MATEMATYKA Z KOMPUTEREM

ĆWICZENIA DLA STUDENTÓW REALIZOWANE ZA POMOCĄ PAKIETU MAXIMA

Anna Szadkowska

Joanna Rzepecka

Monika Potyrała

Łódź 2017

MATEMATYKA Z KOMPUTEREM

ĆWICZENIA DLA STUDENTÓW REALIZOWANE ZA POMOCĄ PAKIETU MAXIMA

Anna Szadkowska, Joanna Rzepecka, Monika Potyrała

ŁÓDŹ 2017

Recenzent
doc. dr Andrzej Just

WYDAWNICTWO POLITECHNIKI ŁÓDZKIEJ
90-924 Łódź, ul. Wólczańska 223
tel. 42 631 20 87, fax 42 631 25 38
e-mail: zamowienia@info.p.lodz.pl
www.wydawnictwa.p.lodz.pl

ISBN 978-83-7283-892-6

Wydanie III poprawione, uzupełnione i rozszerzone – wersja elektroniczna
Opublikowano w grudniu 2017 r.
Nr 2247

Spis treści

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

Przedmowa do wydania trzeciego poprawionego, uzupełnionego i rozszerzonego

Do obecnego wydania skryptu dołączono następujące nowe rozdziały:

Rozdział 13. Szeregi Fouriera,

Rozdział 16. Funkcja uwikłana jednej zmiennej,

Rozdział 20. Pakiet Simplex.

W skrypcie zostały przedstawione rozwiązania zadań przykładowych do nowych rozdziałów, bogato ilustrowane graficznie, z wykorzystaniem Maximy w wersji 5.40.0 oraz kolejne zadania do samodzielnego rozwiązania wraz z odpowiedziami.

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.

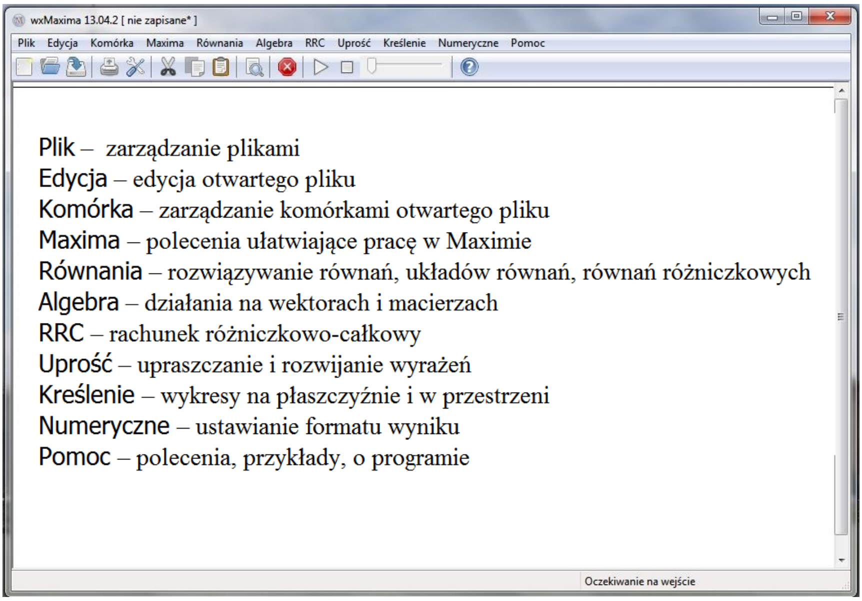
W dalszych rozdziałach zaprezentujemy możliwości programu (a czasem też ograniczenia) głównie w zakresie obliczeń symbolicznych dla wybranych tematów realizowanych na studiach I stopnia na Politechnice Łódzkiej. W skrypcie została przyjęta konwencja oznaczeń zgodna z [5].

1.1. Instalacja

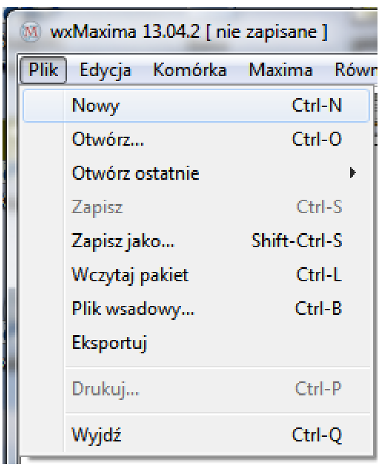
Najnowsza wersja dostępna w trakcie składania tego skryptu, to wersja 5.34.1 (8.09.2014). Należy jednak wspomnieć, że nie wszystkie wersje są dystrybuowane jako pakiety instalacyjne dla środowiska Windows, stąd obecnie najnowsza (oficjalna) dla systemu Windows wersja to: 5.31.2 (8.10.2013).
Pakiety instalacyjne są dostępne na stronie:
http://sourceforge.net/projects/maxima/files/.

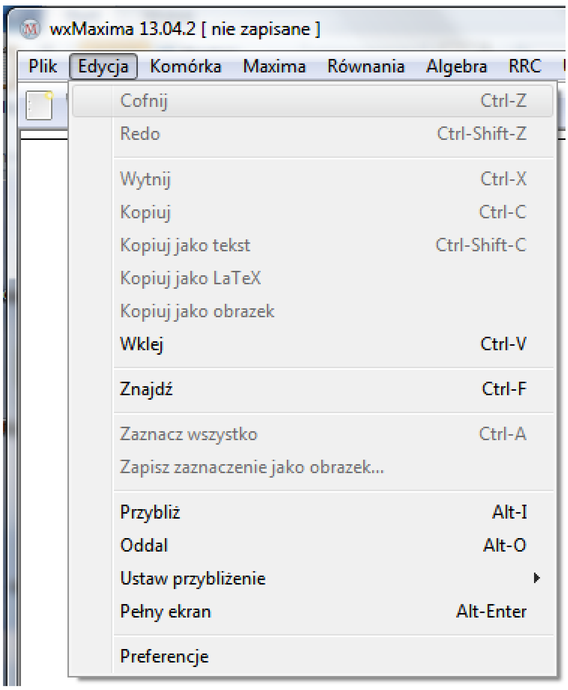
1.2. Menu, pasek narzędzi oraz skróty klawiaturowe

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

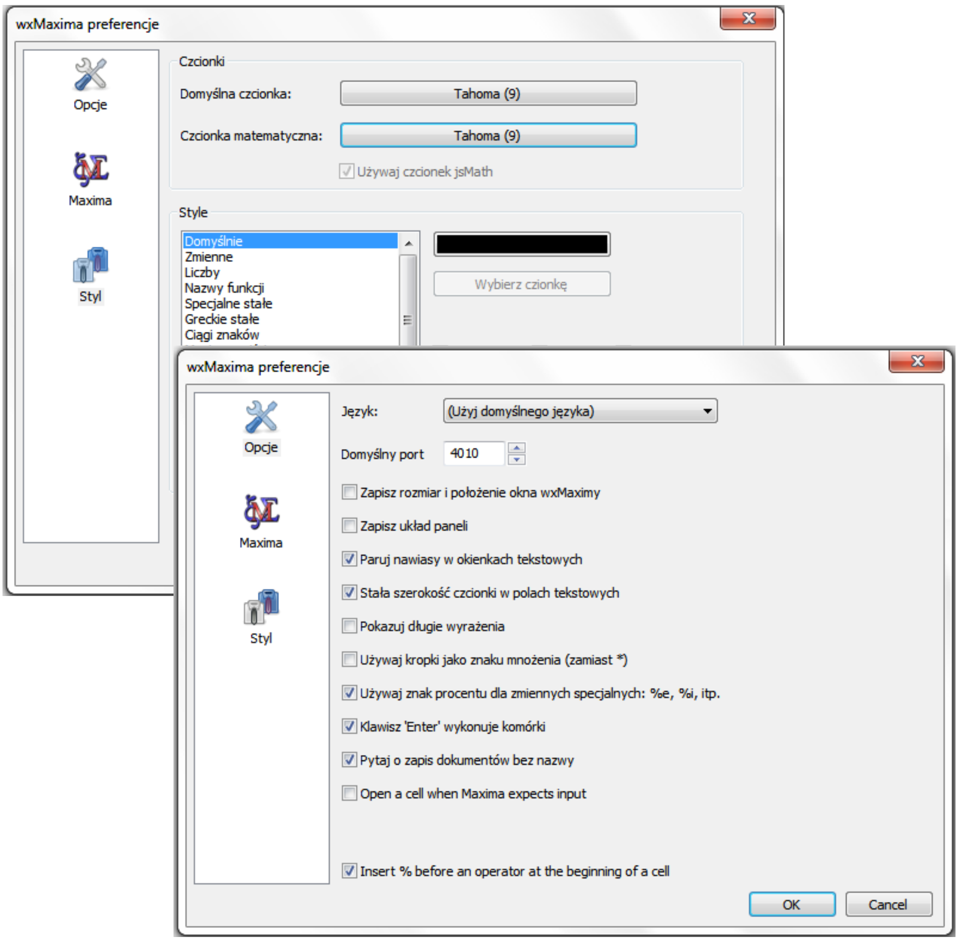
Oprócz korzystania z funkcji menu, paska narzędzi, paneli i udogodnień myszki, wiele możliwości zyskujemy, wczytując pakiety Maximy dostępne jako pliki *.mac (czasem *.lisp) w katalogu Maxima-5.31.2/share/maxima/5.31.2/share.

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
```

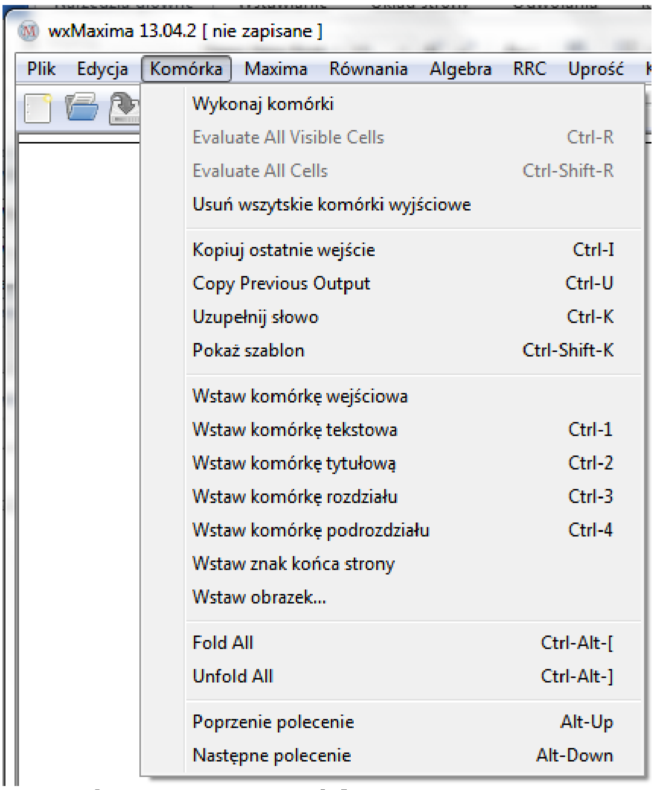
Menu-Edycja/Preferencje

Pojawiające się okno pozwala na ustawienie własnych preferencji.



Rysunek 1.4. **Preferencje**

Omawiając pracę z notatnikiem wxMaximy, 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).



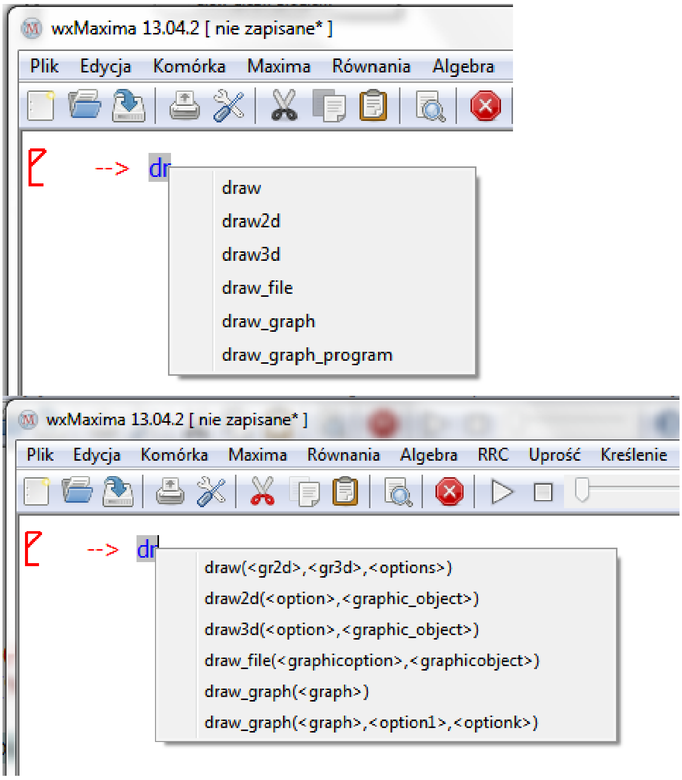
Rysunek 1.5. **Menu-Komórka**

Skróty klawiaturowe

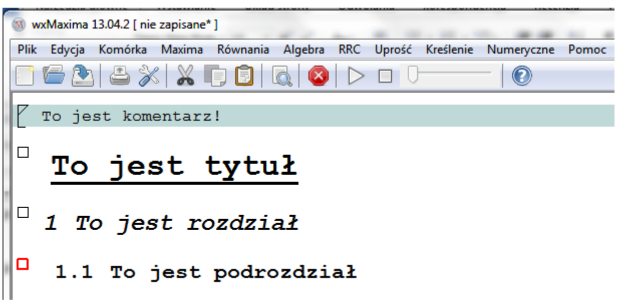
Ctrl-G – przerywa bieżące zadanie Maximy
Ctrl-R – wykonuje wszystkie nieukryte komórki
Ctrl-Shift-R – wykonuje wszystkie komórki
Ctrl-I – kopiuje ostatnie wejście
Ctrl-U – kopiuje wyjście
Ctrl-K – przedstawia warianty uzupełnienia rozpoczętego polecenia (działa już od pierwszego znaku, zarówno dla funkcji 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



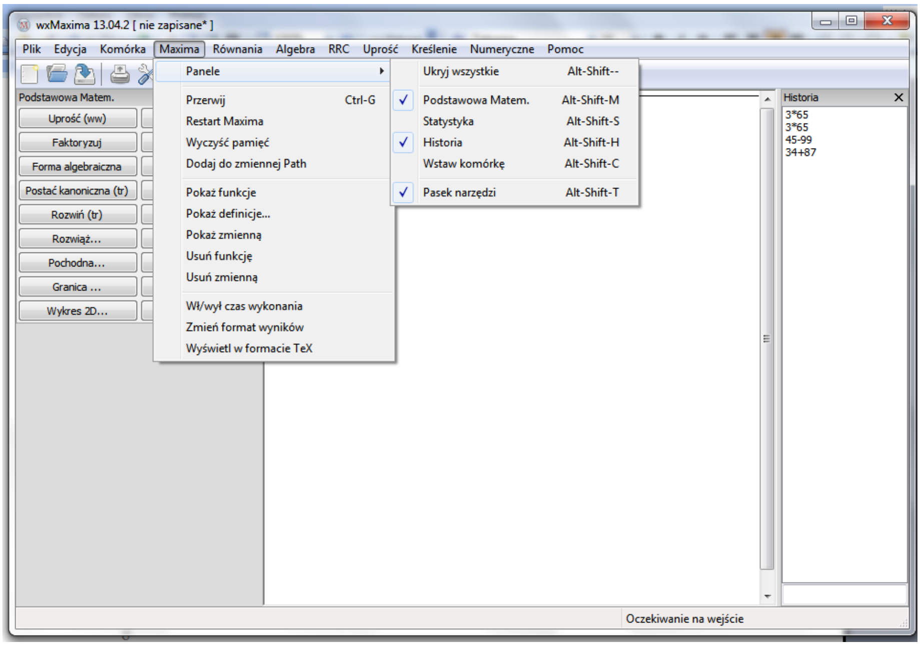
Rysunek 1.6. **Uzupełnianie polecenia i składni**



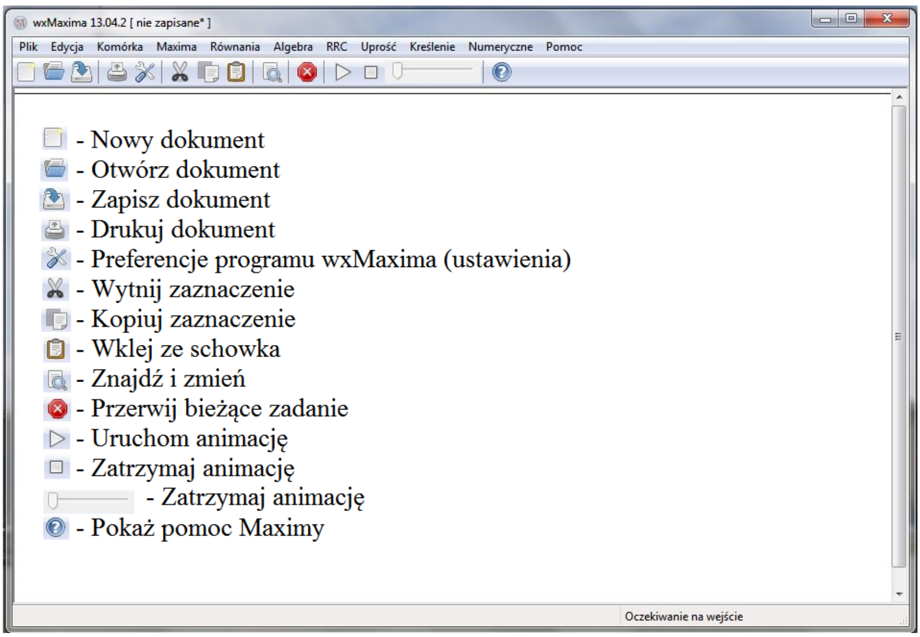
Rysunek 1.7. **Struktura dokumentu**

Uwaga 1
Aktywne komórki są zaznaczane kolorem czerwonym.

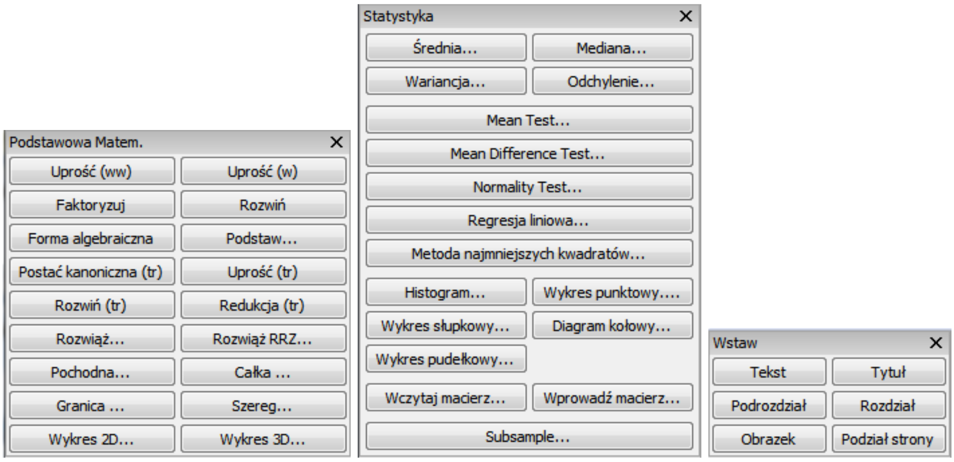
W celu usprawnienia pracy istnieje możliwość wykorzystania paska narzędzi, okna historii oraz paneli tematycznych.
Warto zapamiętać skróty klawiaturowe dla używanych paneli lub w **Preferencjach** zaznaczyć opcję: **Zapisz układ paneli.**



Rysunek 1.8. **Menu-Maxima/Panele**



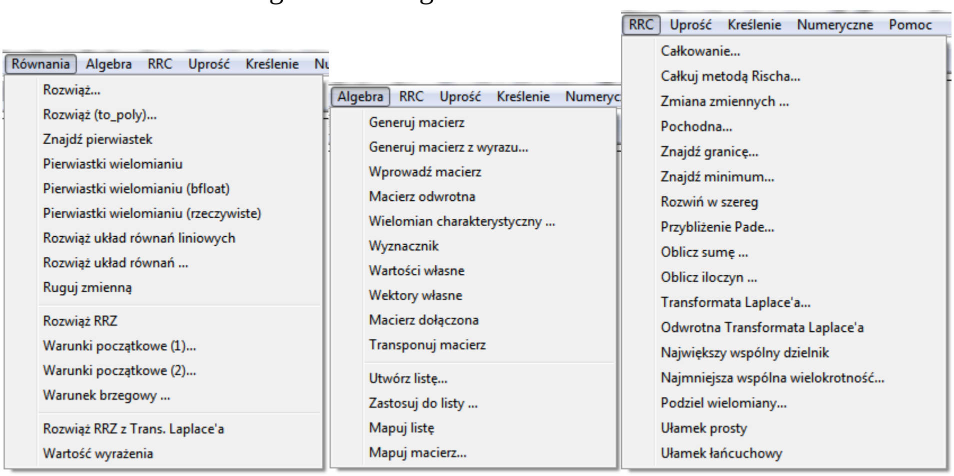
Rysunek 1.9. **Narzędzia**



Rysunek 1.10. **Panele tematyczne**

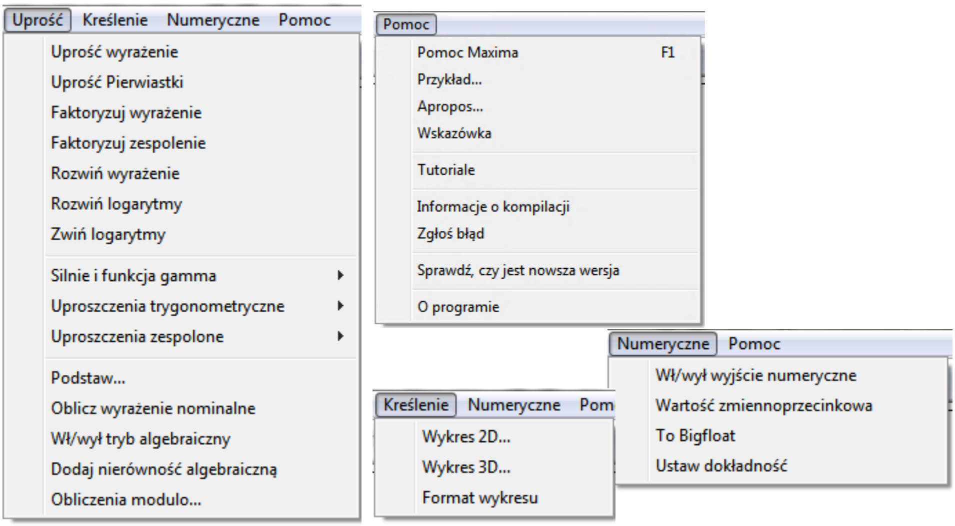
W **Menu-Maxima** znajdziemy polecenie **Restart Maxima**. Zaczyna ono sesję Maximy od nowa, tzn. resetuje ustawienia, numerację etykiet, usuwa wartości, funkcje, tablice, własności. Wybór polecenia **Maxima/Przerwij** (skrót klawiaturowy Ctrl-G) przerywa bieżące obliczenia.

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



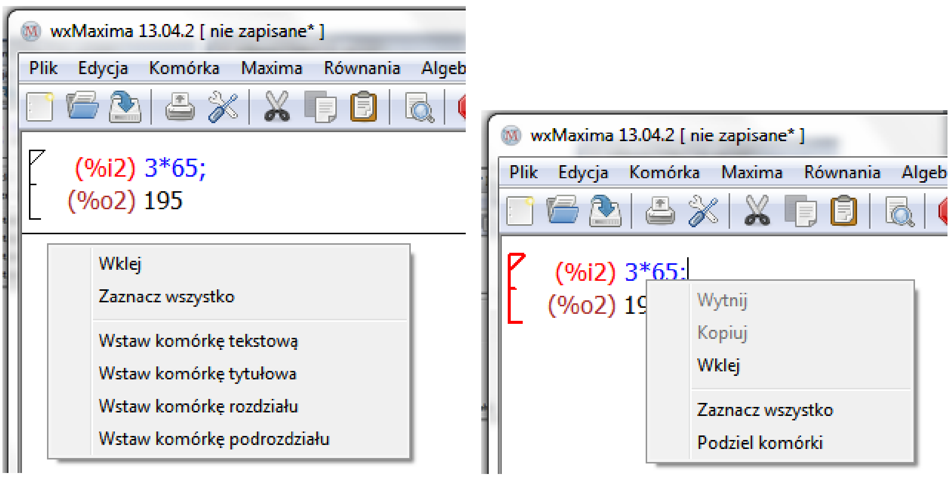
Rysunek 1.11. **Menu-Równania, Menu-Algebra, Menu-RRC**

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**

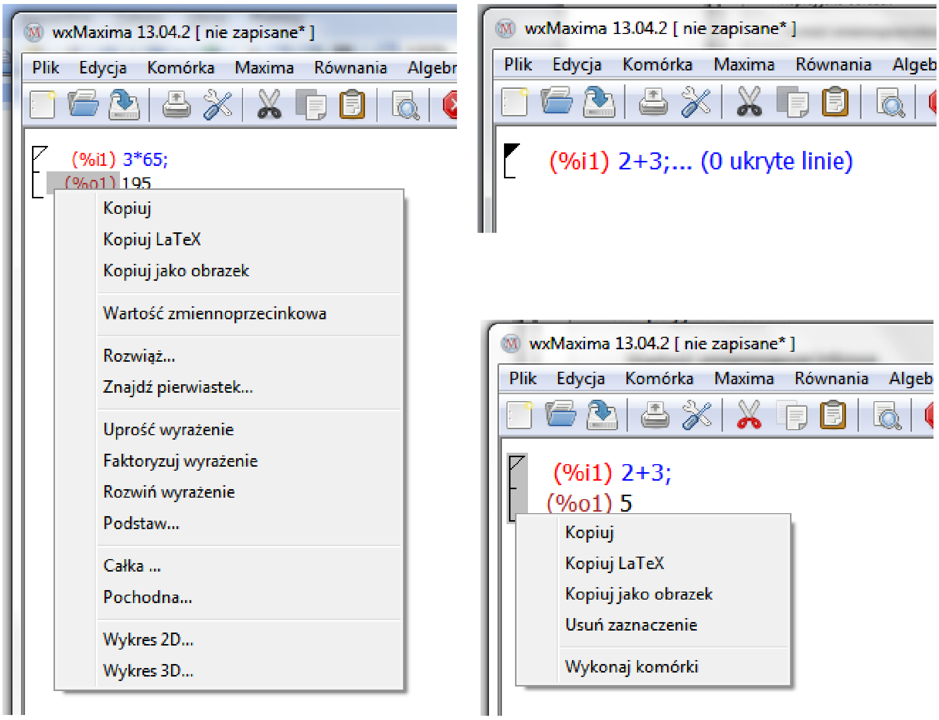
Dostęp do poleceń Maximy możemy też uzyskać, klikając **prawym przyciskiem myszy**. W zależności od miejsca kliknięcia: w wolnym polu, na linii wprowadzania input - (% i nr komórki), na linii wyniku output - (% o nr komórki) lub na zaznaczeniu komórki otwierają się odpowiednio różne menu kontekstowe:



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 ^
sqrt()	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

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

[1)] *a* może być funkcją lub zmienną, natomiast *w* np.: noun, mainvar, integer, odd, even, rational, irrational, real, complex, scalar, constant, evenfun, oddfun, additive, increasing, decreasing, posfun, commutative, integervalued, analytic, integervalued.

```
(%i1) a:3;
(%o1) 3

(%i2) f(x):=x^2;
(%o2) f(x):=x^2

(%i3) f(5)+a;
(%o3) 28

(%i4) kill(all);
(%o0) done

(%i1) a;
(%o1) a
```

```
(%i2) assume (a>0,a<1);
(%o2) [a>0,a<1]

(%i3) limit(a^n, n, inf);
(%o3) 0

(%i4) forget(a>0,a<1);
(%o4) [a>0,a<1]

(%i5) assume(a>1);
(%o5) [a>1]

(%i6) limit(a^n, n, inf);
(%o6) ∞
```

```
(%i7) assume(b<=0);
(%o7) [b<=0]

(%i8) abs(b);
(%o8) -b

(%i9) facts();
(%o9) [a>1,0>=b]

(%i10) forget (b<=0);
(%o10) [b<=0]

(%i11) facts();
(%o11) [a>1]
```

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 wyświetlania tego komunikatu.

