A,p[2]);
```



```

(%o10)  $\begin{bmatrix} 12\sqrt{5} & 6\sqrt{5} \\ 6\sqrt{5} & 6\sqrt{5} \end{bmatrix}$ 
(%i11) minory_glowne:[newdet(A2,1),newdet(A2,2)];
(%o11)  $[12\sqrt{5}, 36\sqrt{5}^2]$ 
(%i12) ev(f(x,y),p[2]);
(%o12)  $-10\sqrt{5}$ 
(%i13) A3:ev(A,p[3]);
(%o13)  $\begin{bmatrix} -24 & -6 \\ -6 & 6 \end{bmatrix}$ 
(%i14) minory_glowne:[newdet(A3,1),newdet(A3,2)];
(%o14)  $[-24, -180]$ 
(%i15) A4:ev(A,p[4]);
(%o15)  $\begin{bmatrix} 24 & 6 \\ 6 & -6 \end{bmatrix}$ 
(%i16) minory_glowne:[newdet(A3,1),newdet(A3,2)];
(%o16)  $[24, -180]$ 

```

W punkcie $(-\sqrt{5}, \sqrt{5})$ funkcja f ma maksimum lokalne równe $10\sqrt{5}$, w punkcie $(\sqrt{5}, -\sqrt{5})$ ma minimum lokalne równe $-10\sqrt{5}$. W punktach $(-1, -1)$, $(1, 1)$ funkcja f nie ma ekstremum.

58. Wyznaczyć płaszczyzny trójscianu Freneta krzywej $x = t \cos t$, $y = t \sin t$, $z = t$ dla $t = 0$.

```

(%i1) load(funcs)$
(%i2) load(vect)$
(%i3) f_t:[t*cos(t),t*sin(t),t]$
(%i4) W_t:wronskian(f_t,t);
(%o4)  $\begin{bmatrix} t\cos(t) & t\sin(t) & t \\ \cos(t) - t\sin(t) & \sin(t) + t\cos(t) & 1 \\ -2\sin(t) - t\cos(t) & 2\cos(t) - t\sin(t) & 0 \end{bmatrix}$ 
(%i5) W:ev(W_t,t=0);
(%o5)  $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix}$ 
(%i6) block(f:W[1],f1:W[2],f2:W[3])$
(%i7) pl_normalna:(x-f[1])*f1[1]+(y-f[2])*f1[2]+(z-f[3])*f1[3]=0;
(%o7)  $z + x = 0$ 
(%i8) A:express(f1~f2);
(%o8)  $[-2, 0, 2]$ 
(e i9) pl_scisle_styczna:(x-f[1])*A[1]+(y-f[2])*A[2]+(z-f[3])*A[3]=0;
(%o9)  $2z - 2x = 0$ 
(e i10) X:express(A~f1);
(%o10)  $[0, 4, 0]$ 
(%i11) pl_prostujaca:(x-f[1])*X[1]+(y-f[2])*X[2]+(z-f[3])*X[3]=0;
(%o11)  $4y=0$ 

```

59. Rozwiązać równanie różniczkowe $y' - xy = xe^{x^2}$.

(%i1) 'diff(y,x)-x*y=x*exp(x^2);

(%o1) $\frac{d}{dx}y - xy = xe^{x^2}$

(%i2) ode2(%, y, x);

(%o2) $y = e^{\frac{x^2}{2}} \left(e^{\frac{x^2}{2}} + c \right)$

60. Rozwiązać równanie różniczkowe $x^3y'' + x^2y' = 1$, $y(1) = 2$, $y'(1) = 3$.

(%i1) x^3*'diff(y,x,2)+x^2*'diff(y,x)=1;

(%o1) $x^3 \left(\frac{d^2}{dx^2}y \right) + x^2 \left(\frac{d}{dx}y \right) = 1$

(%i2) ode2(%, y, x);

(%o2) $y = k_2 \log(x) + \frac{1}{x} + k_1$

(%i3) ic2(%,x=1,y=2,diff(y,x)=3);

(%o3) $y = 4\log(x) + \frac{1}{x} + 1$

61. Rozwiązać równanie różniczkowe $y'' - 2y' + 2y = e^{2x} + x$.

(%i1) r:'diff(y,x,2)-2*'diff(y,x)+2*y=exp(2*x)+x;

(%o1) $\frac{d^2}{dx^2}y - 2 \left(\frac{d}{dx}y \right) + 2y = e^{2x} + x$

(%i2) ode2(r, y, x);

(%o2) $y = e^x(k_1 \sin(x) + k_2 \cos(x)) + \frac{e^{2x} + x + 1}{2}$

Literatura

- [1] De Souza P. N. , Fateman R. J., Moses J., Yapp C., *The Maxima Book*, maxima.sourceforge.net/documentation.html
- [2] Hosack J. M., Small D. B., *Ćwiczenia z analizy matematycznej z zastosowaniem systemów obliczeń symbolicznych*, WNT, Warszawa 1995.
- [3] Lachowicz C. T., *Matlab, Scilab, Maxima. Opis i przykłady zastosowań*, Wydawnictwo Politechniki Opolskiej, Opole 2005.
- [4] Ombach J., *Wykłady z równań różniczkowych wspomagane komputerowo - Maple*, Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków 1999.
- [5] Ptak M., *Matematyka dla studentów kierunków technicznych i przyrodniczych*, Wydawnictwo AR w Krakowie, Kraków 2006.
- [6] Rand R. H., *Introduction to Maxima*, maxima.sourceforge.net/documentation.html
- [7] *Maxima Manual*, maxima.sourceforge.net/documentation.html

Skorowidz

abs, 9
acos, 9
acosh, 9
acot, 9
acoth, 9
algsys, 11
allroots, 11
and, 5
asin, 9
asinh, 9
atan, 9
atanh, 9

bc2, 20
block, 6

cabs, 12
carg, 12
changevar, 16
col, 19
conjugate, 12
contrib_ode, 22
cos, 9
cosh, 9
cot, 9
coth, 9

define, 10
del, 16
depends, 10
describe, 8
desolve, 20
determinant, 19
diff, 16
divide, 11
draw, 15

eigen, 18
eigenvalues, 20
eigenvectors, 20
entier, 9
ev, 8
example, 8
exp, 9
expand, 8
express, 18

factor, 8
find_root, 11
float, 6
fpprec, 6
fpprintprec, 6

halfangles, 8

ic1, 20
ic2, 20
ident, 19
if, 10
imagpart, 12
implicit_plot, 14
inf, 7
inprod, 18
integrate, 16
intfactor, 21
invert, 19

kill, 6

lhs, 11
limit, 13
linsolve, 11
lmax, 18
lmin, 18
load, 8
log, 9

map, 10
matrix, 19
max, 9
method, 21
min, 9
minf, 7
multiplicities, 11

ncharpoly, 20
nchrpl, 20
newdet, 19
niceindices, 17
not, 5
numer, 6

ode2, 20
odeindex, 21
or, 5

partfrac, 11
plot2d, 14
plot3d, 14
polarform, 12
powerseries, 17
product, 13

radcan, 8
rank, 19
ratsimp, 8
realonly, 11
realpart, 12
realroots, 11
rectform, 12
rhs, 11
romberg, 16
row, 19

signum, 9
sin, 9
sinh, 9
solve, 11
sqrt, 9
subst, 8
sum, 13

tan, 9
tanh, 9
taylor, 17
transpose, 19
trigexpand, 8
trigreduce, 8
trigsimp, 8

uvect, 18

vect, 18

wronskian, 20
wxplot2d, 14
wxplot3d, 14

yp, 21