```
(%i1) a:3$ b:4$
      a+b;
      values;
(%o3) 7
(%o4) [a,b]
```

```
(%i5) remvalue(a)$
      a+b;
      values;
(%o6) a+4
(%o7) [b]
```

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

```
(%i1) kill(f)$

(%i2) declare(f,oddfun);
(%o2) done

(%i3) f(-x)+f(x);
(%o3) 0
```

```
(%i4) kill(f)$

(%i5) declare(f,evenfun);
(%o5) done

(%i6) f(-x)+f(x);
(%o6) 2 f(x)
```

Tabela 1.3

OPERATORY PORÓWNANIA	OPIS
=	równy
#	różny (negacja równości)
> (>=)	większy (większy lub równy)
< (<=)	mniejszy (mniejszy lub równy)
compare(a,b**)**	porównuje liczby a i b i zwraca symbole: <, > lub =

Tabela 1.4

OPERATORY LOGICZNE	OPIS
not	operator logiczny - negacja
and	operator logiczny „i”
or	operator logiczny „lub”
is(p**)** lub p**, pred**	sprawdza wartość logiczną zdania *p*

```
(%i1) not 2>4;
(%o1) true

(%i2) 2>4 or 2<4;
(%o2) true

(%i3) 2>4 and 2<4;
(%o3) false
```

```
(%i1) is(5=4);
(%o1) false

(%i2) 5=4,pred;
(%o2) false

(%i3) is (5#4);
(%o3) true
```

```
(%i1) compare(log(5),2);
(%o1) <

(%i2) compare(%pi,%e);
(%o2) >
```

Tabela 1.5

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

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

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

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

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

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

Przypomnijmy definicje wybranych funkcji:

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

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

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

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

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

```
(%i5) exp(3)+log(%e);
(%o5) %e^3 + 1
```
$$\%e^3+1$$

```
(%i6) sec(%pi/6)+csc(%pi/6);
```
(%o6) $\frac{2}{\sqrt{3}}+2$

```
(%i7) (exp(1)-exp(-1))/2, numer;
(%o7) 1.175201193643801
```

```
(%i8) sinh(1), numer;
(%o8) 1.175201193643801
```

```
(%i9) is (float((exp(1)+exp(-1))/2)=float(cosh(1)));
(%o9) true
```

```
(%i10) sin(%pi/4)+cos(%pi)+tan(0)+cot(%pi/2);
```
(%o10) $\frac{1}{\sqrt{2}}-1$

```
(%i11) asin(1)+acos(0)+atan(sqrt(3))+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
display(w)	wyświetla nazwę i wartość wyrażenia *w*
print(w) albo **print(**„t1”,w1,…**)**	wykonuje i wyświetla wyrażenie jako łańcuch znakowy, pozwala też łączyć teksty i zmienne
sprint(w)	podobnie jw. tylko nie łamie linii w długich wyrażeniach
%	przechowuje ostatni output, można go przywołać
%th(k)	przywołuje *k*-ty wstecz output
%i7	przywołuje input o numerze 7
%o7	przywołuje output o numerze 7

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

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

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

```
(%i4) b:3$

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

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

```
(%i1) diff(a*x^2, x);
```
(%o1) $2\,a\,x$

```
(%i2) a:b;b:3;
```
(%o2) b

(%o3) 3

```
(%i4) diff(a*x^2, x);
```
(%o4) $2\,b\,x$

```
(%i5) ''%;
```
(%o5) $6\,x$

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

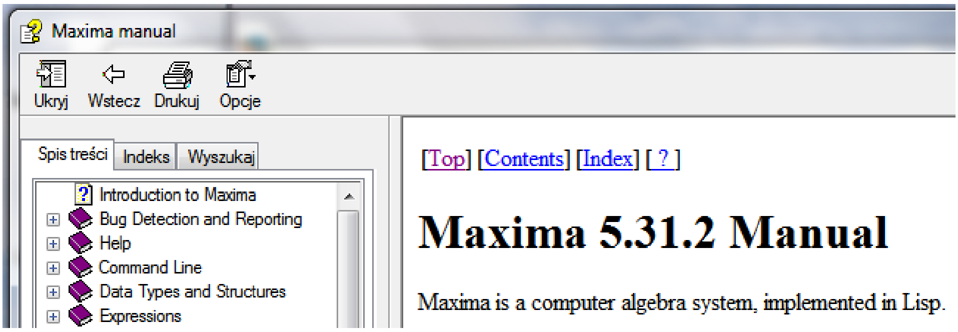
```
(%i7) display(integrate(x^2, x))$
```
$\int x^2 dx = \frac{x^3}{3}$

```
(%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
? nazwa	przedstawia opis tematu o podanej nazwie
?? nazwa	przedstawia listę tematów zawierających podaną nazwę
apropos(temat)	wyświetla nazwy Maximy zawierające dany łańcuch znaków
demo(temat)	wyszukuje pliki demonstracyjne
describe(temat)	przedstawia objaśnienie wpisanego łańcucha znaków
example(temat**)**	przedstawia przykłady do danego tematu
example()	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 14, przy okazji obliczania całek funkcji wymiernych.

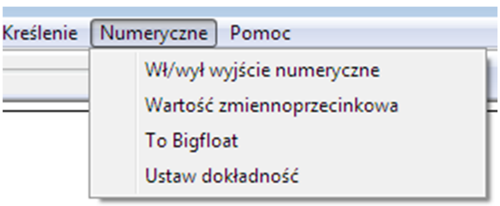
```
(%i1) ? logabs;
-- Option variable: logabs
   Default value: 'false'
   When doing indefinite integration where logs are generated, e.g.
   'integrate(1/x,x)', the answer is given in terms of 'log(abs(...))'
   if 'logabs' is 'true', but in terms of 'log(...)' if 'logabs' is
   'false'.  For definite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

2. Wykonywanie obliczeń symbolicznych oraz numerycznych

2.1. Wartości przybliżone, ustalanie precyzji obliczeń

Tabela 2.1

POLECENIA	OPIS
5.33	**kropka** jest separatorem dziesiętnym
5.33e-2	otrzymamy 0.0533, notacja naukowa
5.33b-2	liczba zmiennoprzecinkowa o dowolnej precyzji, ustalanej przez **fpprec**, zapis ten oznacza $5.33\ 10^{-2}$
fpprec: n	ustawianie precyzji obliczeń zmiennoprzecinkowych (n cyfr znaczących)
fpprintprec: n	ustalanie precyzji wydruku (n cyfr lub n znaków)
float(a)	przybliżenie dziesiętne liczby a w zwykłej precyzji
a, **float**	jw.
a**, numer**	podobnie jak wyżej, przybliżenie dziesiętne liczby a
bfloat(a)	przybliżenie dziesiętne liczby a (o dowolnej precyzji ustalanej za pomocą **fpprec**)
a, **bfloat**	jw.
set_display('xml**)**	domyślny format wyświetlania **xml**
set_display('ascii**)**	format wyświetlania **ascii**



Zilustrujemy najpierw działanie opcji i poleceń z **Menu-Numeryczne**.

```
(%i1) sqrt(5);
(%o1) √5
```

```
(%i2) 2^60;
(%o2) 1152921504606846976
```

Numeryczne/Wartość zmiennoprzecinkowa

```
(%i3) float(sqrt(5)), numer;
(%o3) 2.23606797749979
```

```
(%i4) float(2^60), numer;
```
(%o4) $1.152921504606847\ 10^{18}$

Numeryczne/Ustaw dokładność

```
(%i5) fpprec : 3;
(%o5) 3
```

Deklaracja **fpprec:3** działa globalnie.

Numeryczne/To Bigfloat

```
(%i6) bfloat(sqrt(5));
(%o6) 2.24b0
```

```
(%i7) bfloat(2^60),fpprec=2;
(%o7) 1.2b18
```

Deklaracja **fpprec=2** działa lokalnie.
Zapis 1.2b18 oznacza $1.2 \cdot 10^{18}$.

```
(%i8) %pi,bfloat;
(%o8) 3.14b0
```

Skrócona forma zapisu z **bfloat**.
Tu znów działa **fpprec:3**.

Uwaga 1

Standardowo Maxima wykonuje obliczenia symbolicznie. Wybór pierwszej opcji z **Menu-Numeryczne** spowoduje przełączenie wyświetlania z trybu symbolicznego na numeryczny.

Numeryczne/ Wł/wył wyjście numeryczne

```
(%i1) if numer#false then numer:false else numer:true;
(%o1) true
```

```
(%i2) sqrt(3);
(%o2) 1.732050807568877
```

Zadeklarowane jest **numer:true**,
więc otrzymujemy wartość numeryczną.

```
(%i3) fpprintprec:6$
      sqrt(3);
(%o4) 1.73205
```

Precyzja wyświetlania jest ustawiona
teraz na 6 cyfr.

Ponownie wybieramy **Numeryczne/ Wł/wył wyjście numeryczne**

```
(%i5) if numer#false then numer:false else numer:true;
(%o5) false
```

```
(%i6) fpprintprec:5$
      sqrt(3);sqrt(3),numer;
(%o7) √3
(%o8) 1.732
```

Zadeklarowane jest **numer:false**, więc
jest potrzebne polecenie przybliżające.
Ustawiona jest precyzja 5 cyfr, ale ostatnia
cyfra jest zerem i nie wyświetla się.

Uwaga 2

W przypadku gdy liczbę podajemy w postaci dziesiętnej (przypomnijmy, że miejsce dziesiętne jest oddzielone kropką), jest ona traktowana jako wartość przybliżona i obliczenia związane z tą liczbą są wykonywane numerycznie np.:

```
(%i1) fpprintprec:6$
```

```
(%i2) 4/7=4.0/7;
```
(%o2) $\frac{4}{7}=0.5714$

```
(%i3) sqrt(2)=sqrt(2.0);
```
(%o3) $\sqrt{2}=1.41421$

```
(%i4) log(3)+%e=log(3.0)+%e;
```
(%o4) log(3) + %e = %e + 1.09861

```
(%i5) log(3.0)+%e,numer;
(%o5) 3.81689
```

Uwaga 3

Ilość wyświetlanych znaków jest ograniczona. W poniższym przykładzie 98 cyfr zostało pominiętych.

```
(%i1) 100!;
(%o1)
933262154439441526816992388562[98 digits]916864000000000000000000000000
```

Aby wyznaczyć ilość cyfr tak dużej liczby można zastosować polecenie **bfloat**

```
(%i2) 100!,bfloat;
(%o2) 9.332621544394415b157
```

albo też na przykład notację z kropką.

```
(%i3) 100.0!;
```
(%o3) $9.33262154439465810^{157}$

Po przejściu do formatu wyświetlania **ascii** możemy odczytać wartość liczby z bardzo dużą dokładnością albo podać wszystkie cyfry dużej liczby.

```
(%i4) set_display('ascii)$
```

```
(%i5) 100!;
(%o5) 93326215443944152681699238856266700490715968264381621468592963895217599\
93229915608941463976156518286253697920827223758251185210916864000000000000000000\
00000000
```

Po zastosowaniu kolejno poleceń:

```
fpprec:1000$ %e,bfloat$
```

wyświetli się 1000 cyfr liczby e.

Na koniec przywracamy domyślny format **xml**.

```
(%i6) set_display('xml)$
```

ZADANIA PRZYKŁADOWE

Przykład 1

Poniższe wyrażenia zapiszemy w postaci ułamków dziesiętnych z dokładnością do 5 cyfr znaczących.

a) $3^{12} - (\sqrt{45} + 5^8)$, b) $2^6 \cdot \sin 17^\circ$, c) $\sqrt[3]{4} + \ln 5$, d) $\operatorname{arctg} e^{-2} + \cos\frac{\pi}{4}$.

```
(%i1) fpprec : 5$
```

```
(%i2) a:3^12-(sqrt(45)+5^8);
      bfloat(a);
(%o2) 140816-3√5
(%o3) 1.4081b5
```

```
(%i4) b:2^6*sin(17*%pi/180);
      bfloat(b);
(%o4) 64 sin(17π/180)
(%o5) 1.8712b1
```

```
(%i6) c:4^(1/3)+log(5);
      c,bfloat;
(%o6) log(5)+4^(1/3)
(%o7) 3.1968b0
```

```
(%i8) d:atan(%e^(-2))+cos(%pi/4);
      d,bfloat;
(%o8) atan(%e^-2)+1/√2
(%o9) 8.4163b-1
```

Przykład 2

Obliczymy wartości funkcji $f(x) = \sin x$ kolejno w punktach: -6.28, $\frac{5\pi}{3}$, a wyniki zapiszemy w postaci ułamków dziesiętnych z dokładnością do 3 cyfr znaczących.

```
(%i1) fpprec : 3$
```

```
(%i2) f(x):=sin(x);
(%o2) f(x):=sin(x)
```

```
(%i3) f(-6.28);
(%o3) 0.0031853017931379
```

```
(%i4) f(-6.28),bfloat;
(%o4) 3.18b-3
```

```
(%i5) f(5*%pi/3);
(%o5) -√3/2
```

```
(%i6) f(5*%pi/3),bfloat;
(%o6) -8.66b-1
```

Otrzymaliśmy $f(-6.28) = 0{,}00318$ oraz $f\left(\frac{5\pi}{3}\right) = -0.866$.

2.2. Podstawianie oraz deklaracje środowiska lokalnego

Tabela 2.2

POLECENIA	OPIS
subst(a,z,s**)** lub **s, z=a**	zmienna *z* w wyrażeniu *s* zostaje zastąpiona przez zmienną *a*
ev(s,w1,w2…**)** albo **s,w1,w2…**	środowisko lokalne pozwalające na modyfikację wyrażenia *s* zgodnie z podanym warunkiem (warunkami) [1]
ev(s,**nouns)** lub **s,nouns**	wykonanie wszystkich operacji symbolicznych dla *s* [2]
simp: *false*	zatrzymanie upraszczania (domyślnie *true*)
s,simp	przywrócenie symplifikacji dla wyrażenia *s*

[1] Lista przykładowych komend, które mogą być użyte jako warunki w poleceniu **ev**: **pred, float, bfloat, detout, numer, simp, ratsimp, radcan, factor, expand, ratexpand, rectform, demoivre, exponentialize, polarform, trigexpand, trigreduce, solve, sum, simpsum, diff, integrate, nouns.** Większość z nich jest opisana w rozdziale 4.
[2] Maxima rozróżnia operatory typu ***noun*** (symboliczne) i ***verb*** (takie, które można wykonać). Domyślnie wszystkie funkcje są operatorami typu ***verb***. Operator *p* typu ***verb*** można przekształcić w ***noun*** stosując zapis **'(***p***)** (patrz też tabela 1.7).

Poniżej przedstawimy kilka przykładów zastosowania polecenia **ev**.
A) W praktyce najczęściej polecenie **ev** stosuje się, gdy potrzebujemy kilku komend upraszczających (zastosowanych w ustalonej kolejności) lub gdy dane przekształcenie jest częścią większej całości obliczeniowej. W dalszych rozdziałach będziemy raczej używać krótszej notacji z przecinkiem.

```
(%i1) ev(sqrt(3)+sqrt(7)>sqrt(10),pred);
(%o1) true
```

```
(%i2) ev(cos(3*%i),exponentialize);
```
(%o2) $\dfrac{\%e^{3}+\%e^{-3}}{2}$

```
(%i3) ev(cos(2),bfloat,fpprec=7);
(%o3) -4.161468b-1
```

```
(%i4) ev(x^4-11*x^3+45*x^2-81*x+54,factor);
```
(%o4) $(x-3)^{3}\,(x-2)$

B) Polecenie **ev** można wykorzystać do łączenia różnych operacji obliczeniowych i przekształcających oraz do obserwacji etapów pośrednich obliczeń.

(%i5) w:sin(3 * x) / cos(x)^2;

(%o5) $\frac{\sin(3\,x)}{\cos(x)^2}$

(%i6) ev(w, trigexpand);

(%o6) $\frac{3\cos(x)^2\sin(x)-\sin(x)^3}{\cos(x)^2}$

(%i7) w, trigexpand,ratexpand;

(%o7) $3\sin(x)-\frac{\sin(x)^3}{\cos(x)^2}$

(%i8) a*sin(omega*t+phi);

(%o8) $a\sin(\omega\,t+\phi)$

(%i9) ev(%,a=2,phi=%pi/3);

(%o9) $2\sin\left(\omega\,t+\frac{\pi}{3}\right)$

(%i10) %,omega=1,t=%pi/6;

(%o10) 2

(%i11) 'diff(x^3,x,1);

(%o11) $\frac{d}{d\,x}x^3$

(%i12) ev(%,diff);

(%o12) $3\,x^2$

(%i13) 'integrate(x^5, x,a,b);

(%o13) $\int_a^b x^5\,dx$

(%i14) ev(%,integrate,a=1,b=2);

(%o14) $\frac{21}{2}$

(%i15) 'limit(n/(2-n),n,inf);

(%o15) $\lim_{n->\infty}\frac{n}{2-n}$

(%i16) ev(%,nouns);

(%o16) -1

C) Rozwiązania równań uzyskane za pomocą polecenia **solve** (więcej o rozwiązywaniu równań w rozdziale 7) są przedstawiane w postaci równości. Poniżej przykłady ilustrujące, jak rozwiązania te mogą być wykorzystane do dalszych obliczeń po użyciu polecenia **ev**.

(%i1) rownanie1:2*x^3+3*x^2-3*x-2=0$

(%i2) rozwiazanie:solve(rownanie1);

(%o2) $[x=-\frac{1}{2},x=-2,x=1]$

Obliczymy teraz wartość wyrażenia $x\log\left(3-\frac{x}{3}\right)$ dla x równego drugiemu z podanych pierwiastków równania $2x^3+3x^2-3x-2=0$.

(%i3) ev(x*log(3-x/2), rozwiazanie[2]);

(%o3) $-2\log(4)$

Wykonujemy podstawienie.

(%i4) ev(rownanie1,rozwiazanie[1],pred);

(%o4) true

Sprawdzamy, czy po wykonaniu podstawienia otrzymamy zdanie prawdziwe.

```
(%i1) rownanie2:log(x)^2-4*log(x)+1=0;
```
(%o1) $\log(x)^2-4\log(x)+1=0$

```
(%i2) rozwiazanie:solve(rownanie2);
```
(%o2) $[x=\%e^{2-\sqrt{3}}, x=\%e^{\sqrt{3}+2}]$

```
(%i3) ev(x,rozwiazanie[2]);
```
(%o3) $\%e^{\sqrt{3}+2}$

Wybieramy drugie spośród rozwiązań.

```
(%i4) ev(rownanie2,rozwiazanie[2]);
```
(%o4) $(\sqrt{3}+2)^2-4(\sqrt{3}+2)+1=0$

Podstawiamy drugie rozwiązanie do *rownanie2.*

```
(%i5) ev(%,expand);
```
(%o5) $0=0$

Rozwijamy lewą stronę równania i otrzymujemy tożsamość.

```
(%i6) ev(rownanie2,rozwiazanie[2],pred);
(%o6) false
```

Tu Maxima „nie radzi sobie" z porównaniem.

```
(%i7) ev(rownanie2,rozwiazanie[2],expand,pred);
(%o7) true
```

Dodanie polecenia **expand** powoduje, że otrzymujemy poprawny wynik.

ZADANIA PRZYKŁADOWE

Przykład 1

Sprawdzimy działanie poleceń przedstawionych w tabeli 2.2 dla wyrażenia $mx+n$.

```
(%i1) kill(all)$
```

Najpierw „czyścimy" zmienne.

```
(%i1) s:m*x+n;
```
(%o1) $m\,x+n$

Wykonujemy przypisanie do zmiennej s.

```
(%i2) subst(2,m,s);
```
(%o2) $2\,x+n$

Podstawiamy za m wartość 2.

```
(%i3) subst(-1,n,s);
```
(%o3) $m\,x-1$

Podstawiamy za n wartość -1.

Zauważmy, że podstawiać w ten sposób możemy tylko za jedną zmienną. Dalsze przykłady pokażą, jak deklarować wartości zmiennych lokalnie.

```
(%i4) block([m:2,n:-1],m*x+n);
(%o4) 2 x - 1
```

W poleceniu **block** (rozdział 21) możemy zadeklarować zmienne lokalne, a następnie wykonać obliczenia i przekształcenia.

```
(%i5) ev(s,m=2,n=-1);
(%o5) 2 x - 1
```

Efekt „zwykłego" podstawienia (też działającego jedynie lokalnie).

```
(%i6) m:3$
```

Przypisujemy wartość globalnie.

```
(%i7) s,m:2,n:-1;
(%o7) 2 x - 1
```

Podobnie jak w (%i5), tu również przypisanie działa lokalnie.

```
(%i8) m*x+n;
(%o8) 3 x + n
```

Weryfikujemy.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Poniższe wyrażenia zapisać w postaci ułamków dziesiętnych z dokładnością do 4 cyfr znaczących:

a) $2^{10} - \sqrt[3]{900}$, b) $\sqrt{6} + \log 200$, c) $\cos 130^\circ$, d) $\operatorname{arctg} e^2 + \sin\frac{\pi}{5}$.

Zadanie 2

Obliczyć wartości funkcji $f(x) = \operatorname{tg} x$ kolejno w punktach: -2.16, $\frac{\pi}{6}$ i wyniki zapisać w postaci ułamków dziesiętnych z dokładnością do 3 cyfr znaczących.

Zadanie 3

Jaki jest rząd wielkości następujących liczb: a) 2^{400},

b) $k = \frac{F_e}{F_g}$, czyli stosunek oddziaływania sił elektrycznych do grawitacyjnych, jeżeli $F_e = \frac{q^2}{4\pi\varepsilon_0 r^2}$, $F_g = G\frac{m^2}{r^2}$, przy czym wielkość ładunku elektrycznego $q = 1.6 \cdot 10^{-19}\ C$, masa elektronu $m = 9.1 \cdot 10^{-31}\ kg$, stała grawitacyjna $G = 6.67 \cdot 10^{-11}\ \frac{m^3}{kg \cdot s^2}$, przenikalność dielektryczna próżni $\varepsilon_0 = 8.85 \cdot 10^{-12}\ \frac{F}{m}$,

c) $\binom{49}{6}$, czyli liczba kombinacji w standardowym zakładzie Lotto,

d) $\binom{49}{6}$ oraz $\binom{64}{7}$.Wygrana w jakim zakładzie jest bardziej prawdopodobna: $\binom{49}{6}$ czy $\binom{64}{7}$?

Odpowiedzi

1. a) $1.014 \cdot 10^3$, b) 4.751, c) $-6.428 \cdot 10^{-1}$, d) 2.024.
2. $f(-2.16) \approx 1.5$, $f\left(\frac{\pi}{6}\right) \approx 5.77 \cdot 10^{-1}$.
3. a) 10^{120}, b) 10^{42}, c) 10^7, d) w zakładzie $\binom{49}{6}$.

3. Definiowanie funkcji

Tabela 3.1

POLECENIA	OPIS
f(x):=	definiuje funkcję f jednej zmiennej
f(x1,x2,…xn):=	definiuje funkcję f zależną od n zmiennych
define(f(x),p**)**	definiuje funkcję f zmiennej x, po uprzednim wykonaniu operacji p (p może być np. funkcją diff(…), integrate(…) lub poleceniem upraszczającym) [1)]
f(x):= ' '(p)	jw.
functions	podaje listę funkcji zdefiniowanych przez użytkownika w bieżącej sesji pracy z Maximą
remfunction(f,g**)**	usuwa definicje podanych funkcji lub wszystkie (**all**)

[1)] Wykonanie (ewaluacja) operacji p następuje już w inpucie.

ZADANIA PRZYKŁADOWE

Przykład 1

Porównamy dwa sposoby definiowania funkcji zmiennej x, która jest pochodną funkcji arcus tangens.

```
(%i1) f1(x):=diff(atan(x),x);
(%o1) f1(x):=diff(atan(x),x)
```

```
(%i2) f1(x);
```
(%o2) $\frac{1}{x^2+1}$

```
(%i3) f1(1);
diff: second argument must be a variable; found 1
#0: f1(x=1)
 -- an error. To debug this try: debugmode(true);
```

```
(%i4) f2(x):=''(diff(atan(x),x));
```
(%o4) $f2(x):=\frac{1}{x^2+1}$

```
(%i5) f2(x);
```
(%o5) $\frac{1}{x^2+1}$

```
(%i6) f2(1);
```
(%o6) $\frac{1}{2}$

Tylko drugi sposób definiowania pozwala na bezpośrednie obliczanie wartości funkcji. Zauważmy, że taki sam efekt jak przez zastosowanie operatora ' ' uzyskamy, stosując polecenie **define**.

```
(%i7) define(f2_podobnie(x),diff(atan(x),x));f2_podobnie(1);
```
(%o7) $f2\_podobnie(x):=\frac{1}{x^2+1}$

(%o8) $\frac{1}{2}$

Stosując poniższe polecenie, otrzymamy wszystkie funkcje zdefiniowane w trakcie danej sesji pracy z Maximą.

```
(%i9) functions;
(%o9) [f1(x),f2(x),f2_podobnie(x)]
```

W dalszych rozdziałach będziemy często korzystać z operatora ′ ′.

Przykład 2

Zdefiniujemy kilka przydatnych funkcji.

Sinus dla argumentu podanego w stopniach

```
(%i1) sind(x):=sin(x*%pi/180);
```

(%o1) $\mathrm{sind}(x):=\sin\left(\frac{x\,\pi}{180}\right)$

```
(%i2) sind(120);
```

(%o2) $\frac{\sqrt{3}}{2}$

Logarytm przy podstawie 2
(jako funkcję jednej zmiennej)

```
(%i1) log2(x):=log(x)/log(2);
```

(%o1) $\mathrm{log2}(x):=\frac{\log(x)}{\log(2)}$

```
(%i2) log2(8); radcan(%);
```

(%o2) $\frac{\log(8)}{\log(2)}$

(%o3) 3

Logarytm przy podstawie p
(jako funkcję dwóch zmiennych)

```
(%i1) Log(x,p):=log(x)/log(p);
```

(%o1) $\mathrm{Log}(x,p):=\frac{\log(x)}{\log(p)}$

```
(%i2) Log(8,2);%,radcan;
```

(%o2) $\frac{\log(8)}{\log(2)}$

(%o3) 3

Zauważmy, że w ostatnim przykładzie nazwa funkcji rozpoczyna się dużą literą (Maxima rozróżnia wielkość liter, a nazw funkcji wbudowanych nie możemy powtarzać).

Uwaga 1

Możemy wyszukiwać nazwy i automatycznie uzupełniać składnię samodzielnie zdefiniowanych przez nas funkcji, procedur, czy poleceń (przypomnijmy skróty klawiaturowe Ctrl-K oraz Ctrl-Shift-K, patrz również rozdział 1.2).

```
--> Log(<x>,<p>)
```

Możemy też zbudować pakiet (moduł z definicjami funkcji) z rozszerzeniem **mac** i wczytać go przez **Menu-Plik/ Wczytaj pakiet** lub polecenie **load**(".....”).

Przykład 3

Pokażemy jak definiować funkcje sklejane:

a) $\mathrm{modul}(x) = \begin{cases} -x \text{ dla } x < 0 \\ x \text{ dla } x \geq 0 \end{cases}$, b) $\mathrm{znak}(x) = \begin{cases} -1 \text{ dla } x < 0 \\ 0 \text{ dla } x = 0 \\ 1 \text{ dla } x > 0 \end{cases}$.

Istnieją w Maximie wbudowane funkcje **abs** oraz **signum** (patrz tabela 1.6). Jednak, aby pokazać jak można zdefiniować funkcje sklejane, w tym przykładzie użyjemy konstrukcji warunkowej **if** (patrz tabela 21.1).

a)

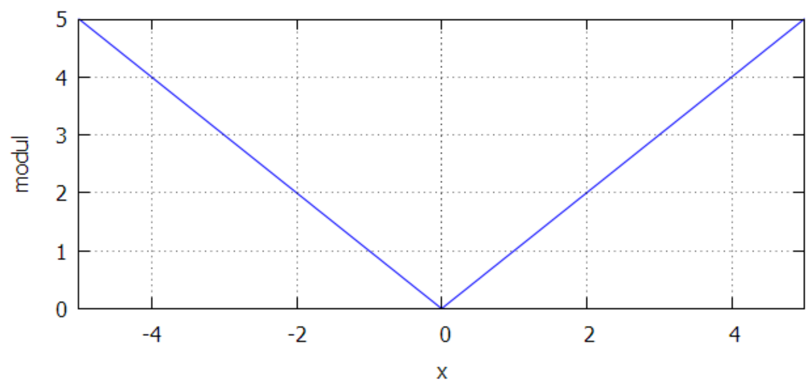
```
(%i1) modul(x):=if x>=0 then x else -x$
```

```
(%i2) modul(3);
(%o2) 3
```

```
(%i3) modul(-3);
(%o3) 3
```

```
(%i4) plot2d(['modul], [x,-5,5],grid2d,[yx_ratio,0.4])$
```

Zauważmy, że nazwę funkcji **modul** możemy tu pisać bez argumentu.
W rozdziale 6 znajduje się więcej informacji o opcjach poleceń graficznych.



b)

```
(%i1) znak(x):=if x<0 then -1 elseif x=0 then 0 else 1$
```

```
(%i2) map(znak,[-5,-2,0,4]);
(%o2) [-1,-1,0,1]
```

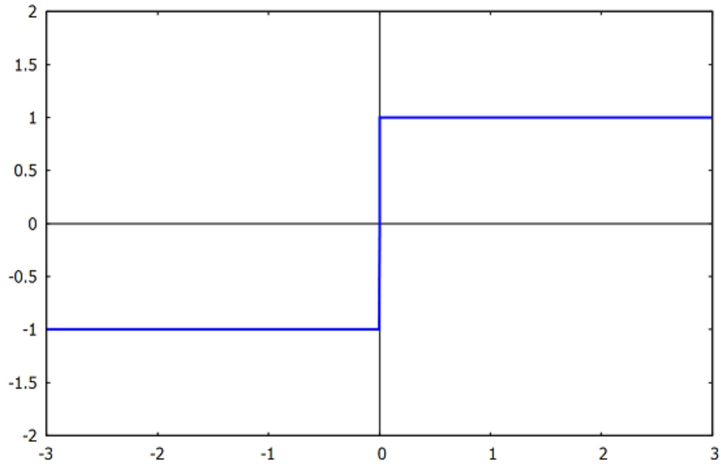
Polecenie **map** (patrz tabela 8.1) pozwala zastosować funkcję znak do kilku argumentów równocześnie.

Tym razem do naszkicowania funkcji zastosujemy pakiet **draw** i jego polecenia (patrz rozdział 6.2). Po wczytaniu pakietu najpierw wprowadzimy kilka ustawień domyślnych za pomocą **set_draw_defaults(…)**.

```
(%i3) load(draw)$
      set_draw_defaults(
      xrange=[-3,3],yrange=[-2,2],line_width=2,xaxis=true,yaxis=true,
      xaxis_type=solid,yaxis_type=solid,proportional_axes=xy )$
```

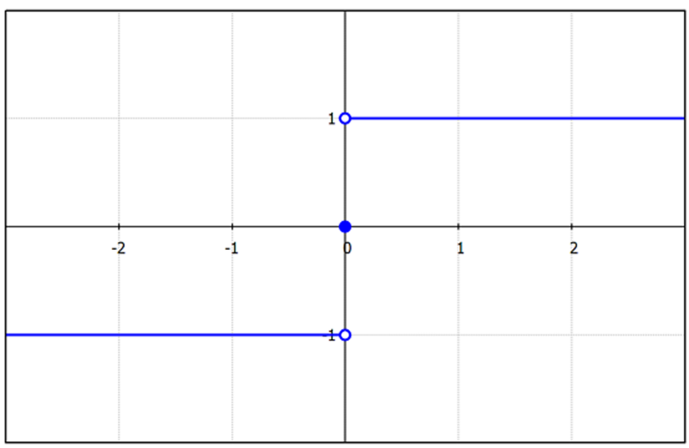
Najpierw narysujemy funkcję **znak** przy powyższych ustawieniach.

```
(%i5) draw2d(explicit(znak(x),x,-3,3))$
```



Następnie narysujemy funkcję **znak** „po kawałku”, wyraźnie zaznaczając punkt nieciągłości.

```
(%i6) draw2d(xtics_axis=true,xtics=[-2,1,2], ytics_axis=true,ytics={-1,1},
        explicit(-1,x,-3,0),explicit(1,x,0,3),grid=true,
        point_size=1.4,point_type=7,points([0,0,0],[-1,0,1]),
        point_size=0.7,color=white, points([0,0],[-1,1]))$
```



Przykład 4

Zdefiniujemy pochodną i całkę funkcji $f(x) = x^n$ jako funkcję zmiennej n.

Najpierw przywołajmy oba wzory stosując operator ‘, który „zatrzymuje obliczenia”. Symbolicznie przedstawia się to następująco:

```
(%i1) 'diff(x^n,x)=diff(x^n,x);
```
(%o1) $\frac{d}{d\,x}x^n = n\,x^{n-1}$

```
(%i2) 'integrate(x^n,x)=integrate(x^n,x);
Is n equal to -1?n;
```
(%o2) $\int x^n\,dx = \frac{x^{n+1}}{n+1}$

```
(%i3) 'integrate(x^n,x)=integrate(x^n,x);
Is n equal to -1?y;
```
(%o3) $\int x^n\,dx = \log(x)$

W przypadku całki pojawiło się zapytanie o założenia dotyczące n. Odpowiedzieliśmy raz negatywnie, tzn. $n \neq -1$, a za drugim razem pozytywnie, tzn. że $n = -1$. Odpowiedzi, które akceptuje Maxima w tym zapytaniu, to: potwierdzające (yes, y, Y), negujące (no, n, N).

Teraz zdefiniujemy funkcje zmiennej n:

```
(%i4) pochodna(n):=''(diff(x^n,x));
```
(%o4) pochodna(n):= $n\,x^{n-1}$

```
(%i5) assume(notequal(n,-1));
```
(%o5) [notequal(n, -1)]

```
(%i6) calka(n):=''(integrate(x^n,x));
```
(%o6) calka(n):= $\frac{x^{n+1}}{n+1}$

Zanim zdefiniujemy funkcję z całką, z góry przyjmiemy założenie $n \neq -1$, by zapobiec zapytaniom Maximy. Zastosujemy w tym celu polecenie **assume**.

Zdefiniujmy jeszcze funkcję potęgową w zależności od n i porównamy działanie wszystkich funkcji.

```
(%i7) funkcja(n):=x^n;
```
(%o7) funkcja(n):= x^n

```
(%i8) funkcja(1);pochodna(1);calka(1);
```
(%o8) x

(%o9) 1

(%o10) $\frac{x^2}{2}$

```
(%i11) funkcja(2);pochodna(2);calka(2);
```
(%o11) x^2

(%o12) $2\,x$

(%o13) $\frac{x^3}{3}$

Możemy użyć też notacji wektorowej.

```
(%i1) kill(labels)$
```

Resetowanie numeracji etykiet.

```
(%i1) [funkcja(n),pochodna(n),calka(n)],n=100;
```

(%o1) $[x^{100}, 100\,x^{99}, \frac{x^{101}}{101}]$

```
(%i2) [funkcja(n),pochodna(n),calka(n)],n=-5;
```

(%o2) $[\frac{1}{x^5}, -\frac{5}{x^6}, -\frac{1}{4\,x^4}]$

```
(%i3) [funkcja(n),pochodna(n),calka(n)],n=1/4;
```

(%o3) $[x^{1/4}, \frac{1}{4\,x^{3/4}}, \frac{4\,x^{5/4}}{5}]$

Przykład 5

Dane są funkcje $f(x) = e^x$ oraz $g(x) = x^2 + 1$. Zilustrujemy na ich przykładzie działanie $'$ oraz $''$, przy wyznaczaniu pochodnych funkcji złożonych: $h1(x) = f(g(x))$, $h2(x) = g(h(x))$.

```
(%i1) f(x):=exp(x)$
      g(x):=x^2+1$
```

```
(%i3) 'f(g(x));
```

(%o3) $f(x^2+1)$

```
(%i4) f('g(x));
```

(%o4) $\%e^{g(x)}$

```
(%i5) f(g(x));
```

(%o5) $\%e^{x^2+1}$

```
(%i6) f(g(0));
```

(%o6) $\%e$

```
(%i7) 'g(f(x));
```

(%o7) $g(\%e^x)$

```
(%i8) g('f(x));
```

(%o8) $f(x)^2+1$

```
(%i9) g(f(x));
```

(%o9) $\%e^{2\,x}+1$

```
(%i10) g(f(0));
```

(%o10) 2

```
(%i11) h1(x):=''(f(g(x))); diff(h1(x),x);
```

(%o11) $h1(x):=\%e^{x^2+1}$

(%o12) $2\,x\,\%e^{x^2+1}$

```
(%i13) h2(x):=''(g(f(x))); diff(h2(x),x);
```

(%o13) $h2(x):=\%e^{2\,x}+1$

(%o14) $2\,\%e^{2\,x}$

```
(%i15) diff(f('g(x)),x);
```

(%o15) $\%e^{g(x)}\left(\frac{d}{d\,x}\,g(x)\right)$

```
(%i16) diff(g('f(x)),x);
```

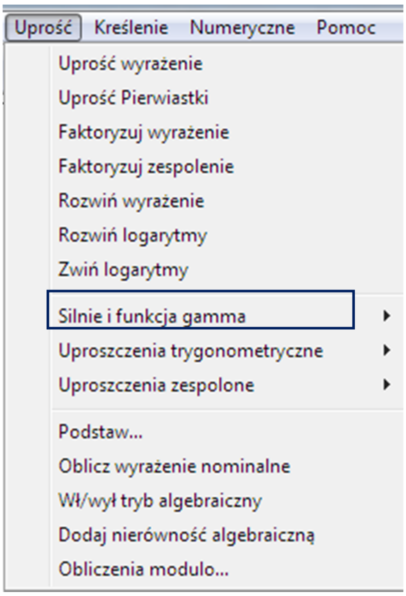
(%o16) $2\,f(x)\left(\frac{d}{d\,x}\,f(x)\right)$

4. Przekształcanie oraz upraszczanie wyrażeń algebraicznych

Polecenia upraszczające dostępne z palety **Podstawowa Matematyka**:

ratsimp(%**)**	**radcan(**%**)**
factor(%**)**	**expand(**%**)**
rectform(%**)**	**subst(**y,x,%**)**
trigrat(%**)**	**trigsimp(**%**)**
trigexpand(%**)**	**trigreduce(**%**)**

Polecenia upraszczające dostępne w **Menu-Uprość**:



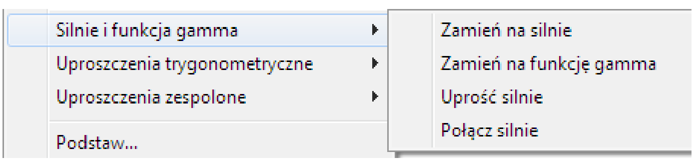
ratsimp(%**)**
radcan(%**)**
factor(%**)**
gfactor(%**)**
expand(%**)**
%,**logexpand=super**
logcontract(%**)**

Patrz poniżej.

Takie jak w panelu **Podstawowa Matematyka**.

Omówione zostały w rozdziale 5.

ev(%, **nouns)**
algebraic:**not(**algebraic**)**



makefact(%**)**
makegamma(%**)**
minfactorial(%**)**
factcomb(%**)**

Uwaga 1

Większość poleceń przytoczonych w poniższych tabelach można wywołać dwojako tzn. w postaci funkcji **polecenie(***w***)**, jak również w formie: *w*, **polecenie**. W zamieszczonych w tym rozdziale przykładach będziemy stosować oba sposoby wywołania.

Tabela 4.1

POLECENIA	OPIS
ordergreat(a,b,c**)**	ustala porządek wyświetlania zmiennych
unorder()	odwołuje powyższą deklarację i przywraca ustawienia domyślne, tzn. odwrotny porządek alfabetyczny
declare(x,**mainvar**)	ustala zmienną x jako główną zmienną

```
(%i1) a-b-c;
(%o1) -c-b+a

(%i2) ordergreat(a,b,c)$
      a-b-c;
(%o3) a-b-c

(%i4) unorder();
(%o4) [c,b,a]
```

```
(%i5) (a+x)^2+y;
(%o5) y+(x+a)^2

(%i6) declare(a,mainvar)$

(%i7) (a+x)^2+y;
(%o7) (a+x)^2+y
```

Tabela 4.2

POLECENIA	OPIS
factor(n**)**	rozkłada liczbę całkowitą n na czynniki pierwsze
ifactors(n**)**	zwraca czynniki pierwsze i ich potęgi dla liczby n
divide(m,n**)**	zwraca iloraz oraz resztę z dzielenia m przez n
remainder(m,n**)** lub **mod(**m,n**)**	zwraca resztę z dzielenia liczby m przez n
gcd(m,n**)**	NWD(m,n) – największy wspólny dzielnik
lcm(m,n**)**	NWW(m,n) – najmniejsza wspólna wielokrotność

```
(%i1) divide(1001,7);
      factor(1001)= 7*11*13;
      ifactors(1001);
(%o1) [143,0]
(%o2) 7 11 13 =1001
(%o3) [[7,1],[11,1],[13,1]]
```

```
(%i7) 1/128+1/56;56/128;
(%o7) 23/896
(%o8) 7/16
```

```
(%i4) display(mod(19,5))$
      display(gcd(128,56))$
      display(lcm(128,56))$
mod(19,5)=4
gcd(128,56)=8
lcm(128,56)=896
```

```
(%i9) display(mod(128,56))$
      display(mod(56,16))$
      display(mod(16,8))$
mod(128,56)=16
mod(56,16)=8
mod(16,8)=0
```

(%i9) jest ilustracją algorytmu Euklidesa zastosowanego do liczb 128 oraz 56. Największym wspólnym dzielnikiem jest ostatnia niezerowa reszta, tzn. liczba 8.

Tabela 4.3

POLECENIA	OPIS
minfactorial(w)	upraszcza silnie
factcomb(w)	łączy silnie
makefact(w)	wyraża funkcję Gamma oraz symbole typu $\binom{n}{k}$ za pomocą symbolu !
makegamma(w)	wyraża silnie za pomocą funkcji Gamma (**Γ**)
gamma(z)	dla liczb naturalnych $\Gamma(n+1) = n!$ dla zespolonych $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt\,,\ re(z) > 0$

(%i7) `a[n+1];`

(%o7) $\dfrac{(n+1)!\,(n+4)!}{(n+2)!^2}$

(%i8) `minfactorial(a[n+1]/a[n]);`

(%o8) $\dfrac{(n+1)(n+4)}{(n+2)^2}$

(%i9) `(2*n+2)*(n+2)*(n!);`
`factcomb(%);`

(%o9) $(n+2)(2\,n+2)\,n!$

(%o10) $2\,(n+2)!$

(%i11) `(n+1)^2*(n!);`
`factcomb(%);`

(%o11) $(n+1)^2\,n!$

(%o12) $(n+2)!-(n+1)!$

(%i1) `gamma(2*n+1);`
`makefact(%);`

(%o1) $\Gamma(2\,n+1)$

(%o2) $(2\,n)!$

(%i3) `binomial(n,k);`
`makefact(%);`
`makegamma(%);`

(%o3) $\binom{n}{k}$

(%o4) $\dfrac{n!}{k!\,(n-k)!}$

(%o5) $\dfrac{\Gamma(n+1)}{\Gamma(k+1)\,\Gamma(n-k+1)}$

(%i6) `a[n]:=(n+3)!*n!/((n+1)!)^2;`

(%o6) $a_n := \dfrac{(n+3)!\,n!}{(n+1)!^2}$

(%i13) `[gamma(2.3),1.3!,gamma(sqrt(2))];`

(%o13) $[1.16671190519816, 1.16671190519816, \Gamma(\sqrt{2})]$

Tabela 4.4

POLECENIA	OPIS
divide(w,p)	iloraz i reszta z dzielenia wielomianu w przez wielomian p zwracane w formie listy
quotient(w,p)	iloraz z dzielenia wielomianu w przez wielomian p
remainder(w,p)	reszta z dzielenia wielomianu w przez wielomian p
gcd(w,p)	NWD(w,p), gdzie w oraz p są wielomianami
lcm(w,p)	NWW(w,p), gdzie w oraz p są wielomianami
coeff(p,x,n)	współczynnik wielomianu przy x^n, gdzie p jest wielomianem

POLECENIA	OPIS
ratcoef(p,x,n)	współczynnik stojący przy x^n (po uproszczeniu p), gdzie p jest wyrażeniem wymiernym
denom(w)	mianownik wyrażenia wymiernego w (lub liczby wymiernej) po uproszczeniu ułamka
num(w)	licznik wyrażenia wymiernego w (lub liczby wymiernej) po uproszczeniu ułamka
horner(p)	grupuje wielomian p wg schematu Hornera (redukcja mnożeń do minimum)

1)

```
(%i1) wielomian:x^5-3*x^4+x^2+4*x +1;
      coeff(wielomian,x,4);
```

(%o1) $x^5-3x^4+x^2+4x+1$

(%o2) -3

```
(%i3) makelist(coeff(wielomian,x,i),i,[5,4,3,2,1,0]);
```

(%o3) $[1,-3,0,1,4,1]$

2)

```
(%i4) W(x):=x^4+3*x^2+3$   P(x):=x^2+x$
```

```
(%i6) I(x):=''(quotient(W(x),P(x)))   /*iloraz*/;
```

(%o6) $\mathrm{I}(x):=x^2-x+4$

```
(%i7) R(x):=''(remainder(W(x),P(x)))   /*reszta*/;
```

(%o7) $\mathrm{R}(x):=3-4x$

```
(%i8) divide(W(x),P(x))   /*[iloraz,reszta]*/;
```

(%o8) $[x^2-x+4,3-4x]$

```
(%i9) W(x)=P(x)*I(x)+R(x);
```

(%o9) $x^4+3x^2+3=(x^2-x+4)(x^2+x)-4x+3$

```
(%i10) W(x)/P(x)=I(x)+R(x)/P(x);
```

(%o10) $\dfrac{x^4+3x^2+3}{x^2+x}=\dfrac{3-4x}{x^2+x}+x^2-x+4$

3)

```
(%i11) w:(x-1)*y+y^2/x+(x+y)/2;
       ratcoef(w,x,1);
```

(%o11) $\dfrac{y^2}{x}+\dfrac{y+x}{2}+(x-1)\,y$

(%o12) $\dfrac{2y+1}{2}$

4)

```
(%i13) q:x^4-x^3+2*x^2+3*x-4$ horner(q,x);
```

(%o14) $x\,(x\,((x-1)\,x+2)+3)-4$

Tabela 4.5

POLECENIA	OPIS
expand(w)	rozwija wyrażenia (rozwija potęgi oraz iloczyny sum)
collectterms(w,z)	porządkuje wyrażenie *w* względem zmiennych *z*
ratexpand(w)	rozwija wyrażenia i upraszcza ułamki
ratsimp(w)	upraszcza wyrażenia wymierne (sprowadza do wspólnego mianownika, upraszcza ułamki)
radcan(w)	upraszcza wyrażenia potęgowe, wykładnicze i logarytmiczne
combine(w)	sumuje wyrażenia o jednakowych mianownikach
distrib(w)	rozdziela na sumę ułamków względem licznika
factor(w)	rozkłada wielomian na czynniki w dziedzinie $\mathbb{R}$
gfactor(w)	rozkłada wielomian na czynniki w dziedzinie $\mathbb{C}$
facsum(w,z)	grupuje wyrażenie *w* względem zmiennej *z*
rat(w)	przekształca liczby, z postaci zmiennoprzecinkowej do ułamków zwykłych, oraz upraszcza wyrażenia wymierne
rat(w,a,b, ...**)**	jw. porządkujące kolejno względem zmiennych $a,b,\ldots$
xthru(w)	sprowadza wszystkie składniki wyrażenia *w* do wspólnego mianownika
partfrac(w,x)	rozkłada wyrażenie wymierne *w* na sumę ułamków prostych względem zmiennej *x*
rootscontract(w)	zamienia iloczyn (iloraz) pierwiastków (potęg) na pierwiastek iloczynu (ilorazu)
sqrtdenest(w)	upraszcza pierwiastki, wymaga pakietu **sqdnst**

1)

```
(%i1) (b+(a+1)^2)*(2*b-a);
      expand(%);
      factor(%);
      factorsum(%);
```

(%o1) $\left(b+(a+1)^2\right)(2\,b-a)$

(%o2) $2\,b^2+2\,a^2\,b+3\,a\,b+2\,b-a^3-2\,a^2-a$

(%o3) $(b+a^2+2\,a+1)(2\,b-a)$

(%o4) $\left(b+(a+1)^2\right)(2\,b-a)$

2)

```
(%i5) -sin(x)*B-cos(x)*B-sin(x)*A+cos(x)*A;  rat(%,cos(x),sin(x));
```

(%o5) $-\sin(x)\,B-\cos(x)\,B-\sin(x)\,A+\cos(x)\,A$

(%o6)/R/ $(-B-A)\sin(x)+(-B+A)\cos(x)$

3)

```
(%i7) w:(x-2*a+y)^2;
      w,expand;collectterms(%,y);
      ratcoef(w, a*y);
      rat(w, x);rat(w,a);
```

(%o7) $(y+x-2\,a)^2$

(%o8) $y^2+2\,x\,y-4\,a\,y+x^2-4\,a\,x+4\,a^2$

(%o9) $y^2+(2\,x-4\,a)\,y+x^2-4\,a\,x+4\,a^2$

(%o10) -4

(%o11)/R/ $x^2+(2\,y-4\,a)\,x+y^2-4\,a\,y+4\,a^2$

(%o12)/R/ $4\,a^2+(-4\,y-4\,x)\,a+y^2+2\,x\,y+x^2$

4)

```
(%i13) w:(x-1)^2/(x*(x+1));
```

(%o13) $\dfrac{(x-1)^2}{x\,(x+1)}$

```
(%i14) expand(w);
```

(%o14) $\dfrac{x^2}{x^2+x}-\dfrac{2\,x}{x^2+x}+\dfrac{1}{x^2+x}$

```
(%i15) ratexpand(w);
```

(%o15) $\dfrac{1}{x^2+x}+\dfrac{x}{x+1}-\dfrac{2}{x+1}$

```
(%i16) expandwrt(w,x);
```

(%o16) $\dfrac{x}{x+1}+\dfrac{1}{x\,(x+1)}-\dfrac{2}{x+1}$

```
(%i17) xthru(%);
```

(%o17) $\dfrac{x^2-2\,x+1}{x\,(x+1)}$

```
(%i18) distrib(%);
```

(%o18) $\dfrac{x}{x+1}+\dfrac{1}{x\,(x+1)}-\dfrac{2}{x+1}$

```
(%i19) combine(%);
```

(%o19) $\dfrac{x-2}{x+1}+\dfrac{1}{x\,(x+1)}$

```
(%i20) partfrac(w,x);
```

(%o20) $-\dfrac{4}{x+1}+\dfrac{1}{x}+1$

5)

```
(%i21) (b+(a+1)^2)*(2*b-a);
       expand(%);
       factor(%);
       factorsum(%);
```

(%o21) $\left(b+(a+1)^2\right)(2\,b-a)$

(%o22) $2\,b^2+2\,a^2\,b+3\,a\,b+2\,b-a^3-2\,a^2-a$

(%o23) $(b+a^2+2\,a+1)(2\,b-a)$

(%o24) $\left(b+(a+1)^2\right)(2\,b-a)$

6)

```
(%i25) 4*x^3+x,factor;
```

(%o25) $x\,(4\,x^2+1)$

```
(%i26) gfactor(4*x^3+x);
```

(%o26) $x\,(2\,x-\%i)(2\,x+\%i)$

7)

```
(%i27) (%e^(-x)+1)/(%e^(x)+1);
       %,radcan;
```

(%o27) $\dfrac{\%e^{-x}+1}{\%e^{x}+1}$

(%o28) $\%e^{-x}$

```
(%i31) (x+4*sqrt(x)+4)/(2+sqrt(x));
       %,radcan;
```

(%o31) $\dfrac{x+4\sqrt{x}+4}{\sqrt{x}+2}$

(%o32) $\sqrt{x}+2$

```
(%i29) (log(x^2+x)-log(x+1))^3/log(x)^2;
       %,radcan;
```

(%o29) $\dfrac{(\log(x^2+x)-\log(x+1))^3}{\log(x)^2}$

(%o30) $\log(x)$

8)

```
(%i1) (sqrt(2+sqrt(3)))*(sqrt(2-sqrt(3)));
      %,rootscontract;
```

(%o1) $\sqrt{2-\sqrt{3}}\,\sqrt{\sqrt{3}+2}$

(%o2) 1

9) Opcja **algebraic** pozwala na przekształcanie pierwiastków. Można ustawić ją globalnie (**algebraic:true**) albo lokalnie, tak jak w poniższych przykładach.

```
(%i3) [1/sqrt(3),ev(rat(1/sqrt(3)),algebraic)];
```

(%o3)/R/ $[\dfrac{1}{\sqrt{3}},\dfrac{\sqrt{3}}{3}]$

```
(%i4) [8/sqrt(2),rat(8/sqrt(2)),ev(rat(8/sqrt(2)),algebraic)];
```

(%o4)/R/ $[2^{5/2},\sqrt{2}^5,4\sqrt{2}]$

```
(%i5) w:(x+1)/(sqrt(x)-1);
      w,ratsimp,algebraic;
      %,factor;
```

(%o5) $\dfrac{x+1}{\sqrt{x}-1}$

(%o6) $\dfrac{\sqrt{x}\,(x+1)+x+1}{x-1}$

(%o7) $\dfrac{(\sqrt{x}+1)(x+1)}{x-1}$

```
(%i8) 1/(2-sqrt(3));
      ratsimp(%),algebraic;
```

(%o8) $\dfrac{1}{2-\sqrt{3}}$

(%o9) $\sqrt{3}+2$

10)

```
(%i10) load(sqdnst)$
       p:sqrt(7-4*sqrt(3))$
       p=sqrtdenest(p);
```
(%o12) $\sqrt{7-4\sqrt{3}}=2-\sqrt{3}$

```
(%i13) (2-sqrt(3))^2;expand(%);
```
(%o13) $(2-\sqrt{3})^2$

(%o14) $7-4\sqrt{3}$

Tabela 4.6

POLECENIA	OPIS
logcontract(w)	zwija wyrażenie logarytmiczne *w*
logexpand(w)	rozwija wyrażenie logarytmiczne *w*
w,**logexpand=super**	mocniejsza wersja polecenia **logexpand**
logarc(w)	przy całkowaniu pewnych funkcji niewymiernych zamienia funkcje cyklometryczne (hiperboliczne) na równoważne funkcje z logarytmem
logarc:true	jw. (ale ustawienie działa globalnie)
logabs:true	przy całkowaniu funkcji wymiernych pojawiają się moduły w logarytmach (domyślnie jest: false)
integrate(...),**logabs**	jw. (ale ustawienie działa lokalnie)

1)

```
(%i1) log(x)-2*log(x+1);
      %,logcontract;
```
(%o1) $\log(x)-2\log(x+1)$

(%o2) $\log\left(\frac{x}{(x+1)^2}\right)$

```
(%i3) log(n^2-n);
```
(%o3) $\log(n^2-n)$

```
(%i4) %,factor;
```
(%o4) $\log((n-1)\,n)$

```
(%i5) %, logexpand;
```
(%o5) $\log((n-1)\,n)$

```
(%i6) %, logexpand=super;
```
(%o6) $\log(n)+\log(n-1)$

2)

```
(%i7) asinh(x),logarc:true;
```
(%o7) $\log\left(\sqrt{x^2+1}+x\right)$

```
(%i8) integrate(1/x,x),logabs:true;
```
(%o8) $\log(|x|)$

Tabela 4.7

POLECENIA	OPIS
trigexpand(w) w,**trigexpand**	rozwija funkcje trygonometryczne sumy i wielokrotności argumentu, zastępując je funkcjami z pojedynczym argumentem
trigreduce(w) w,**trigreduce**	zastępuje potęgi i iloczyny funkcji trygonometrycznych liniowymi kombinacjami funkcji trygonometrycznych

POLECENIA	OPIS
	wielokrotności argumentu
trigsimp(w)	upraszcza, wykorzystując „jedynkę trygonometryczną”
trigrat(w)	upraszcza wymierne wyrażenia trygonometryczne
load(atrig1)	daje więcej możliwości w zakresie przekształceń funkcji cyklometrycznych
load(ntrig)	dodaje możliwości w zakresie przekształceń funkcji trygonometrycznych o argumencie postaci $(n\ \%pi/k)$, dla $k=5$ oraz $k=10$, gdy n jest liczbą całkowitą
halfangles:true	pozwala na przekształcanie funkcji trygonometrycznych z kątami połówkowymi

1)

```
(%i1) (sin(alpha)+cos(alpha))^2;
      %,expand;
```

(%o1) $(\sin(\alpha)+\cos(\alpha))^2$

(%o2) $\sin(\alpha)^2+2\cos(\alpha)\sin(\alpha)+\cos(\alpha)^2$

```
(%i3) trigrat(%);
```

(%o3) $\sin(2\,\alpha)+1$

2)

```
(%i4) sin(2*beta)/(cos(2*beta)+1);
      %,trigexpand;
      trigsimp(%);
      %,trigreduce;
```

(%o4) $\dfrac{\sin(2\,\beta)}{\cos(2\,\beta)+1}$

(%o5) $\dfrac{2\cos(\beta)\sin(\beta)}{-\sin(\beta)^2+\cos(\beta)^2+1}$

(%o6) $\dfrac{\sin(\beta)}{\cos(\beta)}$

(%o7) $\tan(\beta)$

3)

```
(%i8) l:[sin(x)/tan(x),sec(x)/sin(x),tan(x)/cos(x),csc(x)];
      map(trigrat,l);
```

(%o8) $\left[\dfrac{\sin(x)}{\tan(x)},\dfrac{\sec(x)}{\sin(x)},\dfrac{\tan(x)}{\cos(x)},\csc(x)\right]$

(%o9) $\left[\cos(x),\dfrac{2}{\sin(2\,x)},\dfrac{2\sin(x)}{\cos(2\,x)+1},\dfrac{1}{\sin(x)}\right]$

4)

```
(%i10) tan(3*alpha);
       %,trigexpand;
       trigrat(%);
```

(%o10) $\tan(3\,\alpha)$

(%o11) $\dfrac{3\tan(\alpha)-\tan(\alpha)^3}{1-3\tan(\alpha)^2}$

(%o12) $\dfrac{\sin(3\,\alpha)}{\cos(3\,\alpha)}$

```
(%i13) tan(3*alpha);
       trigrat(%);
       %,trigexpand;
```

(%o13) $\tan(3\,\alpha)$

(%o14) $\dfrac{\sin(3\,\alpha)}{\cos(3\,\alpha)}$

(%o15) $\dfrac{3\cos(\alpha)^2\sin(\alpha)-\sin(\alpha)^3}{\cos(\alpha)^3-3\cos(\alpha)\sin(\alpha)^2}$

5)

```
(%i16) sin(x)^3;
       %,trigreduce;
```

(%o16) $\sin(x)^3$

(%o17) $\dfrac{3\sin(x)-\sin(3\,x)}{4}$

6)

```
(%i18) sin(x+%pi/3);%,trigexpand;
```

(%o18) $\sin\left(x+\dfrac{\pi}{3}\right)$

(%o19) $\dfrac{\sin(x)}{2}+\dfrac{\sqrt{3}\cos(x)}{2}$

7)

```
(%i20) load(ntrig)$
       cos(%pi/10);
```

(%o21) $\dfrac{\sqrt{\sqrt{5}+5}}{2^{3/2}}$

8)

```
(%i22) halfangles:true$
       cot(theta/2),trigexpand;
```

(%o23) $\dfrac{\cos(\theta)+1}{\sin(\theta)}$

Uwaga 2

Równania w Maximie można przekształcać stronami. Poniżej zamieszczamy prosty przykład.

```
(%i1) r:2*x-5=5*x+3;
```

(%o1) $2\,x-5=5\,x+3$

```
(%i2) r:r+5;
```

(%o2) $2\,x=5\,x+8$

```
(%i3) r:r-5*x;
```

(%o3) $-3\,x=8$

```
(%i4) r:r/(-3);
```

(%o4) $x=-\dfrac{8}{3}$

5. Liczby zespolone

Jeżeli $z = x + iy$, gdzie $x, y \in \mathbb{R}$, to:

- mówimy, że liczba z jest zapisana w postaci kartezjańskiej (algebraicznej),
- moduł liczby z jest określony wzorem $|z| = \sqrt{x^2 + y^2}$,
- argumentem liczby $z \neq 0$ nazywamy każdą liczbę $\varphi \in \mathbb{R}$ spełniającą układ $\cos\varphi = \frac{x}{|z|}$ i $\sin\varphi = \frac{y}{|z|}$,
- argumentem głównym liczby $z \neq 0$ nazywamy argument $\varphi \in (-\pi, \pi]$,
- $|z|(\cos\varphi + i\sin\varphi)$ nazywamy postacią trygonometryczną liczby z,
- $z^n = |z|^n(\cos(n\varphi) + i\sin(n\varphi))$, $n \epsilon \mathbb{N}$, nazywamy wzorem de Moivre'a,
- $|z|e^{i\varphi}$ nazywamy postacią wykładniczą liczby z,
- $\exp(z) = e^z = e^x(\cos y + i\sin y)$ nazywamy funkcją wykładniczą liczby z,
- $\sin z = \frac{1}{2i}\left(e^{iz} - e^{-iz}\right)$ nazywamy funkcją sinus liczby z,
- $\cos z = \frac{1}{2}\left(e^{iz} + e^{-iz}\right)$ nazywamy funkcją cosinus liczby z.

Tabela 5.1

POLECENIA	OPIS
%i	oznaczenie jednostki urojonej
z:x+%i*y	przypisanie
realpart(z**)**	część rzeczywista liczby z
imagpart(z**)**	część urojona liczby z
conjugate(z**)**	sprzężenie liczby z
cabs(z**)**	moduł liczby z
carg(z**)**	argument główny liczby z
rectform(z**)**	postać kartezjańska liczby z
polarform(z**)**	postać wykładnicza liczby z
demoivre(exp(z)**)**	przekształca liczbę $e^z = exp(x + iy)$ do postaci $e^x(\cos y + i\ \sin y)$, gdzie x, $y \in \mathbb{R}$

ZADANIA PRZYKŁADOWE

Przykład 1

Sprawdzimy działanie powyższych poleceń dla liczby zespolonej $z = \sqrt{3} - i$.

Najpierw przypiszemy wartość $\sqrt{3} - i$ do zmiennej z, a następnie (korzystając z **Menu-Uprość** oraz tabeli 5.1) zastosujemy odpowiednie polecenia:

(%i1) z:sqrt(3)-%i;
(%o1) $\sqrt{3}-\%i$

(%i2) realpart(z);
(%o2) $\sqrt{3}$

Uproszczenia zespolone/Część rzeczywista

(%i3) imagpart(z);
(%o3) -1

Uproszczenia zespolone/Część urojona

(%i4) conjugate(z);
(%o4) $\%i+\sqrt{3}$

Wyznaczamy liczbę sprzężoną do z.

(%i5) cabs(z);
(%o5) 2

Obliczamy moduł z.

(%i6) carg(z);
(%o6) $-\frac{\pi}{6}$

Wyznaczamy argument główny liczby z.

(%i7) z1:exp(z);
(%o7) $\%e^{\sqrt{3}-\%i}$

Przypisujemy wartość do zmiennej $z1$.

(%i8) demoivre(z1);
(%o8) $\%e^{\sqrt{3}}\,(\cos(1)-\%i\,\sin(1))$

Zapisujemy $z1$ w postaci trygonometrycznej. **Uproszczenia zespolone/Demoivre**

Przykład 2

Wykonamy poniższe działania w zbiorze liczb zespolonych:

a) $(1+3i)+(2-4i)$, c) $(1-i)^9$, e) $\sqrt{3-4i}$,

b) $(1+3i)(2-4i)$, d) $\frac{2-i}{3+i}$, f) $(-1)^{\frac{1}{4}}$.

a)

(%i1) (1+3*%i)+(2-4*%i);
(%o1) $3-\%i$

b) W tym przykładzie polecenia **rectform** użyjemy na dwa sposoby:

(%i2) (1+3*%i)*(2-4*%i);
(%o2) $(2-4\,\%i)(3\,\%i+1)$

(%i3) rectform((1+3*%i)*(2-4*%i));
(%o3) $2\,\%i+14$

(%i4) (1+3*%i)*(2-4*%i),rectform;
(%o4) $2\,\%i+14$

c)

```
(%i5) (1-%i)^9,rectform,factor;
(%o5) -16(%i-1)
```

d)

```
(%i6) (2-%i )/(3+%i),rectform;
```

(%o6) $\frac{1}{2}-\frac{\%i}{2}$

e) Istnieją dokładnie dwa pierwiastki drugiego stopnia z liczby $3-4i$. Maxima, po zastosowaniu polecenia **rectform**, przedstawia tylko jeden z nich. Aby otrzymać oba pierwiastki, rozwiążemy równanie $z^2=3-4i$ za pomocą polecenia **solve** (patrz tabela 7.1).
Co można powiedzieć o wzajemnej relacji tych dwóch rozwiązań?

```
(%i7) sqrt(3-4*%i),rectform;
(%o7) 2-%i
```

```
(%i8) solve(z^2=3-4*%i),rectform;
(%o8) [z=%i-2, z=2-%i]
```

f) Podobnie jak w poprzednim przykładzie, znajdziemy pierwiastki, rozwiązując odpowiednie równanie, tym razem, czwartego stopnia.

```
(%i9) solve(z^4+1),rectform;
```

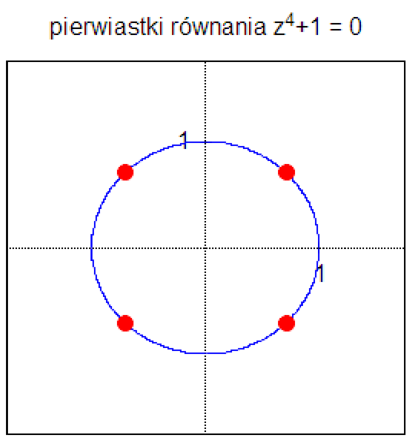
(%o9) $\left[z=\frac{\%i}{\sqrt{2}}-\frac{1}{\sqrt{2}}, z=-\frac{\%i}{\sqrt{2}}-\frac{1}{\sqrt{2}}, z=\frac{1}{\sqrt{2}}-\frac{\%i}{\sqrt{2}}, z=\frac{\%i}{\sqrt{2}}+\frac{1}{\sqrt{2}}\right]$

Przedstawimy teraz graficzną interpretację rozwiązania równania.

```
(%i1) r:z^4+1=0$
      roz:solve(r),rectform$
      p:map(rhs,roz)$
      px:map(realpart,p)$
      py:map(imagpart,p)$
```

Generujemy współrzędne punktów, które zostaną narysowane poniżej (przy użyciu pakietu **draw**). Ustawienia opcji grafiki 2D są opisane w tabeli 6.3.

```
load(draw)$
draw2d(dimensions=[300,240],
       proportional_axes=xy,
       font = "Arial",  font_size=10,
       title=sconcat("pierwiastki równania ",r),
       implicit(x^2+y^2=1,x,-1,1,y,-1,1 ),
       xrange=[-1.75,1.75],
       yrange=[-1.75,1.75],
       xaxis=true,yaxis=true,
       xtics_axis=true,xtics={1},
       ytics_axis=true,ytics={1},
       color=red,point_size=1,
       point_type=filled_circle,
       points(px,py))$
```



ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1
Wyznaczyć części rzeczywiste, części urojone oraz obliczyć moduły podanych liczb zespolonych:

$z_1 = 4 - i$, $z_2 = (1-3i)(2+7i)$, $z_3 = \frac{1+2i}{1-2i}$, $z_4 = \cos\left(\frac{5\pi}{12}\right) + i\sin\left(\frac{7\pi}{12}\right)$.

Zadanie 2
Wyznaczyć argumenty następujących liczb zespolonych:

a) $-4+4i$, b) $1-\sqrt{3}i$, c) $-7\sqrt{3}-7i$, d) $3\sqrt{2}+3\sqrt{2}i$.

Zadanie 3
Podane liczby zespolone zapisać w postaci trygonometrycznej:

a) $-1+\sqrt{3}\,i$, b) $2+2i$, c) $5\sqrt{2}-5\sqrt{2}i$, d) $-3\sqrt{3}-3i$.

Zadanie 4
Obliczyć: a) $\sqrt{24+10i}$, b) $\sqrt{-36i}$, c) $\sqrt[3]{8i}$, d) $\sqrt[4]{16}$.

Zadanie 5
Uprościć wyrażenia: a) $(1-i)^{12}$, b) $\left(-\sqrt{3}+i\right)^7$.

Zadanie 6
Wykonać działania i wynik przedstawić w postaci kartezjańskiej (algebraicznej):

a) $\frac{(-1+i)^{12}}{\left(-1+\sqrt{3}i\right)^6}$, b) $\frac{(1-3i)^{15}}{(2-i)^{23}}$.

Zadanie 7
Wykonać działania i wynik przedstawić w postaci trygonometrycznej:

a) $\frac{4+4i}{1-i}$, b) $\frac{\left(1-\sqrt{3}i\right)^6}{(1+i)^{12}}$, c) $\frac{\left(1+\sqrt{3}i\right)^4}{\sqrt{2}-\sqrt{2}i}$, d) $\left(\frac{1-i}{1+i}\right)^{12}$.

Zadanie 8
Obliczyć:

a) $e^{-3\pi i}$, b) $e^{1+\pi i}$, c) $\sin 7i$, d) $\cos(2\pi + i\pi)$, e) $\sin\left(\frac{3\pi}{2}+i\right)$, f) $\sin\left(\frac{1}{2}+6i\right)$.

Odpowiedzi

1. $\mathrm{re}(z_1) = 4$, $\mathrm{im}(z_1) = -1$, $|z_1| = \sqrt{17}$; $\mathrm{re}(z_2) = 23$, $\mathrm{im}(z_2) = 1$, $|z_2| = \sqrt{530}$; $\mathrm{re}(z_3) = \frac{-3}{5}$, $\mathrm{im}(z_3) = \frac{4}{5}$, $|z_3| = 1$; $\mathrm{re}(z_4) = \cos\frac{5\pi}{12}$, $\mathrm{im}(z_4) = \sin\frac{5\pi}{12}$, $|z_4| = 1$.

2. a) $\frac{3\pi}{4}$, b) $\frac{-\pi}{3}$, c) $\frac{-5\pi}{6}$, d) $\frac{\pi}{4}$.

3. a) $2\left(\cos\frac{2\pi}{3}+i\sin\frac{2\pi}{3}\right)$, b) $2\sqrt{2}\left(\cos\frac{\pi}{4}+i\sin\frac{\pi}{4}\right)$, c) $10\left(\cos\left(-\frac{\pi}{4}\right)+i\sin\left(-\frac{\pi}{4}\right)\right)$,
d) $6\left(\cos\left(-\frac{5\pi}{6}\right)+i\sin\left(-\frac{5\pi}{6}\right)\right)$.

4. a) $-5-i,\ 5+i$, b) $-3\sqrt{2}+3\sqrt{2}i,\ 3\sqrt{2}-3\sqrt{2}i$, c) $-2i,\ \sqrt{3}+i,\ -\sqrt{3}+i$,
d) $-2,\ 2,\ -2i,\ 2i$.

5. a) -64, b) $64\sqrt{3}-64i$.

6. a) -1, b) $-\frac{110464i}{390625}-\frac{24448}{390625}$.

7. a) $4\left(\cos\frac{\pi}{2}+i\sin\frac{\pi}{2}\right)$, b) $\cos\pi+i\sin\pi$, c) $8\left(\cos\left(\frac{-5\pi}{12}\right)+i\sin\left(\frac{-5\pi}{12}\right)\right)$,
d) $\cos 0+i\sin 0$.

8. a) -1, b) $-e$, c) $i\ \sinh(7)$, d) $\cosh(\pi)$, e) $-\cosh(1)$,
f) $i\ \sin\left(\frac{1}{2}\right)\cosh(6)+\cos\left(\frac{1}{2}\right)\sinh(6)$.

6. Wykresy

6.1. Podstawowe polecenia graficzne

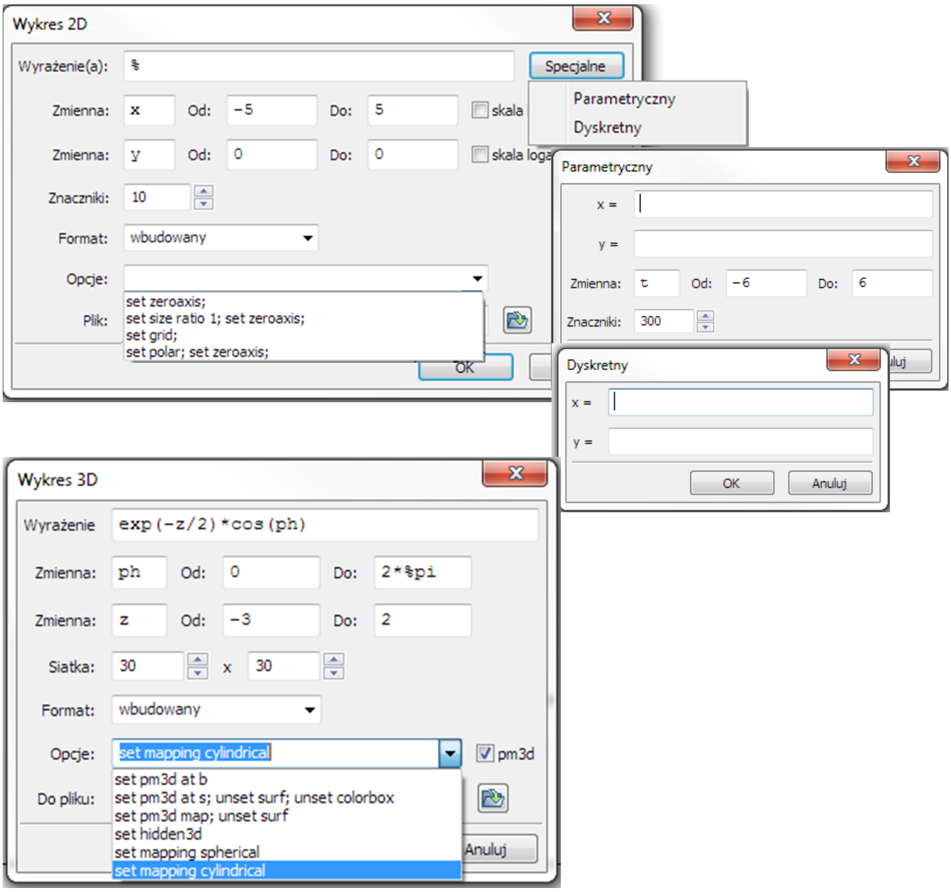
Maxima, podobnie jak większość programów typu CAS, udostępnia szeroki zakres komend dających możliwość tworzenia wykresów funkcji jednej i dwóch zmiennych, wykresów funkcji uwikłanych, krzywych zadanych równaniami parametrycznymi oraz krzywych zadanych biegunowo. Pozwala również generować wykresy statystyczne oraz proste animacje. Pakiet ten nie dysponuje własnymi mechanizmami wizualizacji danych, lecz korzysta z zewnętrznego programu Gnuplot (dystrybuowanego razem z Maximą). Wynik jest prezentowany w osobnym oknie (oknie Gnuplota), dzięki czemu jest możliwa interakcja użytkownika i zmiana pewnych parametrów wizualizacji. Trzeba jednak pamiętać, że interfejs Maximy jest blokowany tak długo, jak długo aplikacja Gnuplot jest aktywna i nie można w tym czasie wykonywać obliczeń.

Istnieje też drugi tryb rysowania wykresów. Jeśli przed nazwą komendy graficznej (patrz tabela 6.1 oraz tabela 6.5) dodamy prefiks *wx*, wówczas wykres funkcji zostanie wyrenderowany przez program Gnuplot, a następnie wynik wyświetlony bezpośrednio w oknie notatnika wxMaximy. Wygenerowane wykresy stają się częścią dokumentu wxMaxima i są dostępne do końca sesji programu lub, w przypadku zapisu do formatu wxmx, również przy ponownym otworzeniu dokumentu.

W następnym podrozdziale omówimy polecenia pakietu *draw*, który pozwala na zaawansowane ustawienia dla wizualizacji danych i wykresów funkcji w postaci grafiki 2D i 3D. Pakiet ten zawiera interfejs udostępniający użytkownikowi Maximy korzystanie z możliwości Gnuplota w pełnym zakresie. Na przykład pozwala eksportować wyniki do zewnętrznych plików w formatach zarówno rastrowych, jak i wektorowych, co może okazać się przydatne w trakcie przygotowywania analiz danych potrzebnych do innych dokumentów (raportów, publikacji). Przykłady użycia pakietu *draw* oraz więcej materiałów na ten temat można znaleźć w [7], na stronie M. R. Riotorto (autora tego pakietu).

Polecenia i opcje służące do wizualizacji są częścią Maximy, która stale się zmienia przy wprowadzaniu nowych wersji. Na wymienione zmiany mają również wpływ nowe odsłony Gnuplota. Niniejszy rozdział powstał w oparciu o Maximę dla Windows w wersji 5.31.2 oraz 5.33.0.

Poniżej zostały zamieszczone okna formularzy do rysowania podstawowych wykresów 2D i 3D dostępne z palety **Podstawowa Matematyka** albo **Menu-Kreślenie**. Dla okna Wykres 3D podajemy przykładowe uzupełnienie wraz z ilustracją. Czytelników zainteresowanych uzyskaniem bardziej zaawansowanych wykresów odsyłamy do zamieszczonych w tym rozdziale tabel oraz [1].




```
(%i1) wxplot3d(exp(-z/2)*cos(ph), [ph,0,2*%pi], [z,-3,2],
      [gnuplot_preamble, "set mapping cylindrical"])$
```

(%t1)

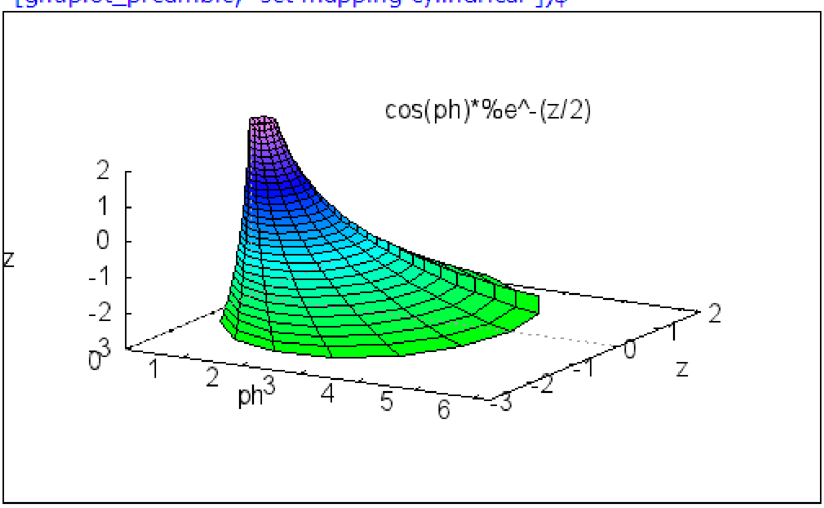


Tabela 6.1

PODSTAWOWE FUNKCJE INTRFEJSU GNUPLOT
plot2d(f(x), [x, x_{min}, x_{max}], *opcje***)** - rysuje wykres funkcji jednej zmiennej f na przedziale $[x_{min}, x_{max}]$
plot2d([f_1(x), f_2(x), …f_n(x)], [x, x_{min}, x_{max}], *opcje***)** - rysuje wykresy funkcji jednej zmiennej $f_1, f_2, \dots, f_n$ na przedziale $[x_{min}, x_{max}]$
plot3d(f(x,y), [x, x_{min}, x_{max}], [y, y_{min}, y_{max}], *opcje***)** - rysuje wykres funkcji dwóch zmiennych f na prostokącie $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$
plot3d([f_1(x,y), f_2(x,y), …, f_n(x,y), [x, x_{min}, x_{max}], [y, y_{min}, y_{max}]], *opcje***)** - rysuje wykresy funkcji dwóch zmiennych $f_1, f_2, \dots, f_n$ na prostokącie $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$
plot2d([**parametric**, x(t), y(t), [t, t_{min}, t_{max}]], *opcje***)** - rysuje krzywą zadaną równaniami parametrycznymi: $x = x(t),\ y = y(t),\ t \in [t_{min}, t_{max}]$
plot3d([x(u,v), y(u,v), z(u,v)], [u, u_{min}, u_{max}], [v, v_{min}, v_{max}], *opcje***)** - rysuje powierzchnię zadaną równaniami parametrycznymi: $x = x(u,v), y = y(u,v), z = z(u,v), u \in [u_{min}, u_{max}], v \in [v_{min}, v_{max}]$
plot2d([**discrete**, [x_1,x_2, …], [y_1,y_2, …]], *opcje***)** **plot2d(**[**discrete**, [[x_1,y_1], [x_2,y_2], …]], *opcje***)** - rysuje punkty dane w formie dwóch list albo w formie listy punktów
plot2d(f(phi), [phi, phi_1, phi_2], [gnuplot_preamble, "set polar; "], *opcje***)** - rysuje krzywą $r = f(phi), phi \in [phi_1, phi_2]$ zadaną w postaci biegunowej
plot3d(f(r,phi), [r, r_1, r_2], [phi, phi_1, phi_2], [transform_xy, polar_to_xy], *opcje***)** - rysuje powierzchnię $z = f(r, phi)$ określoną za pomocą współrzędnych biegunowych (w kartezjańskim układzie współrzędnych)
plot3d(f(th, phi), [th, th_1, th_2], [phi, phi_1, phi_2], [transform_xy, spherical_to_xyz], *opcje***)** - rysuje powierzchnię $r = f(th, phi)$ określoną za pomocą współrzędnych sferycznych (w kartezjańskim układzie współrzędnych)
contour_plot(f(x,y), [x, x_{min}, x_{max}], [y, y_{min}, y_{max}], *opcje***)** - rysuje poziomice funkcji f
with_slider(a, [wartosci_dla_a],obiekty graficzne, opcje**)** - tworzy interaktywne okno graficzne z możliwością ustawiania suwakiem wartości *a* spośród elementów listy *wartości_dla_a*

Tabela 6.2 zawiera ustawienia dla podstawowych funkcji (poleceń) graficznych wymienionych w tabeli 6.1. Aby sprawdzić aktualne ustawienia, wystarczy wywołać **set_plot_option()**. Polecenie to służy również do wprowadzania własnych ustawień.

Tabela 6.2. Style stosowane w grafice 2D oraz 3D

OPCJA	USTAWIENIA DOMYŚLNE	INNE / OPIS
[axes, …**]**	true	rysuje osie układu, **false** je ukrywa, ***x*** **-** tylko oś Ox, ***y*** - tylko oś Oy, **solid -** obie osie linią ciągłą
[azimuth, …**]**	30	azymut, miara kąta wyrażona w stopniach
[box, …**]**	true	rysuje box ograniczający w grafice 2D oraz 3D, **false** go ukrywa
[color, …**]**	blue [1)]	w grafice 2D: **k1**, **k2**, ... oznaczają kolory dla kolejnych krzywych, w grafice 3D: **k1**,**k2** oznaczają kolory dla linii siatki (wierzch i spód)
[color_bar, …**]**	false	**true**, wtedy wyświetla się skala kolorów w grafice 3D określona przez ***palette***
[elevation, …**]**	60	kąt elewacji kamery, miara wyrażona w stopniach
[gnuplot_ preamble, …**]**		"set … ; set …; …" polecenia Gnuplota
[grid, …**]**	30,30	w grafice 3D liczby punktów kratowych w kierunku osi Ox oraz Oy
grid2d		rysuje siatkę zgodnie z ustawieniami ***xtics*** oraz ***ytics***
[label, …**]**		**[„tekst", x0, y0]** umieszcza etykietę „tekst" w punkcie o współrzędnych (x_0, y_0)
[legend, …**]**	nazwy [2)]	**false** (nie wyświetlają się etykiety obiektów graficznych), inne np.: sconcat("f(x)=",f), „tekst"
[mesh_lines_ color, …**]**	black	kolor linii siatki (gdy działa ***palette***), **false** (linie siatki nie są widoczne)
[palette, …**]**	[hue,0.25, 0.7,0.8,0.5]	**false** (powierzchnia nie jest cieniowana), dla każdego obiektu można ustawić inną paletę, istnieje wiele sposobów określania palet opisanych w [1]
[point_type,..**]**	bullet	typ znacznika do rysowania punktów [3)]
[same_xy, …**]**	false	**true** (proporcje jednostek na osiach są 1:1)
[same_xyz, …**]**	false	**true** (proporcje jednostek na osiach są 1:1:1)
[style, …**]**	lines	style dla obiektów 2D: **points**, **linespoints**, **dots** [4)]
[title, …**]**	" "	tytuł nad rysunkiem
[transform _xy, …**]**	brak	zamiana współrzędnych (patrz tabela 6.1): **polar_to_xy**, **spherical_to_xyz**
[5)] **[xtics,** a, k, b**]**	1	rysuje znaczniki na osi Ox: od ***a*** do ***b*** z krokiem ***k*** lub **[xtics,** a] z krokiem **a** (początek auto)
[5)] **[xlabel**, …**]**	x	zmienia nazwę etykiety osi Ox
[5)] **[x**,x1,x2**]**		zakres na osi Ox
[yx_ratio, …**]**	auto	w grafice 2D współczynnik proporcji ramki

[1)] Kolory oraz ich liczbowe odpowiedniki: cyan(0), blue(1), red(2), green(3), magenta(4), black(5), cyan(6), ... i dalej kolory powtarzają się cyklicznie.

2) Nazwy użytych obiektów graficznych.
3) Dostępne nazwy i ich liczbowe odpowiedniki: bullet(1), circle(2), plus(3), times(4), asterisk(5), box(6), square(7), triangle(8), delta(9), wedge(10), nabla(11), diamond(12), lozenge(13).
4) np.: **[lines**,a**]** lub **[lines**,a,b**]**, a-szerokość linii, b-kolor, **[points**,a,b**]** lub **[points**,a,b,c**]**, a-wielkość punktu, b-kolor, c-typ znacznika
5) Opcje te mają odpowiedniki dla y oraz z.

ZADANIA PRZYKŁADOWE

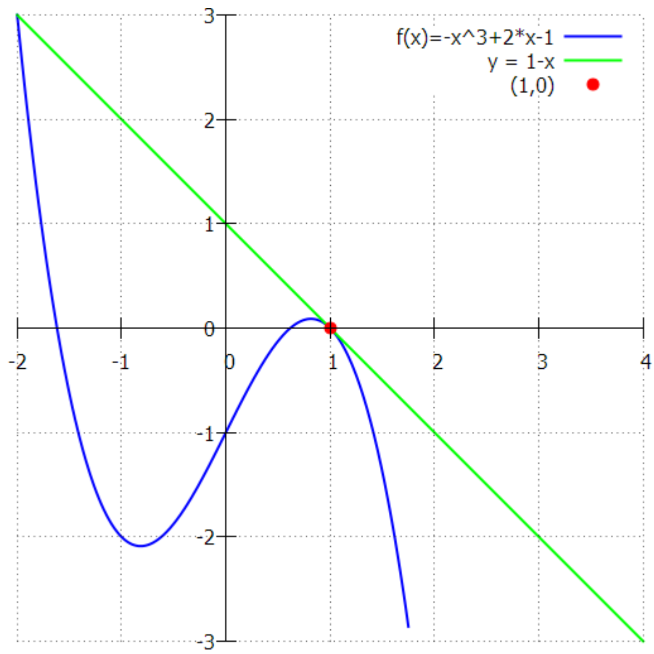
Przykład 1

Naszkicujemy wykres funkcji danej wzorem $f(x) = -x^3 + 2x - 1$, styczną do tego wykresu poprowadzoną w punkcie o odciętej $x_0 = 1$ oraz zaznaczymy punkt styczności.

Najpierw wykonamy odpowiednie obliczenia, a następnie, korzystając z polecenia **plot2d** i jego ustawień, umieścimy trzy obiekty graficzne na jednym rysunku.

```
(%i1) f:-x^3+2*x-1$ x0:1$ y0:f,x=x0$ k:at(diff(f,x),x=x0)$
      s:y=y0+k*(x-x0),expand;
(%o5) y=1-x
```

```
(%i6) plot2d([f,rhs(s),[discrete,[[x0,y0]]]], [x,x0-3,x0+3], [y,-3,3], grid2d,
             [box, false], [axes, true], [axes, solid], same_xy,
             [legend, sconcat("f(x)=",f), string(s), sconcat("(",x0,",",y0,")")],
             [style, [lines,2,1], [lines,2,3], [points,3,2,1]] )$
plot2d: some values were clipped.
```



Przykład 2

Wyznaczymy funkcję kwadratową P, której wykres przechodzi przez punkty: $(-1,1)$, $(0,1)$, $(1,-1)$. Wykres funkcji oraz punkty przedstawimy na jednym rysunku.

Najpierw wprowadzimy odpowiednie deklaracje, a następnie, korzystając z polecenia **algsys**, rozwiążemy układ równań. W ten sposób otrzymamy współczynniki szukanego trójmianu kwadratowego P.

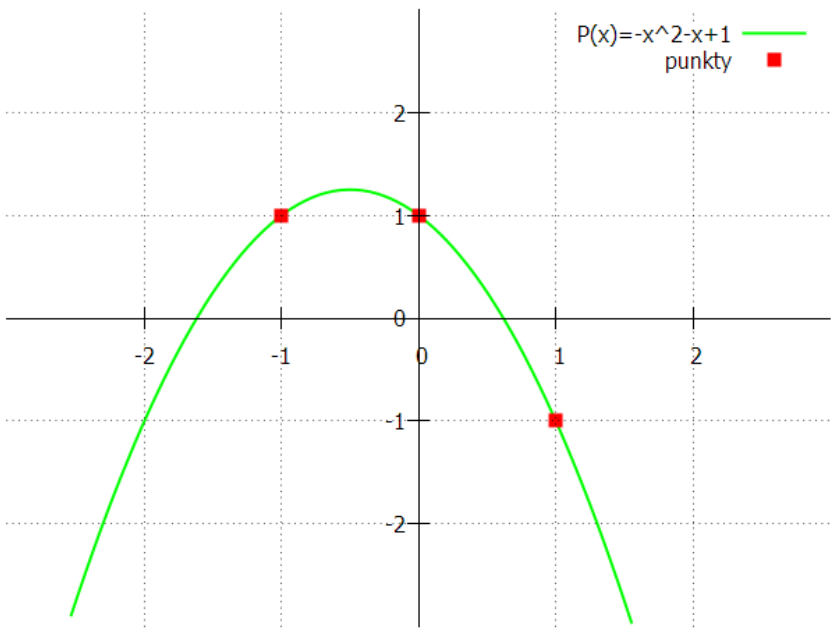
```
(%i1) P(x) := a*x^2+b*x+c$  [x1,y1]:[-1,1]$ [x2,y2]:[0,1]$ [x3,y3]:[1,-1]$
```

```
(%i5) wsp: algsys([P(x1)=y1, P(x2)=y2, P(x3)=y3], [a,b,c]); P:P(x),wsp;
```

(%o5) $[[a=-1, b=-1, c=1]]$

(%o6) $-x^2-x+1$

```
(%i7) plot2d([ P, [discrete, [x1,x2,x3],[y1,y2,y3]]], [x,-3,3], [y,-3,3], grid2d,
             [xtics,-2,1,2], [ytics,-2,1,2], [legend, sconcat("P(x)=",P), "punkty"],
             [style, [lines,1,3], [points,3,2,2]], [box,false], [axes,solid] )$
plot2d: some values were clipped.
```



Uwaga 1

Komunikaty *some values were clipped* pojawiające się w obu powyższych przykładach informują, że wykresy funkcji na podanych przedziałach nie są prezentowane w całości ze względu na „wymuszenie” zakresu na osi Oy. Oczywiście dla wielomianów zakres automatyczny może być bardzo duży i wtedy słabo widoczne będą inne istotne elementy wykresu.

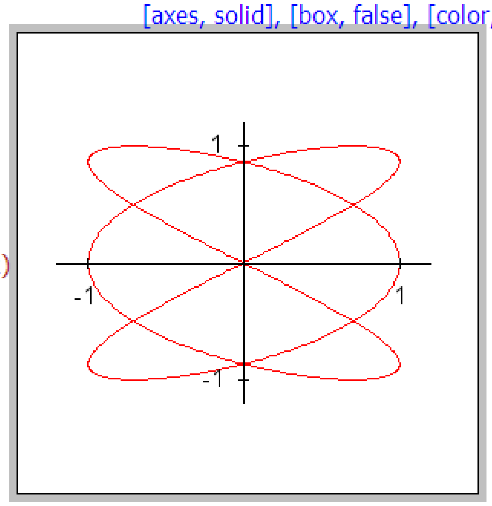
Przykład 3

Korzystając z polecenia **with_slider**, naszkicujemy krzywe opisane równaniami parametrycznymi dla wybranych wartości a (zmienianych za pomocą suwaka)

$$x(t) = \cos(at),\ y(t) = \cos\left(2t - \frac{\pi}{2}\right),\ t \in [0,2\pi].$$

```
(%i1) wxplot_size:[300,300];
(%o1) [300,300]

(%i2) with_slider(a, [1,2,3,4,5,6],
                  [parametric, cos(a*t), cos(2*t-%pi/2)],
                  [t,0,2*%pi],[x,-1.2,1.2],[y,-1.2,1.2],
                  [xtics,-1,2,1],[ytics,-1,2,1], [nticks,100],
                  [axes, solid], [box, false], [color,red] )$
```

(%t2)



Rozważane w tym przykładzie krzywe są krzywymi Lissajous.
Na pasku narzędziowym dostępne są przyciski do nawigacji okna (%t2).
Można dzięki nim uruchomić animację rodziny krzywych.

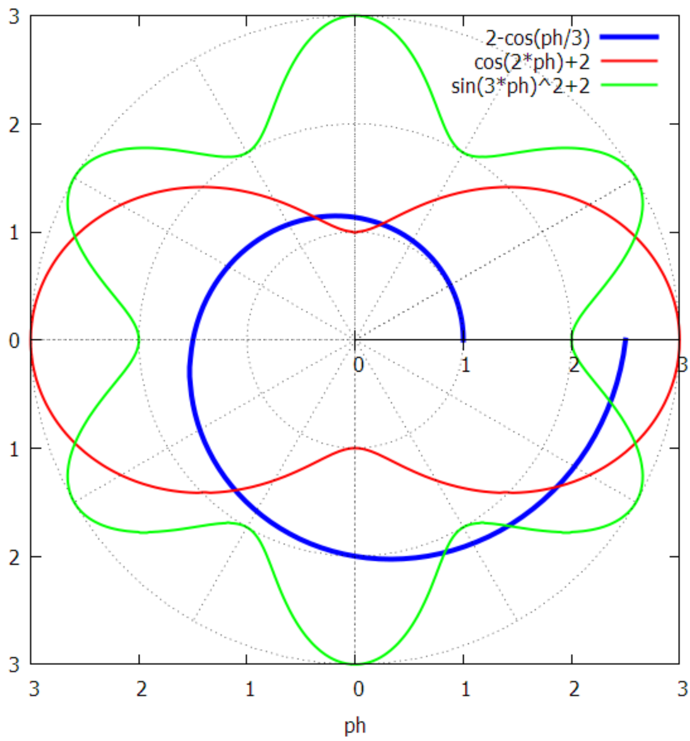
Przykład 4

Zaprezentujemy wykresy trzech krzywych danych równaniami biegunowymi:

$$r = 2 - \cos\left(\frac{\varphi}{3}\right),\ r = 2 + \cos(2\varphi),\ r = 2 + \sin^2(3\varphi),\ \varphi \in [0,2\pi].$$

By narysować podane krzywe w układzie biegunowym, zastosujemy bezpośrednie polecenia pakietu Gnuplot wywoływane przez [**gnuplot_preamble**, …**]** oraz ustawienia dla **plot2d** (patrz tabela 6.2).

```
(%i1) plot2d([2-cos(ph/3), 2+cos(2*ph), 2+sin(3*ph)^2], [ph,0,2*%pi], [y,-3,3],
             [style, [lines,4], [lines,2], [lines,2]], same_xy, grid2d,
             [gnuplot_preamble, "set polar; set grid polar; "] )$
```

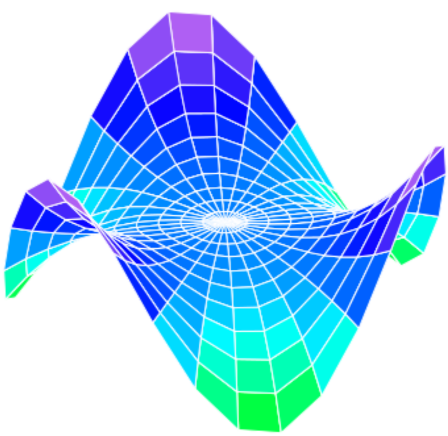


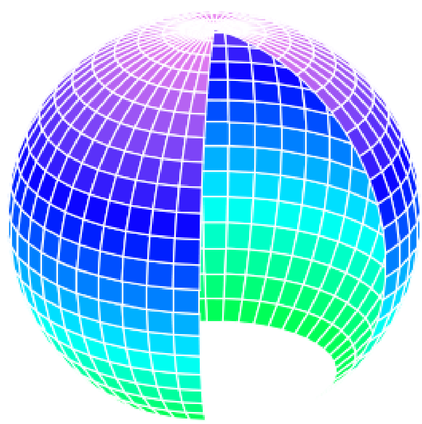
Przykład 5

Poniżej dwie przykładowe powierzchnie. Pierwsza określona za pomocą współrzędnych biegunowych, natomiast druga za pomocą współrzędnych sferycznych.

```
(%i1) plot3d( r^3*cos(3*th), [r,0,2], [th,0,5*%pi], [box,false], [grid,10,80],
      [mesh_lines_color,white], [transform_xy,polar_to_xy], [legend,false]  )$
```

```
(%i2) plot3d( 1, [theta,0,4/5*%pi], [phi,0,5/3*%pi],  [box,false], [grid, 20,40],
                  [transform_xy, spherical_to_xyz], [mesh_lines_color,white],
                  same_xyz, [legend, false]  )$
```





Przykład 6

Dane są funkcje:

$f_1(x,y) = \cos^2 x - \sin^2 x$ na zbiorze $D_1 = [-2,2] \times [-2,2]$;

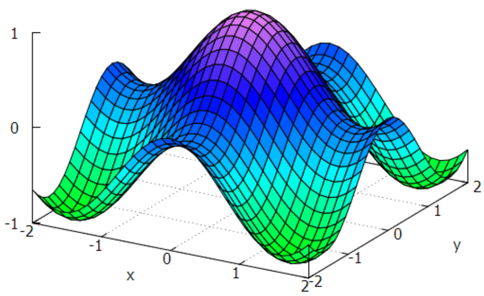
$f_2(x,y) = x^2 - y^2$ na zbiorze $D_2 = \{(x,y) \in \mathbb{R}^2 \colon x^2 + y^2 \leq 4\}$;

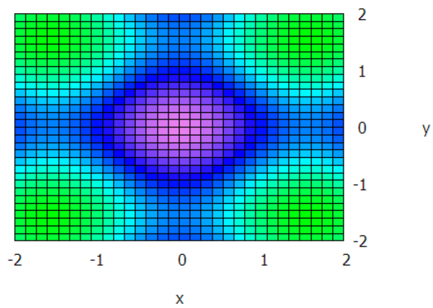
$f_3(x,y) = |y^2 - x^2|$ na zbiorze $D_3 = \{(x,y) \in \mathbb{R}^2 \colon x^2 + y^2 \leq 4 \wedge |x| + |y| \geq 1\}$.

Zaprezentujemy wykresy tych funkcji w dwóch różnych ustawieniach widoku (automatycznym i z góry). Pozwoli to zobaczyć zarówno wykres funkcji jak i zbiór, na którym funkcja jest określona.

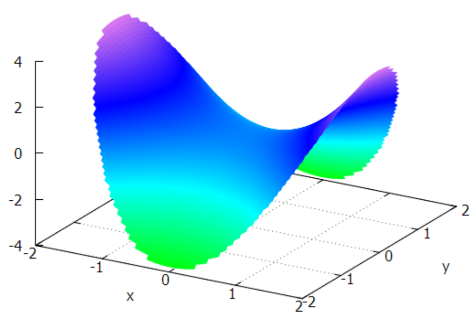
```
(%i1) set_plot_option([legend,false],[xtics,1], [ytics,1], grid2d)$
```

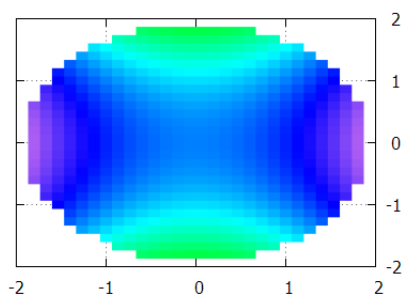
```
(%i2) f1(x,y):=cos(x)^2-sin(y)^2$
      plot3d( f1, [x,-2,2], [y,-2,2], [z,-1,1], [ztics,1] )$
      plot3d( f1, [x,-2,2], [y,-2,2], [ztics,false], [azimuth,0], [elevation,0])$
```



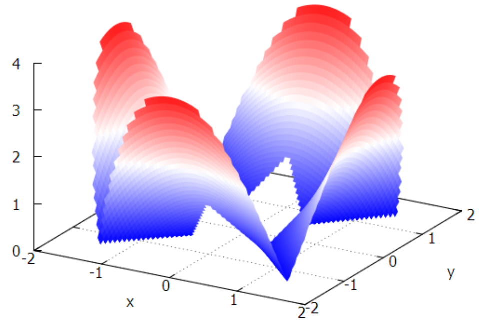


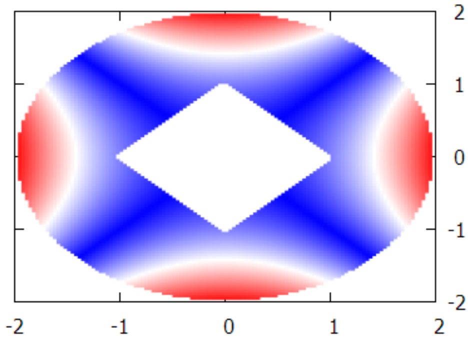

```
(%i5) f2(x,y):= if  x^2+y^2<=4  then  x^2-y^2$
      plot3d( f2, [x,-2,2], [y,-2,2], [z,-4,4], [grid,100,100], [ztics,2],
                 [mesh_lines_color,false]  )$
      plot3d( f2, [x,-2,2], [y,-2,2], [azimuth,0], [elevation,0], [ztics,false],
                 [mesh_lines_color,false]  )$
```






```
(%i8) f3(x,y):= if  x^2+y^2<=4  and  abs(x)+abs(y)>=1  then  abs(y^2-x^2)$
      set_plot_option([palette, [gradient, blue,white,red]],
                      [mesh_lines_color,false], [grid,120,120])$
      plot3d( f3, [x,-2,2], [y,-2,2], [z,0,4], [ztics,1] )$
      plot3d( f3, [x,-2,2], [y,-2,2], [azimuth,0], [elevation,0], [ztics,false] )$
```



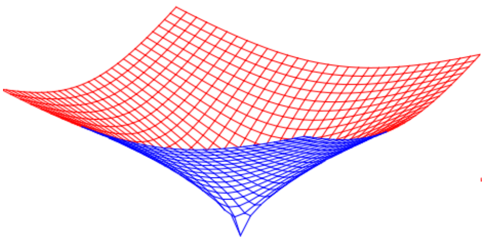


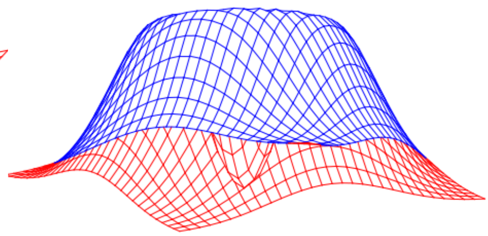
Przykład 7

Przykładowe powierzchnie i ich kolorowanie.

```
(%i1) plot3d( (x^2+y^2)^(1/3), [x,-2,2], [y,-2,2], [z,0,2], [box,false],
              [palette,false], [color,blue,red], [legend,false]  )$
```

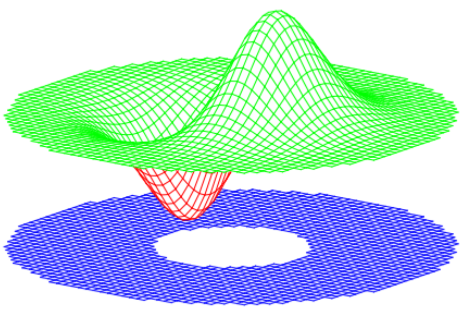
```
(%i2) plot3d( (x^2+y^2)*exp(-x^2-y^2), [x,-2,2], [y,-2,2], [z,0,0.4],
              [box,false], [palette,false], [color,red,blue], [elevation,110]  )$
```

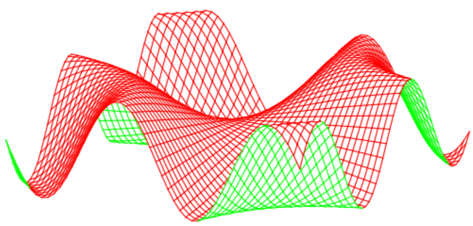




```
(%i3) f(x,y):=if x^2+y^2<9 then x*exp(-x^2-y^2)$
      g(x,y):=if x^2+y^2<9 and x^2+y^2>1then -0.5$
      plot3d([f(x,y),g(x,y),[x,-3,3],[y,-3,3]],[z,-0.6,0.5], [box,false],[legend,false],
              [palette,false], [grid,50,50], [elevation,70], [color,red,green,blue] )$
```

```
(%i6) plot3d(4*sin(x*y/4), [x,-5,5], [y,-5,5], [z,-6,6], [box,false], [grid,50,50],
             [color,green,red], [palette,false]  )$
```





6.2. Pakiet *draw*

Tabela 6.3. Wybrane style stosowane w grafice 2D

OPCJA	USTAWIENIA DOMYŚLNE	INNE / OPIS
allocation =	[[0,0],[1, 1]]	**[[0,0],[1/4, 1/4]]** (wstawienie obiektu w skali **1:4** w dolnym lewym rogu)
background_color=	white	ustala kolor tła [1)]
color =	blue	ustala kolory dla linii, punktów i etykiet [1)]
columns =	1	liczba kolumn przy łączeniu różnych typów wykresów
head_angle =	45	opcja polecenia **vector** (kąt strzałki)
head_length =	2	jw. (długość strzałki)
head_type =	filled	**nofilled**, jw. (wypełnienie strzałki)
delay =	5	opóźnienie (ustawienie dla animowanych gifów)
dimensions =	[600,500]	**[szerokość, wysokość]** (wielkości zależne od opcji **terminal**)
file_name =	"maxima_out"	nazwa lub nazwa z lokalizacją pliku (rozszerzenie dodawane jest automatycznie)
fill_color =	red	kolor wypełnienia [1)]
fill_density =	0	liczba od 0 do 1, intensywność wypełnienia (wielokątów i histogramów)
filled_func =	false	**true** (wypełnia kolorem obszar poniżej wykresu funkcji zadeklarowanej w ***explicit***), **f(x)** (wypełnia kolorem obszar pomiędzy wykresem funkcji *f* a funkcją z **explicit**)
font =	" "	czcionka, np. „Arial", opcja aktywna, gdy działa **terminal**
font_size =	10	rozmiar czcionki, opcja aktywna, gdy ustawiony jest **font**
grid =	false	**true**, wtedy są rysowane linie siatki albo **[nx,ny]** - ilość linii na znacznik
ip_grid =	[50,50]	liczby punktów kratowych dla pierwszego próbkowania (przy rysowaniu wykresów funkcji uwikłanych)
ip_grid_in =	[5,5]	liczby punktów kratowych dla drugiego próbkowania (przy rysowaniu wykresów funkcji uwikłanych)
key =	" "	etykieta dla użytych obiektów graficznych, np. **sconcat**("*y*= ", *f*(*x*))
label([„…",a,b]**)**		etykieta w punkcie o współrzędnych (a,b)
label_alignment =	centered	**left**, **right** ustawienie etykiety (**label**)

OPCJA	USTAWIENIA DOMYŚLNE	INNE / OPIS
label_orientation =	horizontal	**vertical**, orientacja etykiety
line_type =	solid	typ rysowanej linii, **dots** (linia kropkowana)
line_width =	1	grubość rysowanej linii
nticks =	29	liczba wyświetlanych punktów przy rysunkach parametrycznych
point_size =	1	rozmiar punktu (liczba nieujemna) [2)]
point_type =	plus albo 1	typ punktu (nazwa znacznika albo jej liczbowy odpowiednik) [3)]
points_joined =	false	**true** (punkty uzyskane przez **points** są łączone łamaną)
proportional_axes =	none	**xy** (ustawia proporcje jednostek na osiach 1:1)
terminal =	‘screen	export do wybranego formatu np.: **‘png**, **‘wxt**, **‘eps**, **‘eps_color**, **‘pdf**, **‘gif**, **‘jpg**, **‘svg**, **‘animated_gif**
title =	“ ”	tytuł rysunku, np. **sconcat**("a=", a),
user_preamble =	“ “	ustawienia i opcje pakietu Gnuplot np. "set grid polar;"
xaxis =	false	**true**, pokazuje oś Ox [4)]
xaxis_color =	black	kolor osi Ox [1)]
xaxis_type =	dots	typ linii dla osi Ox, **solid** (linia ciągła)
xaxis_width =	1	grubość linii dla osi Ox, liczba nieujemna
xlabel =	“ ”	etykieta osi Ox, w zapisie można stosować symbole, np. "{/Symbol a}" wyświetli α
xrange =	auto	**[x_{min},x_{max}]**, zakres zmiennej x
xtics =	auto	ustala format znaczników, np.: **none** (bez znaczników), **{0,1/2,1}** (znaczniki w punktach: 0,1/2,1), **1/2** (znaczniki w odległości 1/2), **[0,2,10]** (od 0 do 10 z krokiem 2), **{["{/Symbol p}", 3.14], ["e",2.72]}** (zaznacza na osi Ox liczbę π oraz e z opisem)
xtics_axis =	false	**true** (pokazuje znaczniki na osi Ox)

1) Przykładowe nazwy kolorów: *white*, *black*, *gray*, *grey*, *grey10*, *grey90*, *grey80*, *red*, *light_red*, *dark_red*, *green*, *light_green*, *dark_green*, *blue*, *light_blue*, *dark_blue*, *navy*, *yellow*, *magenta*, *pink*, *coral*, *orange*, *beige*, *brown*, *violet*, *plum*, *purple*, *"#1faf75"*.

2) Styl ignorowany w przypadku *point_type=dot* (0).

3) Znaczniki dla punktów: *$none* (-1), *dot* (0), plus (1), *multiply* (2), *asterisk* (3), *square* (4), *filled_square* (5), *circle* (6), *filled_circle* (7), *up_triangle* (8), *filled_up_triangle* (9), *down_triangle* (10), *filled_down_triangle* (11), *diamant* (12) oraz *filled_diamant* (13).

4) Wszystkie opcje rozpoczynające nazwę od $\boldsymbol{x}$ mają swoje odpowiedniki dla $\boldsymbol{y}$ oraz $\boldsymbol{z}$.

Tabela 6.4. Style stosowane w grafice 3D (inne niż w grafice 2D)

OPCJA	USTAWIENIA DOMYŚLNE	INNE / OPIS
axis_3d =	true	rysuje osie układu, **false** je ukrywa
colorbox =	true	rysuje skalę kolorów, **false** ją ukrywa
contour =	none	zaznacza poziomice funkcji f [1] dla ustalonych przez **contour_levels** poziomów: **base** (na **xyplane**), **surface** (na wysokości *h*, tzn. na wykresie *f*), **both** (**base** i **surface** równocześnie), **map** (w układzie *Oxy*)
contour_levels =	5	***n*** (liczba naturalna), **[od, krok, do]** albo **{z1,z2, …, zn}** (lista poziomów)
enhanced3d =	false	**true** lub równoważnie **[z,x,y,z]** albo ogólnie **[f(x,y,z),x,y,z]**
palette =	color gray	**[a,b,c, …]**, gdzie a, b, c, … to kolory albo liczby od -36 do 36, działa, gdy **enhanced3d** = *true*
proportional_axes =	none	**xyz** (ustawia proporcje jednostek na osiach $1:1:1$)
surface_hide =	false	**true** (ukryte części powierzchni nie są widoczne)
view =	[60,30]	miary kątów widzenia obiektu (od 0° do $360°$), pierwszy to obrót pionowy względem osi Ox, drugi poziomy względem Oz
wired_surface =	false	**true** (rysuje siatkę na powierzchni)
x_voxel =	10	zwiększa dokładność rysunków **implicit** w 3D oraz **region** w 2D w kierunku x
y_voxel =	10	jw. tylko w kierunku y
z_voxel =	10	jw. tylko w kierunku z
xu_grid =	30	gęstość siatki wzdłuż osi Ox, (przy stosowaniu funkcji **explicit** oraz **parametric_surface**)
yv_grid =	30	gęstość siatki wzdłuż osi Oy, jw.
xyplane =	false	automatycznie ustawia położenia płaszczyzny poziomej, gdy podana jest liczba **a**, to na wysokości **a** (mierzonej na osi Oz)

[1] Zbiór $\{(x,y) \in D_f : f(x,y) = h\}$ jest poziomicą wykresu funkcji f odpowiadającą wartości $h \in \mathbb{R}$. Patrz też rozdział 16.

Aby nie wprowadzać tych samych opcji graficznych wielokrotnie, można zdefiniować własne ustawienia domyślne **set_draw_defaults(**opcja1,opcja2,…**)**. Jeżeli będziemy chcieli przywrócić ustawienia automatyczne, to wystarczy wywołać powyższą funkcję bez wpisywania argumentów.

Tabela 6.5

PODSTAWOWE FUNKCJE PAKIETU DRAW
draw2d(opcje, obiekty_graficzne, …**)** **draw3d(**opcje, obiekty_graficzne, …**)** **gr2d(**opcje, obiekty_graficzne, …**)** **gr3d(**opcje, obiekty_graficzne, …**)** - argumentami powyższych funkcji są opcje i obiekty graficzne **draw(***gr2d*, …, *gr3d*, …, *opcje*, …**)** - łączenie różnych typów obiektów graficznych

Tabela 6.6

PODSTAWOWE OBIEKTY GRAFICZNE PAKIETU DRAW
explicit(f(x),x,x_{min},x_{max}**)** - wykres funkcji jednej zmiennej na przedziale $[x_{min}, x_{max}]$ **explicit(**f(x,y),x,x_{min},x_{max},y,y_{min},y_{max}**)** - wykres funkcji dwóch zmiennych na prostokącie $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ **implicit(**F(x,y)=0,x,x_{min},x_{max},y,y_{min},y_{max}**)** - krzywa opisana równaniem F(x,y)=0, gdzie $(x, y) \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$ **implicit(**F(x,y,z)=0,x,x_{min},x_{max},y,y_{min},y_{max},z,z_{min},z_{max}**)** - powierzchnia opisana równaniem F(x,y,z)=0, gdzie $(x, y, z) \in [x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$ **parametric(**x(t),y(t),t,t_{min},t_{max}**)** - krzywa zadana równaniami parametrycznymi w przestrzeni $\mathbb{R}^2$ **parametric(**x(t),y(t),z(t),t,t_{min},t_{max}**)** - krzywa zadana równaniami parametrycznymi w przestrzeni $\mathbb{R}^3$ **parametric_surface(**x(t,s), y(t,s), z(t,s),t,t_{min},t_{max},s,s_{min},s_{max}**)** - powierzchnia zadana równaniami parametrycznymi w przestrzeni $\mathbb{R}^3$ **points(**[[x1,y1], [x2,y2], ...]**)** - punkty w przestrzeni $\mathbb{R}^2$ **points(**[[x1,y1,z1], [x2,y2,z2], ...]**)** - punkty w przestrzeni $\mathbb{R}^3$ **region(***warunki*,x,x_{min},x_{max},y,y_{min},y_{max}**)** - obszar opisany przez *warunki* **polar(**r(phi),phi,phi_{min},phi_{max}**)** - krzywa biegunowa r=r(phi) **spherical(**f(phi,th),phi,phi_{min},phi_{max},th,th_{min},th_{max}**)** - powierzchnia opisana za pomocą współrzędnych sferycznych **cylindrical(**f(z,phi),z,z_{min},z_{max},phi,phi_{min},phi_{max}**)** - powierzchnia opisana za pomocą współrzędnych walcowych **vector** ([x,y], [ux,uy]) - wektor o współrzędnych [ux,uy] i o początku w punkcie (x,y) **label(**[„tekst”,x,y],…**)** - etykieta „tekst” w punkcie (x,y), ew. kolejne etykiety **bars(**[x_1,h_1,w_1], [x_2,h_2,w_2], … **)** - wykres słupkowy **ellipse(**…**), polygon(**…**), quadrilateral(**…**), rectangle(**…**), triangle(**…**), …** - figury geometryczne - pozostałe obiekty opisane są w [1] i [7]

ZADANIA PRZYKŁADOWE

Przykład 1

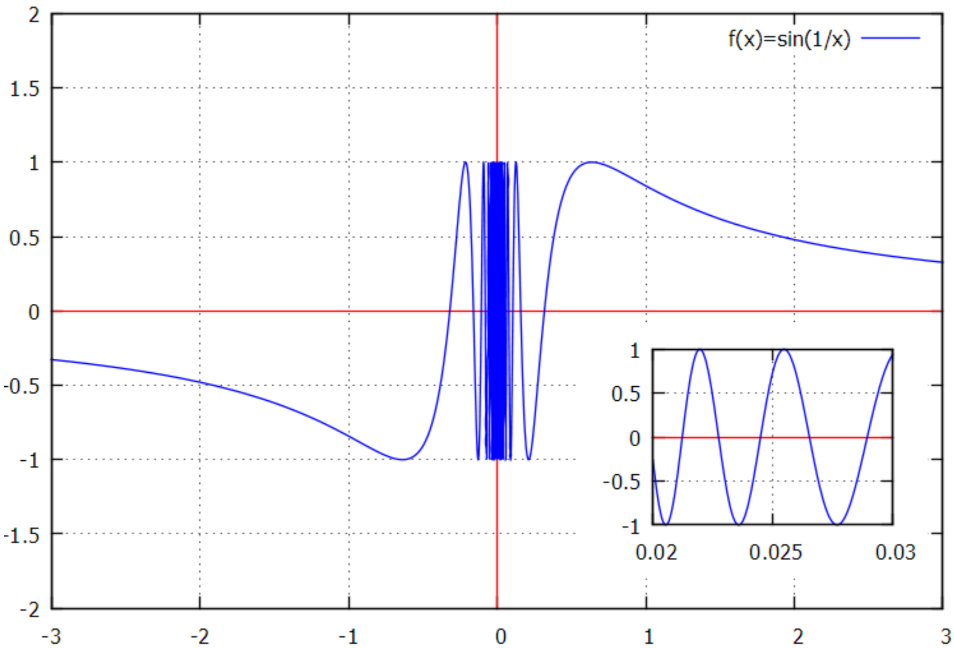
Sprawdzimy (na podstawie wykresu) ile miejsc zerowych ma funkcja

$$f(x) = \sin\left(\frac{1}{x}\right), \;\; x \in \mathbb{R} \setminus \{0\}$$

a) na przedziale $(-3,3)$, b) na przedziale $(0.02,0.03)$.

```
(%i1) load(draw)$   set_draw_defaults(
        grid=true, xaxis=true, yaxis=true, proportional_axes=xy,
        xaxis_color=red, yaxis_color=red, xaxis_type=solid, yaxis_type=solid )$
```

```
(%i3) draw(terminal=wxt,
        gr2d(yrange=[-2,2], key=sconcat("f(x)=",sin(1/x)), explicit(sin(1/x),x,-3,3)),
        gr2d(xtics=0.005, ytics=0.5, allocation=[[0.6,0.2],[0.35,0.3]],
            explicit(sin(1/x),x,0.02,0.03))  )$
```



Powyższy rysunek przedstawia dwa obiekty graficzne (z różnymi ustawieniami).

Przykład 2

Narysujemy poniższe zbiory, korzystając z polecenia **region**.

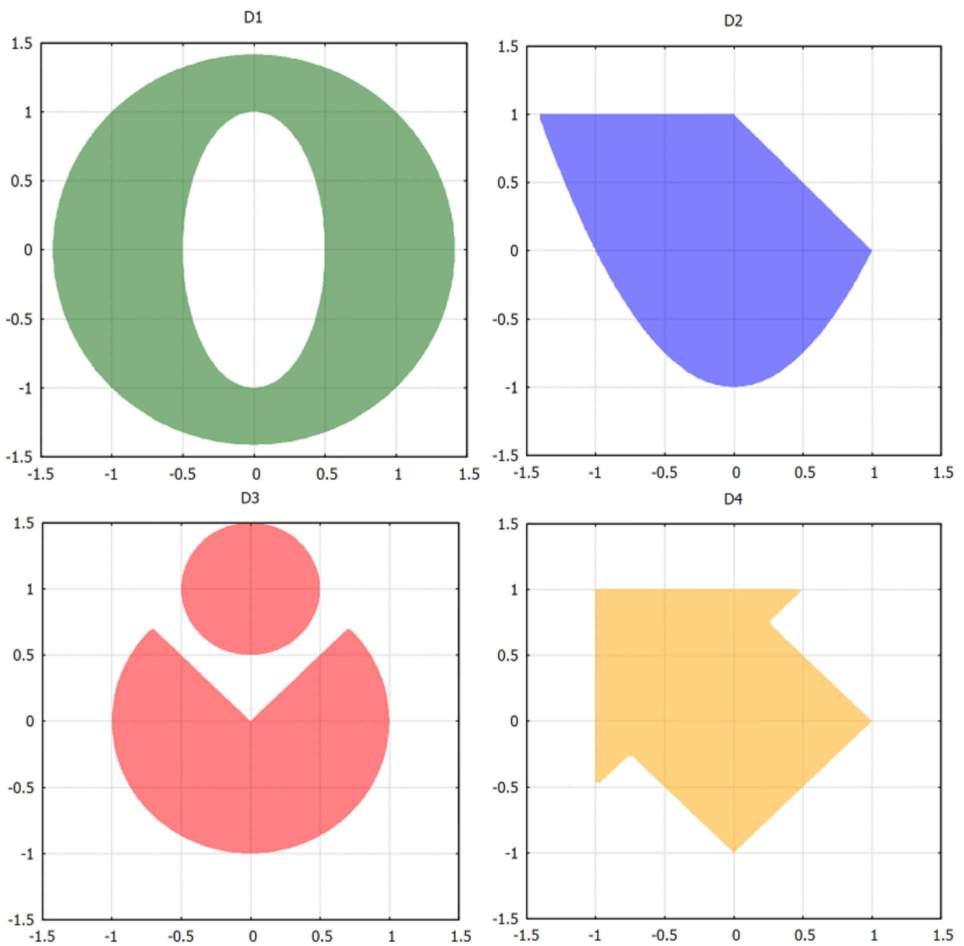
$D_1 = \{(x,y) \in \mathbb{R}^2\colon\; 4\,x^2 + y^2 > 1 \;\wedge\; x^2 + y^2 < 2\,\}$,

$D_2 = \{(x,y) \in \mathbb{R}^2\colon\; y > x^2 - 1 \;\wedge\; y < 1 - x \;\wedge\; y < 1\,\}$,

$D_3 = \left\{(x,y) \in \mathbb{R}^2\colon\; x^2 + y^2 < 1 \wedge\; y < |x|\,\right\} \cup \left\{(x,y) \in \mathbb{R}^2\colon x^2 + (y-1)^2 < \frac{1}{4}\right\}$,

$D_4 = \left\{(x,y) \in \mathbb{R}^2\colon\; |x| + |y| < 1 \;\vee\; \left(y > x + \frac{1}{2} \;\wedge\; x > -1 \;\wedge\; y < 1\right)\right\}$.

```
(%i1) load(draw)$   a:1.5$

(%i3) set_draw_defaults(
      proportional_axes=xy, grid=true, x_voxel=80, y_voxel=80,
      user_preamble="set style fill transparent solid 0.5 noborder" )$

(%i4) draw2d(title="D1", fill_color=dark-green,
             region( 4*x^2+y^2>1 and x^2+y^2 <2, x,-a,a, y,-a,a) )$

(%i5) draw2d(title="D2", fill_color=blue,
             region( y>x^2-1 and y<1-x and y<1, x,-a,a, y,-a,a) )$

(%i6) draw2d(title="D3", region( x^2+y^2<1 and y<abs(x), x,-a,a, y,-a,a),
             region( x^2+(y-1)^2<1/4, x,-a,a, y,-a,a) )$

(%i7) draw2d(title="D4", fill_color=orange,
      region( abs(x)+abs(y)<1 or (y>x+1/2 and x>-1 and y<1), x,-a,a, y,-a,a) )$
```

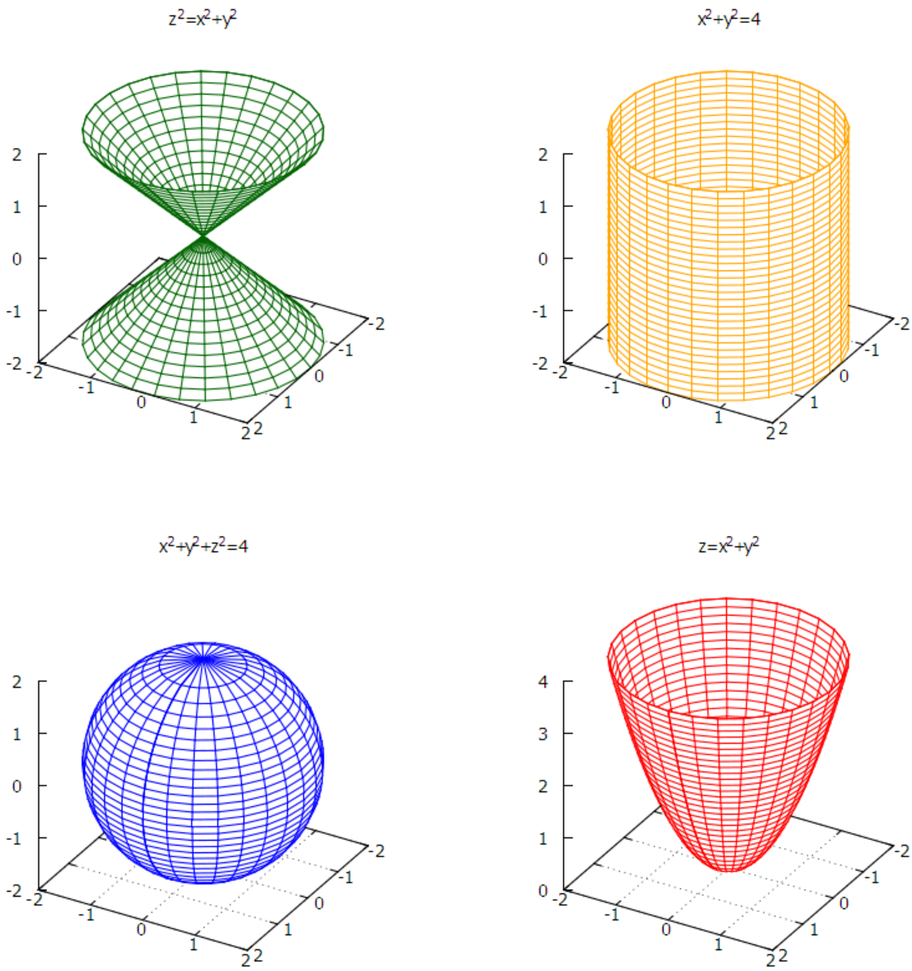


Przykład 3

Narysujemy powierzchnię stożkową, powierzchnię walcową, sferę oraz paraboloidę obrotową wraz z opisującymi je równaniami podanymi w tytule rysunku. Zastosujemy współrzędne walcowe i polecenie **cylindrical**.

```
(%i1) load(draw)$
      set_draw_defaults( xrange = [-2,2], yrange = [-2,2], zrange = [-2,2],
      view = [60,120], dimensions = [800,800], surface_hide = true,
      proportional_axes = xyz, grid = true, xtics =1, ytics =1, ztics = 1 )$
```

```
(%i3) s:gr3d( xyplane = -2, title = "z^2=x^2+y^2", color = dark-green,
          cylindrical(z,z,-2,2,a,0,2*%pi) )$
      w:gr3d( xyplane = -2, title = "x^2+y^2=4", color = orange,
          cylindrical(2,z,-2,2,a,0,2*%pi)  )$
      sf:gr3d( xyplane = -2, title = "x^2+y^2+z^2=4", color = blue,
          cylindrical(sqrt(4-z^2),z,-2,2,a,0,2*%pi) )$
      p:gr3d( xyplane = 0, title = "z=x^2+y^2", color = red,
          cylindrical(sqrt(z),z,0,4,a,0,2*%pi), zrange = [0,4] )$
      draw(terminal=wxt,columns=2,s,w,sf,p)$
```

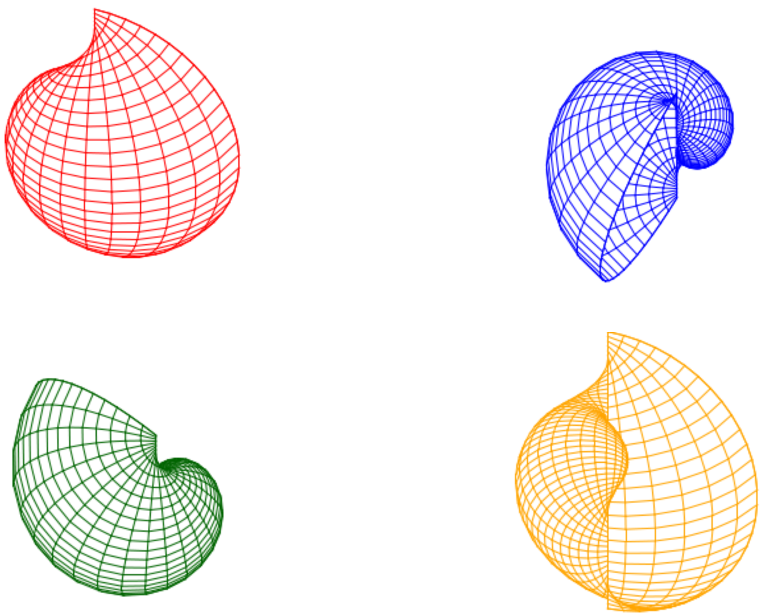


Przykład 4

Zastosujemy polecenie **spherical** oraz opcję **view**, aby przedstawić obraz wybranej powierzchni w zależności od ustawieniach obserwatora.

```
(%i1) load(draw)$
      set_draw_defaults(
      terminal=wxt,
      dimensions=[800,800],
      proportional_axes=xyz,
      surface_hide=true,
      axis_3d=false,
      xtics=none,
      ytics=none,
      ztics=none)$
```

```
(%i3) m:spherical(a/4,a,0,2*%pi,z,0,%pi)$
      p1:gr3d(
         view=[200,140],color=red,m)$
      p2:gr3d(
         view=[20,120],color=blue,m)$
      p3:gr3d(
         view=[50,20],color=dark-green,m)$
      p4:gr3d(
         view=[300,20],color=orange,m)$
      draw( columns = 2, p1, p2, p3, p4 )$
```



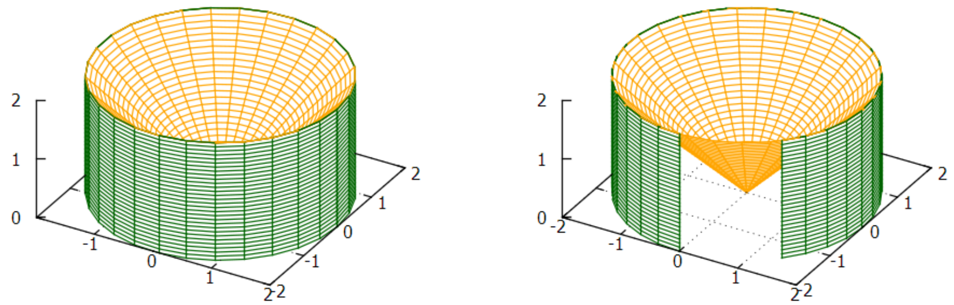
Przykład 5

Przedstawimy możliwości zastosowania współrzędnych walcowych i sferycznych do zilustrowania przekrojów powierzchni walcowej, powierzchni stożkowej, sfery i paraboloidy.

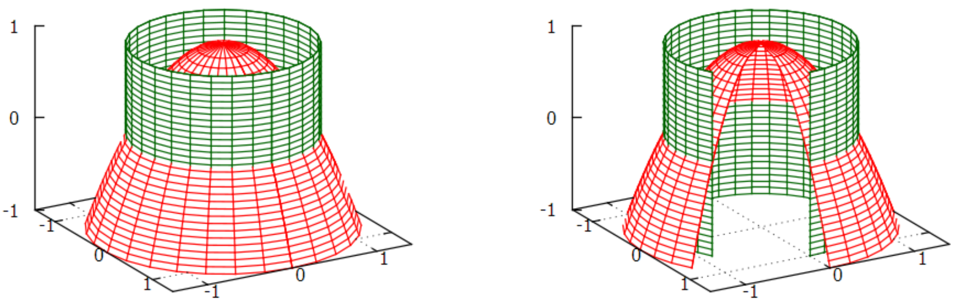
Dla każdej z opisanych poniżej powierzchni wykonamy dwa rysunki. Jeden zgodny z zadeklarowanymi zakresami zmiennej *z* oraz drugi z wyciętym fragmentem, by zobaczyć obiekt „od środka”.

Zastosowanie współrzędnych walcowych daje bardziej „eleganckie” rezultaty niż stosowanie polecenia **explicit**, i zdecydowanie lepsze niż **implicit**.

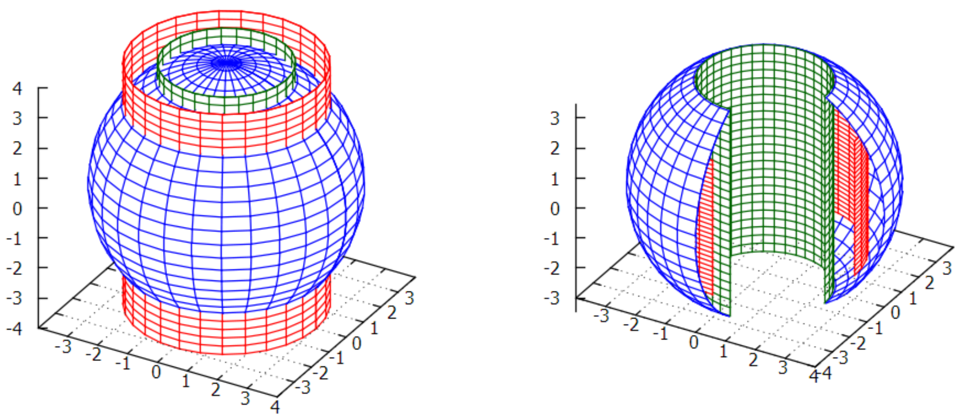
a) Powierzchnia stożkowa opisana równaniem $z = \sqrt{x^2 + y^2}$ dla $z \in [0{,}2]$ oraz powierzchnia walcowa opisana równaniem $x^2 + y^2 = 4$ dla $z \in [0{,}2]$.



b) Paraboloida opisana równaniem $z = 1 - x^2 - y^2$ dla $z \geq -1$ oraz powierzchnia walcowa opisana równaniem $x^2 + y^2 = 1$ dla $z \in [-1{,}1]$.



c) Sfera opisana równaniem $x^2 + y^2 + z^2 = 16$ oraz dwie powierzchnie walcowe opisane równaniami: $x^2 + y^2 = 4,\ x^2 + y^2 = 9$ dla $z \in [-4\,{,}4]$.



Na rysunku z prawej strony powierzchnia walcowa (zaznaczona kolorem zielonym) oraz sfera są rysowane dla $z \in [-2\sqrt{3}\,{,}2\sqrt{3}]$, natomiast druga powierzchnia walcowa (zaznaczona kolorem czerwonym) dla $z \in [-\sqrt{5}, \sqrt{5}]$. W ten sposób wyraźnie zaznaczone zostały przecięcia powierzchni walcowych ze sferą.
Poniżej skrypty Maximy, które pozwoliły wygenerować powyższe rysunki.

```
(%i1) load(draw)$
      set_draw_defaults( surface_hide=true, xtics=1, ytics=1, ztics=1, grid=true,
                         proportional_axes=xyz, dimensions=[800,800])$
```

a)

```
(%i3) g1a:gr3d(zrange=[0,2], xyplane=0,
        color=orange,
        cylindrical(z,z,0,2,a,0,2*%pi),
        color=dark-green,
        cylindrical(2,z,0,2,a,0,2*%pi))$
(%i4) g2a:gr3d(zrange=[0,2], xyplane=0,
        color=orange,
        cylindrical(z,z,0,2,a,0,2*%pi),
        color=dark-green,
        cylindrical(2,z,0,2,a,-%pi/4,3/2*%pi))$
(%i5) draw(columns=2,g1a,g2a)$
```

b)

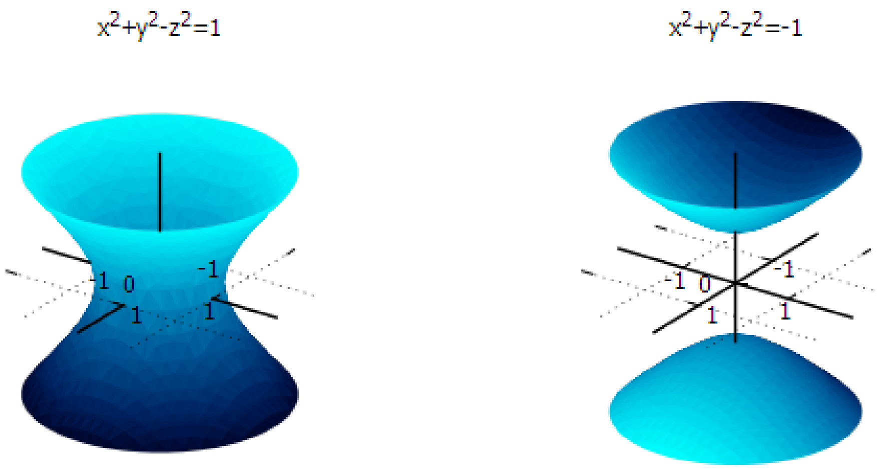
```
(%i6) g1b:gr3d(xyplane=-1, view=[75,60],
        color=dark-green,
        cylindrical(1,z,-1,1,a,0,2*%pi),
        color=red,
        cylindrical(sqrt(1-z),z,-1,1,a,0,2*%pi))$
(%i7) g2b:gr3d(xyplane=-1, view=[75,60],
        color=dark-green,
        cylindrical(1,z,-1,1,a,0,5/3*%pi),
        color = red,
        cylindrical(sqrt(1-z),z,-1,1,a,0,5/3*%pi) )$
(%i8) draw(columns=2,g1b,g2b)$
```

c)

```
(%i9) g1c:gr3d(xyplane=-4,
        zrange=[-4,4],
        color=red,
        cylindrical(3,z,-4,4,a,0,2*%pi),
        color = dark-green,
        cylindrical(2,z,-4,4,a,0,2*%pi),
        color = blue,
        spherical(4,ph,0,2*%pi,th,0,%pi))$
(%i10) g2c:gr3d(xyplane=-3,
        zrange=[-2*sqrt(3),2*sqrt(3)],
        color=red,
        cylindrical(3,z,-sqrt(5),sqrt(5),a,0,3/2*%pi),
        color = dark-green,
        cylindrical(2,z,-4,4,a,0,3/2*%pi),
        color = blue,
        spherical(4,ph,0,3/2*%pi,th,0,%pi) )$
(%i11) draw(columns=2,g1c,g2c)$
```

Przykład 6

Narysujemy hiperboloidę jednopowłokową oraz hiperboloidę dwupowłokową, korzystając z polecenia **implicit**.




```
(%i1) load(draw)$
      set_draw_defaults( terminal=wxt, zrange=[-2.2,2.2], view=[65,125],
      x_voxel=25, y_voxel=25, enhanced3d=true, colorbox=false,
      xaxis=true, xaxis_type=solid, xaxis_color=black, xaxis_width=1.5,
      yaxis=true, yaxis_type=solid, yaxis_color=black, yaxis_width=1.5,
      zaxis=true, zaxis_type=solid, zaxis_color=black, zaxis_width=1.5,
      xtics_axis =true,ytics_axis =true,ztics_axis =true,  proportional_axes =xyz,
      xtics={-1,1}, ytics={-1,1}, ztics={0}, xyplane =0, grid=true,
      palette = [0,3,7], axis_3d=false, surface_hide=true)$
```

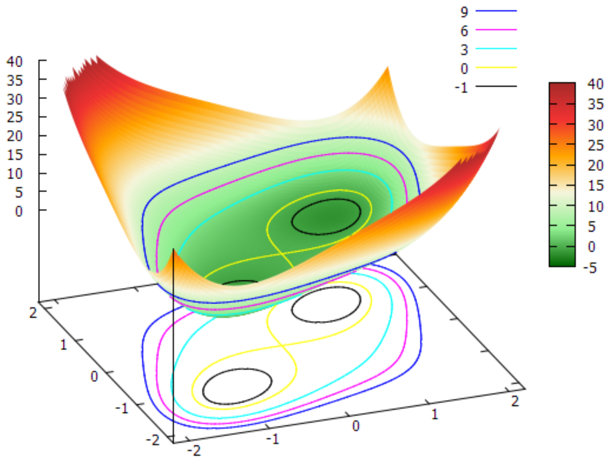
```
(%i3) g1:gr3d( title="x^2+y^2-z^2=1",
            implicit( x^2+y^2-z^2=1, x,-2.2,2.2, y,-2.2,2.2, z,-1.9,1.9)  )$
      g2:gr3d( title="x^2+y^2-z^2=-1", enhanced3d=[x,x,y,z],
            implicit( x^2+y^2-z^2=-1, x,-2.2,2.2, y,-2.2,2.2, z,-2.2,2.2)  )$
      draw( columns=2, g1, g2)$
```

Przykład 7

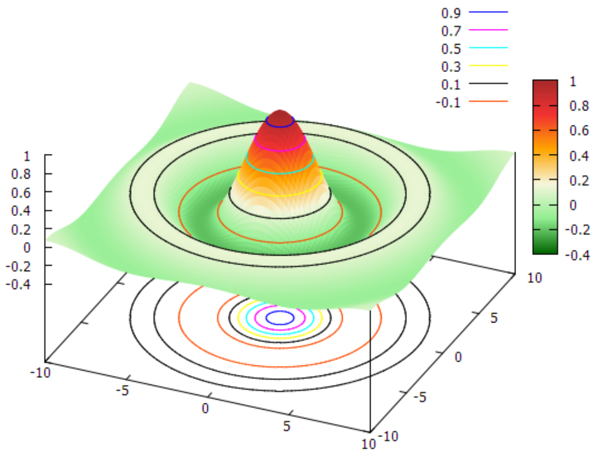
Dla funkcji f dwóch zmiennych narysujemy jej wykres na zadanym prostokącie oraz *krzywe konturowe* (przecięcie płaszczyzny $z = k$ z powierzchnią $z = f(x, y)$) i *poziomice* ($\{(x, y) \in D_f : f(x, y) = k\}$) dla podanych wartości k.

W podpunkcie a) dla $k = 0$ otrzymamy krzywą zwaną lemniskatą Bernoulliego, natomiast w podpunkcie c) dla $k = 0$ otrzymamy liść Kartezjusza.

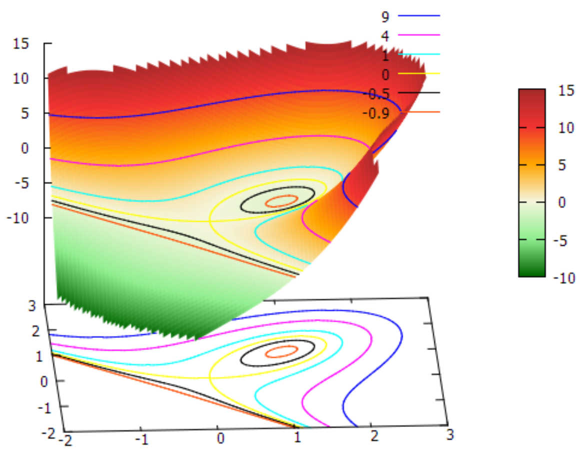
a) $f(x,y) = x^4 + y^4 - 4xy, \quad (x,y) \in [-2,2]^2, \quad k \in \{-1,0,3,6,9\}$



b) $f(x,y) = \frac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}, \quad (x,y) \in [-10,10]^2, \quad k \in \{-0.1,0.1,0.3,0.5,0.7,0.9\}$



c) $f(x,y) = x^3 + y^3 - 3xy, (x,y) \in [-2,3]^2, k \in \{-0.9,-0.5,0,1,4,9\}$

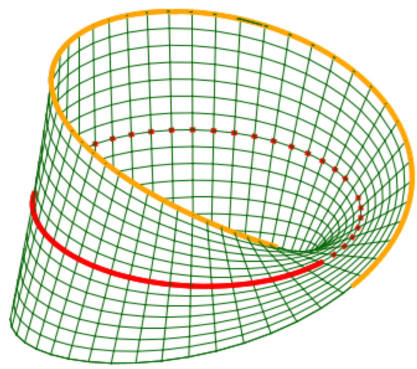


```
(%i1) load(draw)$
      set_draw_defaults( terminal = wxt, enhanced3d = true,
          xtics = 1, ytics = 1, contour = both, surface_hide = true,
          palette = [dark-green,light-green,beige, orange,light-red,brown],
          xu_grid = 100, yv_grid = 100   )$

(%i3) f(x,y):=x^4+y^4-4*x*y$

(%i4) draw3d(
      zrange = [-3,40],
      explicit(f(x,y), x,-2.2,2.2, y,-2.2,2.2),
      contour_levels = {-1,0,3,6,9},
      surface_hide = true
      )$

(%i5) f(x,y):=sin(sqrt(x^2+y^2))/sqrt(x^2+y^2)$

(%i6) draw3d(
      zrange=[-0.5,1],
      explicit(f(x,y), x, -10, 10, y, -10,10),
      contour_levels = [-0.5,.2,1],
      xtics=5,ytics=5  )
      $

(%i7) f(x,y):=x^3+ y^3-3*x*y$

(%i8) draw3d(
      zrange=[-10,15],
      explicit(f(x,y), x, -2, 3, y, -2,3),
      contour_levels = {-0.9,-1/2,0,1,4,9}
      )$
```

Przykład 8

Na przykładzie wstęgi Möbiusa pokażemy sposób rysowania krzywej oraz powierzchni danej równaniami parametrycznymi.

```
(%i1) load(draw)$ set_draw_defaults( xu_grid=50, yv_grid=15, nticks=200,
      surface_hide=true, axis_3d=false, xtics=none, ytics=none, ztics=none)$

(%i3) draw3d( terminal=wxt, color=dark-green,
      parametric_surface(
        cos(u)*(3+v*cos(u/2)), sin(u)*(3+v*cos(u/2)), v*sin(u/2),
        u,-%pi,%pi, v,-1,1), line_width=4, color=red,
      parametric(
        cos(u)*3, sin(u)*3,0,u,-%pi,%pi), color=orange,
      parametric(
        cos(u)*(3+cos(u/2)), sin(u)*(3+cos(u/2)), sin(u/2), u,0,2*%pi)  )$
```



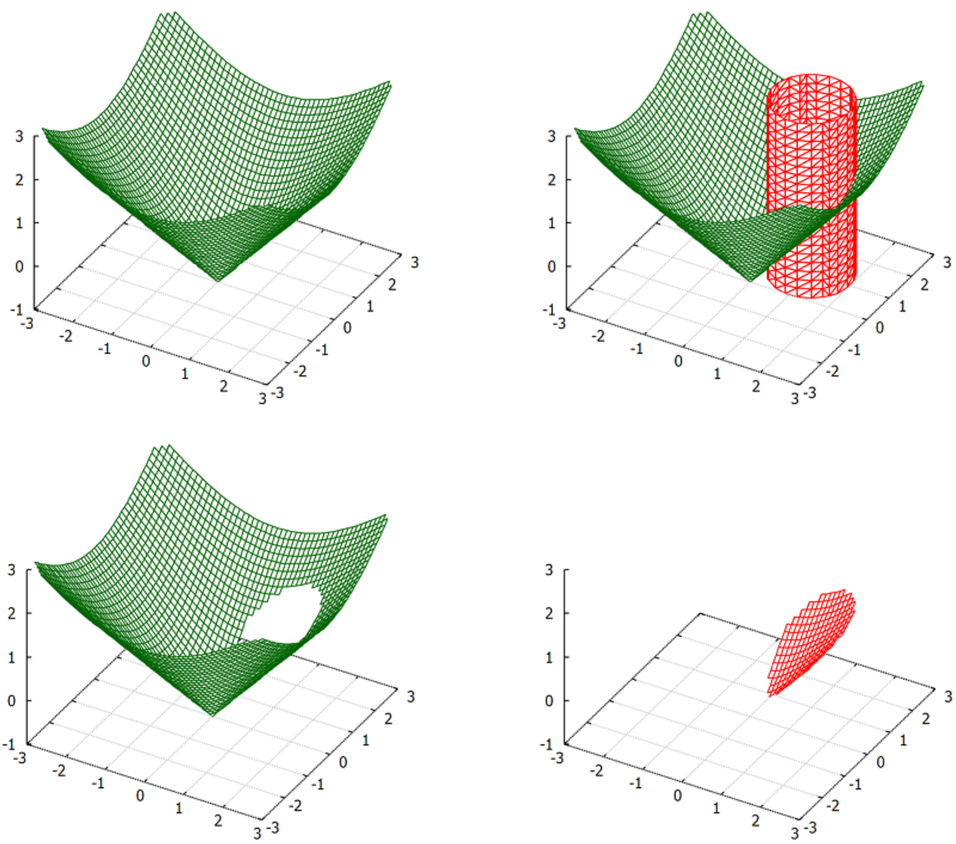
Przykład 9

Ilustracja „wycinania” płata z powierzchni stożkowej opisanej równaniem

$$(z+1)^2 = x^2 + y^2,\ \ z \geq -1$$

przez powierzchnię walcową daną równaniem

$$x^2 + y^2 - 2x - 2y + 1 = 0.$$



```
(%i1) load(draw)$
      set_draw_defaults(
        proportional_axes = xyz,
        xyplane = -1, grid = true,
        zrange = [-1,3], ztics = 1,
        surface_hide = true, enhanced3d = false,
        x_voxel = 20, y_voxel = 20, z_voxel = 20,
        xu_grid = 50, yv_grid = 50  )$
```

```
(%i3) f(x,y):= sqrt(x^2+y^2)-1$
      s(x,y):= x^2+y^2-2*x-2*y+1$
      g1(x,y):= if s(x,y)<0 then f(x,y)$
      g2(x,y):= if s(x,y)>0 then f(x,y)$
```

```
(%i7) p1:gr3d(color = dark-green, explicit( f, x,-3,3, y,-3,3) )$
      p2:gr3d(color = dark-green, explicit( f, x,-3,3, y,-3,3),
              color = red, implicit( s(x,y)=0, x,-3,3, y,-3,3, z,-1,3) )$
      p3:gr3d( xu_grid = 50, yv_grid = 50,
              color=dark-green, explicit( g2, x,-3,3, y,-3,3) )$
      p4:gr3d( xu_grid = 50, yv_grid = 50,
              color = red, explicit( g1, x,-3,3, y,-3,3) )$
```

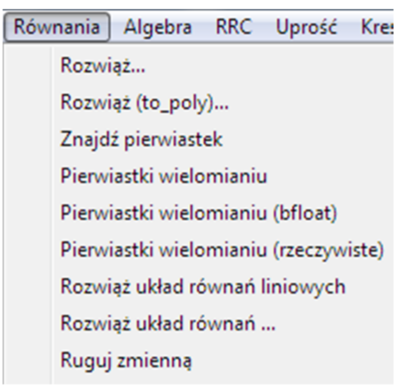
```
(%i11) draw(columns=2,p1,p2)$  draw(columns=2,p3,p4)$
```

7. Rozwiązywanie równań i nierówności

Tabela 7.1

POLECENIA	OPIS
solve([r],[X]**)**	symboliczne rozwiązywanie równania algebraicznego (układu równań algebraicznych) r względem zmiennej (zmiennych) X
find_root(f,a,b**)**	numeryczne wyznaczanie pierwiastka funkcji f w przedziale $[a, b]$
find_root(r,x,a,b**)**	numeryczne wyznaczanie pierwiastka równania r w przedziale $[a, b]$
allroots(w**)**	numeryczne wyznaczanie pierwiastków zespolonych wielomianu w
bfallroots(w**)**	jw. tylko wyniki o zwiększonej precyzji
realroots(w**)**	wyznaczanie wszystkich pierwiastków rzeczywistych wielomianu w
linsolve(u,X**)**	rozwiązywanie układu równań liniowych, gdzie u - lista równań, X - lista niewiadomych
eliminate([r],[x]**)**	eliminowanie zmiennej x z równania (układu równań) r
lhs(r**)**	lewa strona równania r
rhs(r**)**	prawa strona równania r
algsys(u,X**)**	numeryczne wyznaczanie pierwiastków układu równań algebraicznych, gdzie u lista równań, X lista niewiadomych
multiplicities	krotność pierwiastka równania
solve_rat_ineq(n**)**	rozwiązywanie nierówności wymiernych [1]
fourier_elim([n],[x,y]**)**	rozwiązywanie (przekształcanie) układów nierówności liniowych [2]

[1] Stosuje się do nierówności z jedną zmienną, po wczytaniu pakietu **solve_rat_ineq**.
[2] Wymaga wczytania pakietu **fourier_elim.**



solve([%], [x]**)**
to_poly_solve([%], [x]**)**
find_root("(%),x,-1,1**)**
allroots(%**)**
bfallroots(%**)**
realroots(%**)**
linsolve([…], […]**)**
algsys([…], […]**)**
eliminate([%],[…]**)**

Powyżej polecenia dostępne w **Menu-Równania** i ich odpowiedniki z tabeli 7.1.

Uwaga 1

Polecenie **solve** pozwala rozwiązać większości równań wielomianowych, wymiernych, a czasami udaje się je zastosować do równań pierwiastkowych, z modułami, logarytmicznych oraz najprostszych równań trygonometrycznych.

Uwaga 2

Polecenie **algsys** wyznacza pierwiastki zespolone. Jeśli chcemy otrzymać tylko pierwiastki rzeczywiste, to trzeba wcześniej zadeklarować **realonly**:true.

Menu-Równania/Rozwiąż

(%i1) solve(a*x-1/a, x);

(%o1) $[x=\frac{1}{a^2}]$

(%i2) solve(a*x-1/a, a);

(%o2) $[a=-\frac{1}{\sqrt{x}}, a=\frac{1}{\sqrt{x}}]$

(%i3) solve(%e^x-5);

(%o3) $[x=\log(5)]$

(%i4) solve(log(x)=4);

(%o4) $[x=\%e^4]$

(%i5) solve(cos(x)^2=1/2);

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

(%o5) $[x=\frac{3\pi}{4}, x=\frac{\pi}{4}]$

Równania trygonometryczne są rozwiązywane na pewnym przedziale związanym z odpowiednią funkcją cyklometryczną, zatem nie są to wszystkie rozwiązania.

(%i6) r:solve(x^3-1);

(%o6) $[x=\frac{\sqrt{3}\,\%i-1}{2}, x=-\frac{\sqrt{3}\,\%i+1}{2}, x=1]$

r jest listą pierwiastków podaną w formie równości.

(%i7) rhs(r[2]);

(%o7) $-\frac{\sqrt{3}\,\%i+1}{2}$

Prawa strona drugiego elementu listy *r*.

(%i8) fpprintprec:5$

Zmieniamy dokładność wyświetlania wyników.

Równania/Pierwiastki wielomianu

(%i9) allroots(x^3-2);

(%o9) $[x=1.2599, x=1.0911\,\%i-0.629, x=-1.0911\,\%i-0.629]$

```
(%i10) realroots(x^3-2);%,numer;
```
(%o10) $[x = \frac{42275935}{33554432}]$

(%o11) $[x = 1.2599]$

Równania/Pierwiastki wielomianu (rzeczywiste)

```
(%i12) find_root(x^3-2, x, 1, 2);
```
(%o12) 1.2599

Równania/Znajdź pierwiastek

```
(%i13) linsolve([x-a*y=2,b*x-y=1],[x,y]);
```
(%o13) $[x = \frac{a-2}{a\,b-1}, y = -\frac{2\,b-1}{a\,b-1}]$

Równania/Rozwiąż układ równań liniowych

```
(%i14) algsys([x^2+y^2=2*x,x-y=2],[x,y]);
```
(%o14) $[[x = 2, y = 0], [x = 1, y = -1]]$

Równania/Rozwiąż układ równań …

```
(%i15) eliminate([x^2-3*y=7,x+5*y=7],[y]);
```
(%o15) $[-5\,x^2 - 3\,x + 56]$

Równania/Ruguj zmienną

```
(%i16) solve(x^4-5*x^3+9*x^2-7*x+2);
       multiplicities;
```
(%o16) $[x = 2, x = 1]$

(%o17) $[1, 3]$

2 jest pierwiastkiem jednokrotnym, a 1 jest pierwiastkiem trzykrotnym rozważanego wielomianu.

```
(%i18) load(solve_rat_ineq)$
```

Wczytujemy pakiet.

```
(%i19) solve_rat_ineq(2*x^2-7*x+3<0);
```
(%o19) $[[x > \frac{1}{2}, x < 3]]$

Wynik czytamy:

$$x \in \left(\frac{1}{2}, 3\right)$$

```
(%i20) solve_rat_ineq((3*x-1)/(x-2)>=0);
```
(%o20) $[[x <= \frac{1}{3}], [x > 2]]$

Wynik czytamy:

$$x \in \left(-\infty, \frac{1}{3}\right] \cup [2, \infty)$$

ZADANIA PRZYKŁADOWE

Przykład 1

Rozwiązać równania:

a) $2x^4 + x^3 - 11x^2 - 5x + 5 = 0$, b) $x^4 - 3x^3 - 2x^2 + 2x + 12 = 0$,

c) $\ln^2 x + 2\ln x - 35 = 0$, d) $9^x - 12 \cdot 3^x + 27 = 0$.

a)

```
(%i1) solve(2*x^4+x^3-11*x^2-5*x+5);
```
(%o1) $[x = \frac{1}{2}, x = -1, x = -\sqrt{5}, x = \sqrt{5}]$

b)

```
(%i2) solve(x^4-3*x^3-2*x^2+2*x+12);
(%o2) [x =3,x =2,x = -%i -1,x =%i -1]
```

c)

```
(%i3) solve(log(x)^2+2*log(x)-35);
(%o3) [x =%e^5,x =%e^-7]
```

d)

```
(%i4) solve(9^x-12*3^x+27=0);
(%o4) [9^x =4 3^(x+1) -27]
```

```
(%i5) solve(9^x-12*3^x+27=0),solveradcan:true;
(%o5) [x =1,x =2]
```

W podpunkcie d) zostało zastosowane (lokalnie) ustawienie **solveradcan:true**, które, dzięki zastosowaniu uproszczeń z **radcan**, poprawia skuteczność polecenia **solve** w rozwiązywaniu równań wykładniczych i logarytmicznych.

Przykład 2

Rozwiążemy nierówności:

a) $(x^3-2x^2-9)(x^2-25)>0$, b) $\frac{x^5-4x^4+5x^3-2x^2}{x^3+2x^2-5x-6}\leq 0$, c) $\frac{3x-1}{2x-1}\geq\frac{x+3}{x-2}$,
d) $\ln^2 x-\ln x-12>0$, e) $e^{2x}-10e^x+24>0$.

Najpierw wczytamy odpowiedni pakiet.

```
(%i1) load(solve_rat_ineq)$
```

a)

```
(%i2) solve_rat_ineq((x^3-2*x^2-9)*(x^2-25)>0);
(%o2) [[x> -5,x<3],[x>5]]
```

Otrzymany wynik odczytujemy następująco: $x\in(-5,3)\cup(5,\infty)$.

b)

Przypiszmy lewą stronę nierówności do zmiennej w. Wyznaczymy miejsca zerowe mianownika, by określić dziedzinę tej nierówności.

```
(%i3) w:(x^5-4*x^4+5*x^3-2*x^2)/(x^3+2*x^2-5*x-6)$
```

```
(%i4) solve(denom(w));
(%o4) [x = -3,x = -1,x =2]
```

Dziedziną jest więc zbiór $\mathrm{D}=\mathbb{R}\setminus\{-3,-1,2\}$.

```
(%i5) solve_rat_ineq(w<=0);
(%o5) [[x> -3,x< -1],[x =0],[x =1]]
```

Otrzymany wynik odczytujemy następująco: $x\in(-3,-1)\cup\{0,1\}$.

Mogliśmy też przedstawić wyrażenie wymierne w w postaci iloczynowej

```
(%i6) w,factor;
```
(%o6) $\frac{(x-1)^2 x^2}{(x+1)(x+3)}$

i na tej podstawie próbować wyznaczyć dziedzinę oraz rozwiązanie nierówności. Jednak w tym przypadku obraz jest niepełny, gdyż pewne czynniki uprościły się.

c)

Dziedziną tej nierówności jest zbiór $\mathrm{D} = \mathbb{R} \setminus \left\{\frac{1}{2}, 2\right\}$.

```
(%i7) solve_rat_ineq((3*x-1)/(2*x-1)>=(x+3)/(x-2));
```
(%o7) $[[x <= 6 - \sqrt{31}],[x > \frac{1}{2}, x < 2],[x >= \sqrt{31} + 6]]$

Wynik odczytujemy: $x \in \left(-\infty, 6 - \sqrt{31}\right] \cup \left(\frac{1}{2}, 2\right) \cup \left[\sqrt{31} + 6, \infty\right)$.

d)

Dziedziną rozważanej nierówności jest zbiór $\mathrm{D} = (0, \infty)$. Nierówność tę sprowadzimy (przez podstawienie $\ln x = t$) do nierówności wielomianowej.

```
(%i8) n1:log(x)^2-log(x)-12>0;
```
(%o8) $\log(x)^2 - \log(x) - 12 > 0$

```
(%i9) n2:n1,log(x)=t;
```
(%o9) $t^2 - t - 12 > 0$

```
(%i10) r:solve_rat_ineq(n2);
```
(%o10) $[[t < -3],[t > 4]]$

```
(%i11) r,t=log(x);
```
(%o11) $[[\log(x) < -3],[\log(x) > 4]]$

Pomocniczo rozwiążemy dwa równania:

```
(%i12) solve(log(x)=-3);
```
(%o12) $[x = \%e^{-3}]$

```
(%i13) solve(log(x)=4);
```
(%o13) $[x = \%e^4]$

Po uwzględnieniu monotoniczności funkcji logarytmicznej i dziedziny D rozważanej nierówności logarytmicznej otrzymujemy, że $x \in (0, e^{-3}) \cup (e^4, \infty)$.

e)

Dziedziną rozważanej nierówności jest zbiór $\mathrm{D} = \mathbb{R}$. Nierówność tę sprowadzimy (przez podstawienie $e^x = t,\ t \in (0, \infty)$) do nierówności wielomianowej

$$t^2 - 10t + 24 > 0\,.$$

```
(%i14) solve_rat_ineq(t^2-10*t+24>0);
(%o14) [[t<4],[t>6]]
```

Po powrocie do zmiennej x otrzymujemy alternatywę dwóch nierówności wykładniczych.

```
(%i15) %,t=exp(x);
(%o15) [[%e^x<4],[%e^x>6]]
```

Pomocniczo rozwiążemy dwa równania:

```
(%i16) solve(%e^x=4);
(%o16) [x=log(4)]
```

```
(%i17) solve(%e^x=6);
(%o17) [x=log(6)]
```

Po uwzględnieniu monotoniczności funkcji wykładniczej i dziedziny D rozważanej nierówności otrzymujemy, że $x \in (-\infty, \ln 4) \cup (\ln 6, \infty)$.

Przykład 3

Wyznaczymy z dokładnością do 5 cyfr znaczących pierwiastki poniższych funkcji w zadanych przedziałach:

a) $fa(x) = e^x - x^2,\ x \in \left[-1, \frac{1}{2}\right]$, b) $fb(x) = \sin x + e^x + x,\ x \in \left[-\frac{1}{2}, \frac{1}{2}\right]$,

c) $fc(x) = 3\cos x + x - 1,\ x \in [-2,2]$.

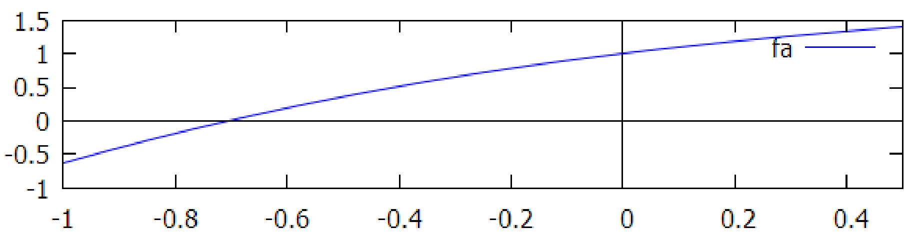
Najpierw zdefiniujemy funkcje.

```
(%i1) fa(x):=exp(x)-x^2$ fb(x):=sin(x)+exp(x)+x$ fc(x):=3*cos(x)+x-1$
```

Następnie sprawdzimy na wykresie ile miejsc zerowych posiada każda z funkcji w zadanym przedziale i wyznaczymy te miejsca zerowe.

a)

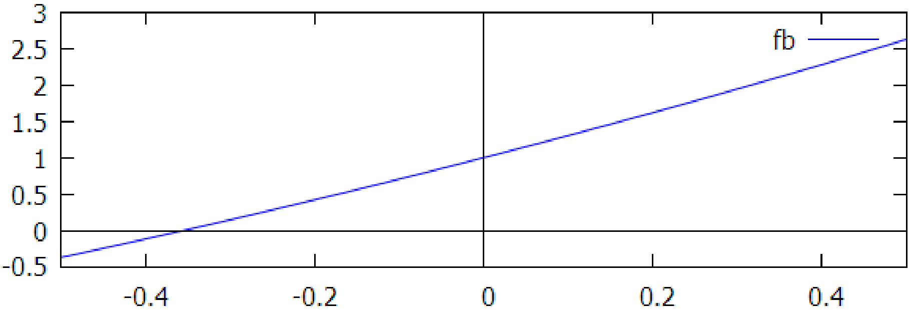
```
(%i4) plot2d([fa(x)],[x,-1,0.5],[yx_ratio,0.2],[legend,"fa"],[axes,solid])$
```




```
(%i5) pierwiastek_fa:find_root(fa, -1, 0.5),bfloat,fpprec=5;
(%o5) -7.0347b-1
```

b)

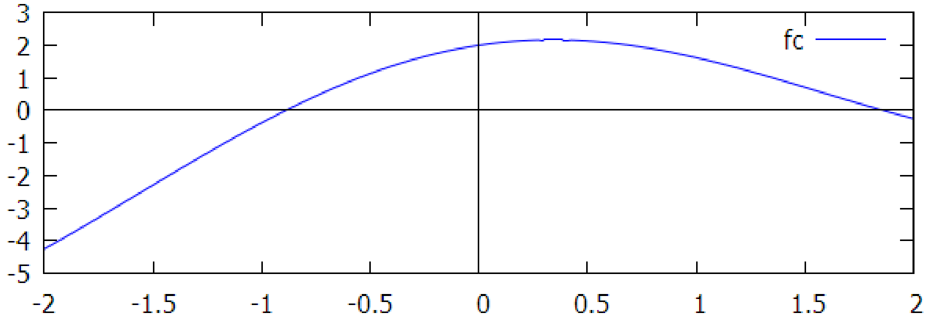
```
(%i6) plot2d([fb(x)],[x,-0.5,0.5],[yx_ratio,0.3],[legend,"fb"],[axes,solid])$
```




```
(%i7) pierwiastek_fb:find_root(fb,-0.5,0.5),bfloat,fpprec=5;
(%o7) -3.5446b-1
```

c)

```
(%i8) plot2d([fc(x)],[x,-2,2],[yx_ratio,0.3],[legend,"fc"],[axes,solid])$
```




```
(%i9) pierwiastek1_fc:find_root(fc,-2,0),bfloat,fpprec=5;
      pierwiastek2_fc:find_root(fc,0,2),bfloat,fpprec=5;
(%o9) -8.8947b-1
(%o10) 1.8624b0
```

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Wyznaczyć wszystkie pierwiastki wielomianów i podać ich krotności:

a) $W_1(x) = x^7 + 2x^6 - 12x^5 - 24x^4 + 48x^3 + 96x^2 - 64x - 128$,

b) $W_2(x) = x^9 + x^8 - 2x^5 - 2x^4 + x + 1$.

Zadanie 2

Podać z dokładnością do 4 cyfr znaczących wszystkie pierwiastki wielomianów:

a) $W_1(x) = 4x^3 - 9x^2 - x + 2$, b) $W_2(x) = x^5 - 5x^3 + 9x^2 - x + 2$.

Zadanie 3

Wyznaczyć z dokładnością do 6 cyfr znaczących pierwiastki poniższych funkcji na podanych przedziałach:

a) $f(x) = x - e^{-x}$, $x\epsilon[0,1]$, b) $f(x) = x - \cos x + e^{x} - 4$, $x\epsilon[1,2]$,

c) $f(x) = 2\sin x + x - 3$, $x\epsilon[2,6]$.

Zadanie 4

Rozwiązać układy równań: a) $\begin{cases} x^2 - 2y^2 = 2 \\ x + y^2 = 3 \end{cases}$, b) $\begin{cases} x^2 - y = -2 \\ 2y + 3x^2 = 3 \end{cases}$.

Zadanie 5

Rozwiązać równania:

a) $3x^4 - 7x^3 - 7x^2 + 21x - 6 = 0$, b) $x^5 + x^3 + 2x^2 - 12x + 8 = 0$,

c) $\ln^2 x + 11\ln x + 24 = 0$, d) $2 \cdot 16^{x} - 33 \cdot 4^{x} + 16 = 0$,

e) $\sin^2 x = \frac{3}{4}$, $x\epsilon\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

Zadanie 6

Rozwiązać nierówności:

a) $x^3 - 2x^2 - 5x + 6 < 0$, b) $\frac{x^4 - 9x^3 + 23x^2 - 3x - 36}{x^3 + x^2 - 4x - 4} \geq 0$, c) $\frac{2x-1}{x+1} \leq \frac{3x-4}{x-3}$,

d) $\frac{(e^{2x}-1)(e^{2x}-e^{x+4})}{e^{x}-3} < 0$, e) $\frac{\ln^3 x - 7\ln x - 6}{\ln x} \geq 0$.

Odpowiedzi

1. a) 2 (trzykrotny), -2 (czterokrotny); b) -1 (trzykrotny), 1 (dwukrotny) $-i$ (dwukrotny), i (dwukrotny).
2. a) $x = 4.637 \cdot 10^{-1}$, $x = -4.765 \cdot 10^{-1}$, $x = 2.263$;
   b) $x = 4.69 \cdot 10^{-1} i - 8.175 \cdot 10^{-3}$, $x = -4.69 \cdot 10^{-1} i - 8.175 \cdot 10^{-3}$, $x = -2.885$, $x = 1.023i + 1.451$, $x = 1.451 - 1.023i$.
3. a) $5.67143 \cdot 10^{-1}$, b) 1.16964, c) 3.28415, 4.94578.
4. a) $(-4, -\sqrt{7})$, $(-4, \sqrt{7})$, $(2, -1)$, $(2,1)$; b) $\left(-\frac{i}{\sqrt{5}}, \frac{9}{5}\right)$, $\left(\frac{i}{\sqrt{5}}, \frac{9}{5}\right)$.
5. a) $x = 2$, $x = \frac{1}{3}$, $x = -\sqrt{3}$, $x = \sqrt{3}$; b) $x = -2i$, $x = 2i$, $x = -2$, $x = 1$;
   c) $x = e^{-3}$, $x = e^{-8}$; d) $x = -\frac{1}{2}$, $x = 2$; e) $x = -\frac{\pi}{3}$, $x = \frac{\pi}{3}$.
6. a) $x \in (-\infty, -2) \cup (1, 3)$, b) $x \in (-2, 2) \cup \{3\} \cup [4, \infty)$,
   c) $x \in (-\infty, -7] \cup (-1, 1] \cup (3, \infty)$, d) $x \in (-\infty, 0) \cup (\ln 3, 4)$,
   e) $x \in (0, e^{-2}] \cup [e^{-1}, 1) \cup [e^3, \infty)$.

8. Wektory, macierze i układy równań liniowych

8.1. Wektory

Tabela 8.1. Operacje na wektorach

POLECENIA	OPIS
x:[x1, x2, … , xk]	$\vec{x} = [x_1, x_2, \dots, x_k]$
x[j]	j-ta współrzędna wektora $\vec{x}$ albo indeks zmiennej x
lmin(x)	współrzędna wektora $\vec{x}$ o najmniejszej wartości
lmax(x)	współrzędna wektora $\vec{x}$ o największej wartości
x±y	$\vec{x} \pm \vec{y}$
x.y	$\vec{x} \circ \vec{y}$ (iloczyn skalarny)
sqrt(x.x)	$\lvert\vec{x}\rvert$ (długość wektora)
express(x~y) [1)]	$\vec{x} \times \vec{y}$ (iloczyn wektorowy)
express((x~y).z) [1)]	$(\vec{x} \times \vec{y}) \circ \vec{z}$ (iloczyn mieszany)
map(f,v)	wartości funkcji (operacji) f dla współrzędnych wektora v
map(r,u,v)	wektory u i v przedstawione w relacji r
append(u,v)	połączenie wektorów (list) w postaci u przed v

[1)] Do stosowania tych funkcji jest wymagane wczytanie pakietu **vect**.

```
(%i1) v:[1,0,5];
(%o1) [1,0,5]
```

```
(%i2) v[3];
(%o2) 5
```

```
(%i3) [u[1],u[2],u[3]];
```
(%o3) $[u_1, u_2, u_3]$

```
(%i4) map("=",%,v);
```
(%o4) $[u_1 = 1, u_2 = 0, u_3 = 5]$

Zauważmy, że w poleceniu **map** może też wystąpić symbol relacji i dwa wektory (listy).

ZADANIA PRZYKŁADOWE

Przykład 1

Sprawdzimy działanie wybranych poleceń dla wektorów:
$\vec{x} = [x_1, x_2, x_3\ ], \vec{y} = [y_1, y_2, y_3\ ], \vec{z} = [z_1, z_2, z_3\ ]$.

Najpierw definiujemy wektory $\vec{x}$, $\vec{y}$, $\vec{z}$ oraz wersory $\vec{\imath}$, $\vec{\jmath}$, $\vec{k}$.

```
(%i1) x:[x1,x2,x3]$ y:[y1,y2,y3]$ z:[z1,z2,z3]$
      i:[1,0,0]$ j:[0,1,0]$ k:[0,0,1]$
```

Wyznaczamy sumę oraz iloczyn skalarny wektorów $\vec{x}$ i $\vec{y}$.

```
(%i7) x+y;
(%o7) [y1+x1,y2+x2,y3+x3]
```

```
(%i8) x.y;
(%o8) x3 y3+x2 y2+x1 y1
```

Obliczamy długość wektora $\vec{z}$.

```
(%i9) sqrt(z.z);
```

(%o9) $\sqrt{z3^2+z2^2+z1^2}$

Wyznaczamy iloczyn wektorowy wektorów $\vec{x}$ i $\vec{y}$.
Najpierw korzystamy z twierdzenia podstawowego o iloczynie wektorowym,

```
(%i10) matrix([i,j,k],x,y); determinant(%);
```

(%o10) $\begin{bmatrix} [1,0,0] & [0,1,0] & [0,0,1] \\ x1 & x2 & x3 \\ y1 & y2 & y3 \end{bmatrix}$

```
(%o11) [x2 y3-x3 y2,x3 y1-x1 y3,x1 y2-x2 y1]
```

a następnie stosujemy komendy dostępne po załadowaniu pakietu **vect**.

```
(%i12) load(vect)$
```

```
(%i13) express(x~y);
(%o13) [x2 y3-x3 y2,x3 y1-x1 y3,x1 y2-x2 y1]
```

Wyznaczamy iloczyn mieszany wektorów $\vec{x}$, $\vec{y}$ i $\vec{z}$.
Najpierw korzystamy z twierdzenia podstawowego o iloczynie mieszanym,

```
(%i14) matrix(x,y,z); determinant(%);
```

(%o14) $\begin{bmatrix} x1 & x2 & x3 \\ y1 & y2 & y3 \\ z1 & z2 & z3 \end{bmatrix}$

```
(%o15) x1 (y2 z3-y3 z2)-x2 (y1 z3-y3 z1)+x3 (y1 z2-y2 z1)
```

a następnie stosujemy komendy dostępne po załadowaniu pakietu **vect**.

```
(%i16) express((x~y).z);
(%o16) x1 (y2 z3-y3 z2)+x2 (y3 z1-y1 z3)+x3 (y1 z2-y2 z1)
```

Przykład 2

Sprawdzimy działanie polecenia **map** dla funkcji $f(x) = \sqrt{x^2}$ kolejno dla wektorów $\vec{u} = [\mathrm{a}, -2, b, 5]$ oraz $\vec{v} = [-1, 0, 2]$.

```
(%i1) f(x):=sqrt(x^2)$  display(f(x))$
f(x)=|x|
```

Polecenie **display** przedstawia w formie równania funkcję (zmienną) oraz jej wartość.

```
(%i3) u:[a,-2,b,5]$ v:[-1,0,2]$  display(u,v)$
u =[a,-2,b,5]
v =[-1,0,2]
(%i6) map(f,u);
(%o6) [|a|,2,|b|,5]
(%i7) map(f,v);
(%o7) [1,0,2]
(%i8) sqrt(u^2);
(%o8) [|a|,2,|b|,5]
```

Wynik z (%o6) można też uzyskać w inny sposób.

Przykład 3

Dane są punkty $A = (2, 3, 1)$, $B = (4, 6, 7)$, $C = (5,7, -2)$. Sprawdzimy, że wektory $\overrightarrow{AB}$ i $\overrightarrow{AC}$ są prostopadłe.

```
(%i1) AB:[4-2,6-3,7-1]$ AC:[5-2,7-3,-2-1]$
```

Definiowanie wektorów.

```
(%i3) AB.AC;
(%o3) 0
```

$\overrightarrow{AB} \circ \overrightarrow{AC} = 0$, więc wektory te są prostopadłe.

Przykład 4

Sprawdzimy, że punkty $A = (4, 1, 3)$, $B = (1, 3, 4)$, $C = (10, -3,1)$ leżą na jednej prostej.

```
(%i1) AB:[1-4,3-1,4-3]$ AC:[10-4,-3-1,1-3]$
(%i3) load(vect)$
```

Wczytanie pakietu **vect**.

```
(%i4) express(AB~AC);
(%o4) [0,0,0]
```

$\overrightarrow{AB} \times \overrightarrow{AC} = [0,0,0]$, więc wektory te są równoległe.

Wynika stąd, że punkty A, B, C są współliniowe.

Przykład 5

Dane są wektory: $\vec{u} = [2, 2, 1]$, $\vec{v} = [-3, 0, 4]$. Obliczymy:
a) $\vec{u} \circ \vec{v}$, b) $\vec{u} \times \vec{v}$, c) cosinus oraz sinus kąta φ między tymi wektorami.

```
(%i1) u:[2,2,1]$ v:[-3,0,4]$
(%i3) sqrt(u.u);sqrt(v.v);
(%o3) 3
(%o4) 5
```

Obliczamy długości wektorów $\vec{u}$ i $\vec{v}$.

```
(%i5) u.v;
(%o5) -2
```

Wyznaczamy iloczyn skalarny.

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

```
(%i7) uxv:express(u~v);
(%o7) [8,-11,6]
```

Wyznaczamy iloczyn wektorowy.

Korzystając z definicji iloczynu skalarnego, wyznaczamy cosinus kąta między wektorami $\vec{u}$ oraz $\vec{v}$.

```
(%i8) cos(phi)=u.v/(sqrt(u.u)*sqrt(v.v));
```

(%o8) $\cos(\phi) = -\frac{2}{15}$

Korzystając z definicji iloczynu wektorowego, wyznaczamy sinus kąta między wektorami $\vec{u}$ oraz $\vec{v}$.

```
(%i9) sin(phi)=sqrt(uxv.uxv)/(sqrt(u.u)*sqrt(v.v));
```

(%o9) $\sin(\phi) = \frac{\sqrt{221}}{15}$

Przykład 6

Obliczymy pole równoległoboku zbudowanego na wektorach
$\vec{u} = [2,3,1]$, $\vec{v} = [3,0,4]$.

```
(%i1) load(vect)$
```

```
(%i2) u:[2,3,1]$ v:[3,0,4]$
```

```
(%i4) uxv:express(u~v);
(%o4) [12,-5,-9]
```

Z interpretacji geometrycznej iloczynu wektorowego wynika, że pole równoległoboku zbudowanego na wektorach $\vec{u}$ oraz $\vec{v}$ jest równe długości iloczynu wektorowego tych wektorów.

```
(%i5) sqrt(uxv.uxv);
```

(%o5) $5\sqrt{10}$

Zatem pole równoległoboku jest równe $5\sqrt{10}$.

Przykład 7

Obliczymy objętość ostrosłupa o wierzchołkach w punktach
$A = (-1,3,1)$, $B = (2,3,4)$, $C = (1,2,-1)$, $D = (-6,3,0)$.

```
(%i1) AB:[2-(-1),3-3,4-1]$ AC:[1-(-1),2-3,-1-1]$  AD:[-6-(-1),3-3,0-1]$
```

Korzystając z interpretacji geometrycznej iloczynu mieszanego, wyznaczamy objętość równoległościanu zbudowanego na wektorach $\overrightarrow{AB}$, $\overrightarrow{AC}$, $\overrightarrow{AD}$.

```
(%i4) load(vect)$

(%i5) il_mieszany:express((AB~AC).AD);
(%o5) -12

(%i6) V=abs(il_mieszany);
(%o6) V =12
```

Objętość ostrosłupa zbudowanego na wektorach $\overrightarrow{AB}$, $\overrightarrow{AC}$, $\overrightarrow{AD}$ jest równa $\frac{1}{6}$ objętości równoległościanu zbudowanego na tych wektorach, więc wynosi 2.

Uwaga 1

Długość wektora $\vec{u}$ możemy wyznaczyć (tak jak w powyższych przykładach), wpisując sqrt($u.u$) albo np. po wczytaniu pakietu **vector_rebuild** jedynie $|u|$.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Dane są punkty $A=(4,-1,2)$, $B=(4,3,2)$, $C=(5,-2,2)$. Wyznaczyć kąt między wektorami $\overrightarrow{AB}$ i $\overrightarrow{AC}$.

Zadanie 2

Dane są wektory: $\vec{u}=[6,0,8]$, $\vec{v}=[-1,2,-2]$. Obliczyć cosinus oraz sinus kąta φ między tymi wektorami.

Zadanie 3

Korzystając z interpretacji geometrycznej iloczynu wektorowego, wyznaczyć pole trójkąta o wierzchołkach w punktach $A=(3,1,-1)$, $B=(5,-1,-2)$ oraz $C=(7,1,-4)$.

Zadanie 4

Korzystając z interpretacji geometrycznej iloczynu mieszanego sprawdzić, czy punkty $A=(2,4,5)$, $B=(3,5,7)$, $C=(1,2,1)$, $D=(3,4,5)$ leżą w jednej płaszczyźnie.

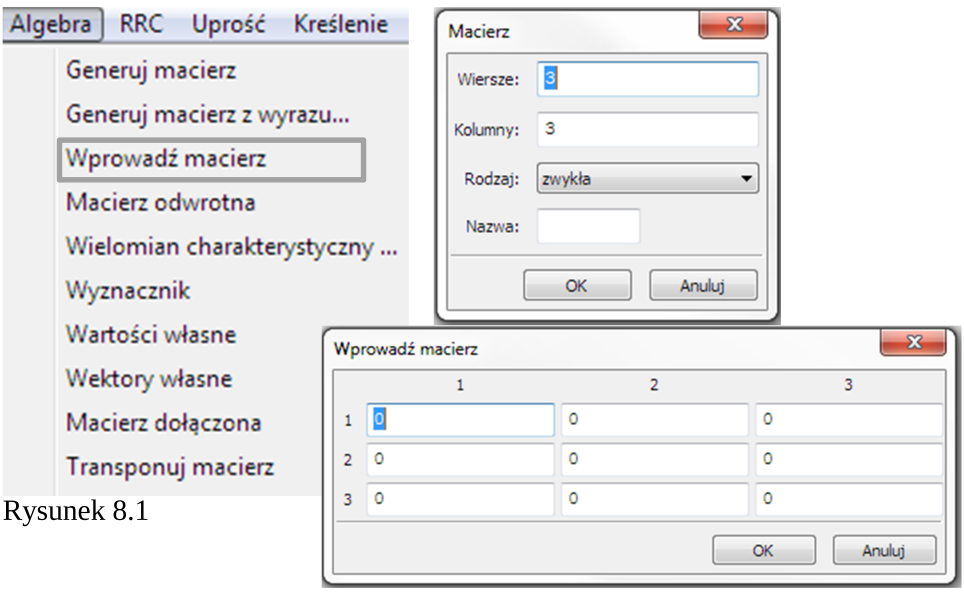
Odpowiedzi

1. $\frac{3\pi}{4}$.
2. $\cos\varphi=-\frac{11}{15}$, $\sin\varphi=\frac{2\sqrt{26}}{15}$.
3. $\sqrt{26}$.
4. Punkty leżą w jednej płaszczyźnie.

8.2. Macierze

Tabela 8.2

POLECENIA	OPIS
matrix([a,b,c],[d,e,f]**)**	$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$
ident(n)	macierz jednostkowa stopnia n
identfor(M)	macierz jednostkowa dedykowana macierzy kwadratowej M
zeromatrix(j,k)	macierz zerowa o j wierszach oraz k kolumnach
zerofor(M)	macierz zerowa dedykowana macierzy M
transpose(M)	transpozycja macierzy M
determinant(M)	wyznacznik macierzy M
invert(M)	macierz odwrotna do macierzy M
invert(M),**detout**	macierz odwrotna do A z wyłączonym jej wyznacznikiem
rank(M)	rząd macierzy M
genmatrix(h,j,k) np. h[j,k]:=…	macierz o j wierszach oraz k kolumnach, której elementy są określone za pomocą funkcji h
eigenvalues(M)	wartości własne macierzy M
eigenvectors(M)	wektory własne odpowiadające wartościom własnym macierzy M



Rysunek 8.1

Tabela 8.3

POLECENIA	OPIS
A + B, A - B, A . B	suma, różnica, iloczyn macierzy A i B [1)]
α*A	iloczyn macierzy A przez liczbę α
A^^n	n-ta potęga macierzy A
M[i,k**]**	wybranie elementu i-tego wiersza i k-tej kolumny
matrix_size(M**)**	wymiar macierzy M
submatrix(i,M**)**	skreślenie i-tego wiersza macierzy M
submatrix(M,k**)**	skreślenie k-tej kolumny macierzy M
submatrix(i,M,k**)**	skreślenie i-tego wiersza oraz k-tej kolumny macierzy M
minor(M,i,k**)**	jw.
submatrix(i_1,..,M,k_1,..**)**	skreślenie wierszy $i_1,\ldots$ oraz kolumn $k_1,\ldots$
row(M,i**)**	i-ty wiersz macierzy M
col(M,k**)**	k-ta kolumna macierzy M
addrow(M,u**)**	dołączenie do macierzy M wiersza u
addcol(M,v**)**	dołączenie do macierzy M kolumny v
rowswap(M,i,j**)**	zamiana wiersza i-tego z wierszem j-tym macierzy M
columnswap(M,i,j**)**	zamiana kolumny i-tej z kolumną j-tą macierzy M
rowop(M,i,j,k**)**	i-ty wiersz minus k razy j-ty wiersz
columnop(M,i,j,k**)**	i-ta kolumna minus k razy j-ta kolumna

[1)] Możliwe jest również wykonywanie **operacji tablicowych** na macierzach na przykład mnożenie $\mathbf{A} * \mathbf{B}$ oraz potęgowanie $\mathbf{A}^{\mathbf{n}}$.

ZADANIA PRZYKŁADOWE

Przykład 1

Sprawdzimy działanie różnych poleceń dla macierzy M zdefiniowanej poniżej w Maximie.

Większość poniższych poleceń można znaleźć w **Menu-Algebra** (patrz rysunek 8.1). Jeśli korzystamy z formularza wprowadzania macierzy, to oprócz podania wymiaru (ilości kolumn i wierszy), typu macierzy (zwykła, diagonalna, symetryczna, antysymetryczna) możemy zadeklarować też nazwę macierzy.

```
(%i1) M:matrix([1,-1,1],
               [0,4,0],
               [9,0,1]);
```

(%o1) $\begin{bmatrix} 1 & -1 & 1 \\ 0 & 4 & 0 \\ 9 & 0 & 1 \end{bmatrix}$

(%i2) transpose(M);

$$(\%o2) \begin{bmatrix} 1 & 0 & 9 \\ -1 & 4 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Algebra/Transponuj macierz

(%i3) determinant (M);

(%o3) -32

Algebra/Wyznacznik

(%i4) invert(M);

$$(\%o4) \begin{bmatrix} -\frac{1}{8} & -\frac{1}{32} & \frac{1}{8} \\ 0 & \frac{1}{4} & 0 \\ \frac{9}{8} & \frac{9}{32} & -\frac{1}{8} \end{bmatrix}$$

Algebra/Macierz odwrotna

(%i5) invert(M),detout;

$$(\%o5) -\frac{\begin{bmatrix} 4 & 1 & -4 \\ 0 & -8 & 0 \\ -36 & -9 & 4 \end{bmatrix}}{32}$$

Macierz odwrotna do M z wyłączonym wyznacznikiem. Więcej na temat macierzy odwrotnej w przykładach: 4, 5, 6.

(%i6) rank(M);

(%o6) 3

Rząd macierzy M.

(%i7) I:identfor(M)$

(%i8) display(M+I)$

$$\begin{bmatrix} 1 & -1 & 1 \\ 0 & 4 & 0 \\ 9 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 5 & 0 \\ 9 & 0 & 2 \end{bmatrix}$$

Zdefiniowanie macierzy jednostkowej I wymiaru „pasującego” do macierzy M. Polecenie **display** pozwala zaprezentować składowe oraz wynik działania.

(%i9) eigenvalues(M);

(%o9) [[-2,4],[1,2]]

Algebra/Wartości własne
Wartość -2 (jednokrotna) oraz 4 (dwukrotna).

(%i10) eigenvectors(M);

(%o10) [[[-2,4],[1,2]],[[[1,0,-3]],[[1,0,3]]]]

Algebra/ Wektory własne

Powyżej wyznaczyliśmy wektory własne odpowiadające wartościom własnym. Wartości -2 odpowiada wektor $[1,0,-3]^T$ oraz wartości 4 - wektor $[1,0,3]^T$.

Przykład 2

Utworzymy macierz losową L o 3 wierszach oraz 6 kolunach, której elementy są wybierane losowo spośród liczb całkowitych 0, 1, 2, …, 9.

```
(%i1) kill(losowanie)$
```

```
(%i2) losowanie[i,k]:=random(10)$
      L:genmatrix(losowanie,3,7);
```

$$(\%o3)\ \begin{bmatrix} 4 & 0 & 1 & 1 & 0 & 2 & 0 \\ 6 & 3 & 1 & 5 & 7 & 2 & 1 \\ 9 & 2 & 5 & 7 & 3 & 4 & 9 \end{bmatrix}$$

Definiujemy funkcję **losowanie,** korzystając z polecenia **random**, następnie na jej bazie generujemy macierz podanego wymiaru. Aby ponownie losować, trzeba wyczyścić funkcję **losowanie**.

Przykład 3

Wykonamy pewne operacje na wierszach oraz kolumnach macierzy M zdefiniowanej poniżej.

```
(%i1) M: matrix([1,-3,4],  [2,-2,5], [3,-1,6]);
```

$$(\%o1)\ \begin{bmatrix} 1 & -3 & 4 \\ 2 & -2 & 5 \\ 3 & -1 & 6 \end{bmatrix}$$

```
(%i2) M[1,3]; M[2,2];
(%o2) 4
(%o3) -2
```

Wybieramy elementy macierzy.

```
(%i4) submatrix(2,M);
```

$$(\%o4)\ \begin{bmatrix} 1 & -3 & 4 \\ 3 & -1 & 6 \end{bmatrix}$$

Skreślamy drugi wiersz.

```
(%i5) submatrix(M,3);
```

$$(\%o5)\ \begin{bmatrix} 1 & -3 \\ 2 & -2 \\ 3 & -1 \end{bmatrix}$$

Skreślamy trzecią kolumnę.

```
(%i6) submatrix(2,M,3);
```

$$(\%o6)\ \begin{bmatrix} 1 & -3 \\ 3 & -1 \end{bmatrix}$$

Skreślamy drugi wiersz oraz trzecią kolumnę.

```
(%i7) row(M,2);
```

$$(\%o7)\ \begin{bmatrix} 2 & -2 & 5 \end{bmatrix}$$

Wybieramy drugi wiersz.

```
(%i8) col(M,3);
```

$$(\%o8)\quad \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

Wybieramy trzecią kolumnę.

```
(%i9) addcol(M,[7,8,9]);
```

$$(\%o9)\quad \begin{bmatrix} 1 & -3 & 4 & 7 \\ 2 & -2 & 5 & 8 \\ 3 & -1 & 6 & 9 \end{bmatrix}$$

Dopisujemy do macierzy kolumnę.

```
(%i10) rowswap(M,2,3);
```

$$(\%o10)\quad \begin{bmatrix} 1 & -3 & 4 \\ 3 & -1 & 6 \\ 2 & -2 & 5 \end{bmatrix}$$

Zamieniamy wiersz drugi z wierszem trzecim.

```
(%i11) columnop(M,1,2,4);
```

$$(\%o11)\quad \begin{bmatrix} 13 & -3 & 4 \\ 10 & -2 & 5 \\ 7 & -1 & 6 \end{bmatrix}$$

Od pierwszej kolumny odejmujemy drugą kolumnę pomnożoną przez 4.

Poniżej zastosowanie polecenia **print** do wyświetlenia macierzy. Polecenie to pozwala łączyć tekst oraz wyniki obliczeń. Wydruk nie ma etykiety, a wszystkie składowe następują bezpośrednio po sobie (inaczej niż w poleceniu **display**).

```
(%i12) print("Macierz  M=",M, " oraz  macierz przekształcona Mp=",%o11,".")$
```

$$\text{Macierz } M=\begin{bmatrix} 1 & -3 & 4 \\ 2 & -2 & 5 \\ 3 & -1 & 6 \end{bmatrix} \text{ oraz macierz przekształcona } Mp=\begin{bmatrix} 13 & -3 & 4 \\ 10 & -2 & 5 \\ 7 & -1 & 6 \end{bmatrix}.$$

Przykład 4

Wyznaczymy macierz odwrotną do macierzy stopnia 2 zdefiniowanej symbolicznie.

```
(%i1) A: matrix( [a,b],  [c,d]);
```

$$(\%o1)\quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
(%i2) invert(A),ratsimp,detout;
```

$$(\%o2)\quad \frac{\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}{a\, d - b\, c}$$

Polecenie **ratsimp** porządkuje ułamki, natomiast **detout** pozostawia wyznacznik poza macierzą.

Przykład 5

Wyznaczymy macierz odwrotną do macierzy A zdefiniowanej poniżej, a następnie obliczymy A A^{-1} oraz A^{-1} A.

```
(%i1) A: matrix([3,4,-1], [2,5,0], [0,2,-3]);
```

$$(\%o1) \begin{bmatrix} 3 & 4 & -1 \\ 2 & 5 & 0 \\ 0 & 2 & -3 \end{bmatrix}$$

```
(%i2) determinant(A);
```

$$(\%o2)\ -25$$

Macierz A jest nieosobliwa, istnieje więc macierz do niej odwrotna.

```
(%i3) 'A^^(-1)=A^^(-1);
```

$$(\%o3)\ A^{-1} = \begin{bmatrix} \frac{3}{5} & -\frac{2}{5} & -\frac{1}{5} \\ -\frac{6}{25} & \frac{9}{25} & \frac{2}{25} \\ -\frac{4}{25} & \frac{6}{25} & -\frac{7}{25} \end{bmatrix}$$

Aby wyznaczyć macierz odwrotną, tym razem skorzystaliśmy z zapisu potęgowego ^^. Zwróćmy uwagę, że pojedynczy znak ^ pozwala na wykonanie operacji tablicowej, tzn. w tym przypadku odwrócenie każdego elementu macierzy A (oczywiście dla tej macierzy jest to niemożliwe).

Wykonamy teraz działania A A^{-1} oraz A^{-1} A.

```
(%i4) display(A.A^^(-1))$
```

$$\begin{bmatrix} 3 & 4 & -1 \\ 2 & 5 & 0 \\ 0 & 2 & -3 \end{bmatrix} . \begin{bmatrix} \frac{3}{5} & -\frac{2}{5} & -\frac{1}{5} \\ -\frac{6}{25} & \frac{9}{25} & \frac{2}{25} \\ -\frac{4}{25} & \frac{6}{25} & -\frac{7}{25} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
(%i5) display(A^^(-1).A)$
```

$$\begin{bmatrix} \frac{3}{5} & -\frac{2}{5} & -\frac{1}{5} \\ -\frac{6}{25} & \frac{9}{25} & \frac{2}{25} \\ -\frac{4}{25} & \frac{6}{25} & -\frac{7}{25} \end{bmatrix} . \begin{bmatrix} 3 & 4 & -1 \\ 2 & 5 & 0 \\ 0 & 2 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Przykład 6

Przedstawimy kolejne etapy wyznaczania macierzy odwrotnej, korzystając z macierzy dopełnień algebraicznych.

```
(%i1) A: matrix([2,-1,4], [4,2,1], [1,2,-2]);
```

$$(\%o1)\quad \begin{bmatrix} 2 & -1 & 4 \\ 4 & 2 & 1 \\ 1 & 2 & -2 \end{bmatrix}$$

1) Deklarujemy macierz A.

```
(%i2) detA:determinant(A)$
(%i3) 'det('A)=detA;
```

$$(\%o3)\quad \det(A)=3$$

2) Sprawdzamy, że **wyznacznik** det(A) macierzy A jest różny od zera.

```
(%i4) W[i,k]:=determinant(minor(A,i,k))$
(%i5) ['W[i,k]]=genmatrix(W,3,3);
```

$$(\%o5)\quad [W_{i,k}]=\begin{bmatrix} -6 & -9 & 6 \\ -6 & -8 & 5 \\ -9 & -14 & 8 \end{bmatrix}$$

3) Wyznaczamy macierz minorów $[W_{ik}]$ macierzy A. Wcześniej definiujemy funkcję W, aby jej użyć w poleceniu **genmatrix** (patrz też tabela 8.2 oraz przykład 2).

```
(%i6) WD[i,k]:=(-1)^(i+k)*W[i,k];
```

$$(\%o6)\quad WD_{i,k}:=(-1)^{i+k}\,W_{i,k}$$

```
(%i7) [((-1)^^(i+k))*'W[i,k]]
      =genmatrix(WD,3,3);
```

$$(\%o7)\quad [(-1)^{k+i}\,W_{i,k}]=\begin{bmatrix} -6 & 9 & 6 \\ 6 & -8 & -5 \\ -9 & 14 & 8 \end{bmatrix}$$

4) Wyznaczamy **macierz dopełnień algebraicznych** $[W^*_{ik}]$, gdzie $W^*_{ik} = (-1)^{i+k}\,W_{ik}$.

```
(%i8) WT:transpose(genmatrix(WD,3,3))$
(%i9) [(-1)^^(i+k)*'W[i,k]]^^T=WT;
```

$$(\%o9)\quad [(-1)^{k+i}\,W_{i,k}]^T=\begin{bmatrix} -6 & 6 & -9 \\ 9 & -8 & 14 \\ 6 & -5 & 8 \end{bmatrix}$$

5) Transponujemy macierz dopełnień algebraicznych

```
(%i10) invA:WT/detA$
```

6) Dzielimy transponowaną macierz dopełnień algebraicznych WT **przez wyznacznik** macierzy A i w ten sposób otrzymujemy macierz odwrotną.

```
(%i11) print('A^^(-1),"=",[((-1)^^(i+k))*'W[i,k]]^^T/'det('A),"=",invA)$
```

$$A^{-1}=\frac{[(-1)^{k+i}\,W_{i,k}]^T}{\det(A)}=\begin{bmatrix} -2 & 2 & -3 \\ 3 & -\frac{8}{3} & \frac{14}{3} \\ 2 & -\frac{5}{3} & \frac{8}{3} \end{bmatrix}$$

Przykład 7

Porównamy mnożenie oraz mnożenie tablicowe macierzy A oraz B zdefiniowanych symbolicznie.

```
(%i1) A: matrix([a,b], [c,d])$
      B: matrix([e,f], [g,h])$
```

Definiujemy macierz A oraz macierz B.

```
(%i3) print("A=",A,",    B=",B)$
```

$$A=\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B=\begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

Polecenie **print** pozwala na wyświetlenie obu zdefiniowanych macierzy w jednym wierszu.

```
(%i4) display(A.B)$
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} . \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} b\,g+a\,e & b\,h+a\,f \\ d\,g+c\,e & d\,h+c\,f \end{bmatrix}$$

```
(%i5) display(B.A)$
```

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix} . \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} c\,f+a\,e & d\,f+b\,e \\ c\,h+a\,g & d\,h+b\,g \end{bmatrix}$$

Zauważmy, że dla mnożenia macierzowego na ogół nie będzie zachodziła przemienność.

```
(%i6) display(A*B)$
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a\,e & b\,f \\ c\,g & d\,h \end{bmatrix}$$

```
(%i7) display(B*A)$
```

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a\,e & b\,f \\ c\,g & d\,h \end{bmatrix}$$

Mnożenie tablicowe jest oczywiście przemienne.

Przykład 8

Dla danych macierzy A, B, C, D obliczymy $\left(A^{-1}\,C + 2\,D^{T}\right)$ B.

```
(%i1) A:matrix([-1,2],[0,3]);
```

$$(\%o1) \quad \begin{bmatrix} -1 & 2 \\ 0 & 3 \end{bmatrix}$$

```
(%i2) B:matrix([1,-2,3],[2,-1,0],[1,2,3]);
```

$$(\%o2) \quad \begin{bmatrix} 1 & -2 & 3 \\ 2 & -1 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

```
(%i3) C: matrix([1,-1,-2], [0,2,3]);
```

$$(\%o3) \quad \begin{bmatrix} 1 & -1 & -2 \\ 0 & 2 & 3 \end{bmatrix}$$

```
(%i4) D: matrix([2,1], [4,3], [1,2]);
```

$$(\%o4) \quad \begin{bmatrix} 2 & 1 \\ 4 & 3 \\ 1 & 2 \end{bmatrix}$$

```
(%i5) ('A^^(-1).'C+2*'D^T).'B=(invert(A).C+2*transpose(D)).B;
```

$$(\%o5)\ (2\,D^T + A^{-1}\,.\,C)\,.\,B = \begin{bmatrix} \frac{89}{3} & -\frac{13}{3} & 27 \\ \frac{61}{3} & -\frac{2}{3} & 21 \end{bmatrix}$$

Przykład 9

Dla danych macierzy A i B wyznaczymy macierz X, jeżeli $\mathrm{A\,X} = 2\,\mathrm{B} - \mathrm{X}$.

Aby wyznaczyć z powyższego równania macierzowego macierz X, przekształcimy je najpierw tożsamościowo, otrzymując równanie $\mathrm{A\,X} + \mathrm{X} = 2\,\mathrm{B}$. Biorąc pod uwagę fakt, że mnożenie macierzy nie jest przemienne, możemy je następnie zapisać w postaci $(\mathrm{A} + \mathrm{I})\,\mathrm{X} = 2\,\mathrm{B}$, gdzie I jest macierzą jednostkową trzeciego stopnia (w przeciwnym wypadku dodawanie macierzy byłoby niewykonalne). Stąd $\mathrm{X} = 2\,(\mathrm{A} + \mathrm{I})^{-1}\,\mathrm{B}$, o ile macierz $(\mathrm{A} + \mathrm{I})^{-1}$ istnieje.

```
(%i1) A:matrix([1,0,2],[3,1,0],[2,1,-1]);
```

$$(\%o1) \begin{bmatrix} 1 & 0 & 2 \\ 3 & 1 & 0 \\ 2 & 1 & -1 \end{bmatrix}$$

```
(%i2) B:matrix([1,0],[2,1],[0,3]);
```

$$(\%o2) \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 0 & 3 \end{bmatrix}$$

```
(%i3) I:ident(3)$
```

```
(%i4) determinant(A+I);
(%o4) -2
```

Definiujemy macierz jednostkową stopnia trzeciego oraz sprawdzamy, że macierz $\mathrm{A} + \mathrm{I}$ jest nieosobliwa. Mamy $\det(\mathrm{A} + \mathrm{I}) = -2$, więc istnieje macierz odwrotna do $\mathrm{A} + \mathrm{I}$.

Zatem ostatecznie otrzymujemy:

```
(%i5) X=2*invert(A+I).B;
```

$$(\%o5)\ X = \begin{bmatrix} -4 & 10 \\ 8 & -14 \\ 5 & -10 \end{bmatrix}$$

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Dana jest macierz $\mathrm{M} = \begin{bmatrix} -1 & 1 & 3 \\ 5 & 2 & 1 \\ 3 & 1 & -2 \end{bmatrix}$. Obliczyć: a) $\det \mathrm{M}$, b) M^T, c) M^{-1}.

Zadanie 2

Dane są macierze $\mathrm{A} = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix}$, $\mathrm{B} = \begin{bmatrix} 2 & -3 \\ 4 & 5 \end{bmatrix}$. Obliczyć:

a) A B, b) B A, c) $\mathrm{A} * \mathrm{B}$, d) A^2, e) $\mathrm{A} ** 2$.

Zadanie 3

Dane są macierze:

$$\mathrm{A}=\begin{bmatrix}2 & 7\\ 1 & 3\end{bmatrix},\ \mathrm{B}=\begin{bmatrix}0 & 1 & -2\\ 1 & 2 & 3\end{bmatrix},\ \mathrm{C}=\begin{bmatrix}2 & 2 & 0\\ -1 & 1 & 1\\ 0 & 1 & -1\end{bmatrix},\ \mathrm{D}=\begin{bmatrix}-1 & 1\\ 0 & -1\\ 1 & 2\end{bmatrix}.$$

Obliczyć $\left(\mathrm{A}^{-1}\,\mathrm{B}-3\,\mathrm{D}^{\mathrm{T}}\right)\mathrm{C}$.

Zadanie 4

Dane są macierze:

$$\mathrm{A}=\begin{bmatrix}1 & 0 & 2\\ 2 & 1 & 0\\ 0 & -1 & 0\end{bmatrix},\ \mathrm{B}=\begin{bmatrix}1 & 0\\ 0 & -1\\ 2 & 1\end{bmatrix},\ \mathrm{C}=\begin{bmatrix}1 & 0\\ 2 & 1\end{bmatrix},\ \mathrm{D}=\begin{bmatrix}1 & 0 & 1\\ 0 & 3 & 0\end{bmatrix}.$$

Obliczyć $A^2\,\mathrm{B}+(2\,\mathrm{C}^{-1}\,\mathrm{D})^{\mathrm{T}}$.

Zadanie 5

Dane są macierze: $\mathrm{A}=\begin{bmatrix}-1 & -3\\ 2 & 0\end{bmatrix},\ \mathrm{B}=\begin{bmatrix}3 & -1\\ 2 & 2\end{bmatrix}.$

Wyznaczyć macierz X, jeżeli $\mathrm{X}\,\mathrm{A}^{T}-12\,\mathrm{A}^2=6\,\mathrm{B}-\mathrm{X}$.

Zadanie 6

Dane są macierze: $\mathrm{A}=\begin{bmatrix}5 & 2 & 0\\ 1 & 2 & 2\\ 2 & 2 & 4\end{bmatrix},\ \mathrm{B}=\begin{bmatrix}-1\\ 2\\ 1\end{bmatrix},\ \mathrm{C}=\begin{bmatrix}5\\ -6\\ -3\end{bmatrix}.$

Wyznaczyć macierz X, jeżeli $\mathrm{A}\,\mathrm{X}-3\,\mathrm{B}=\mathrm{X}+\mathrm{C}$.

Zadanie 7

Dane są macierze: $\mathrm{A}=\begin{bmatrix}1 & 2 & 0\\ -1 & 0 & 1\\ 0 & 1 & -1\end{bmatrix},\ \mathrm{B}=\begin{bmatrix}-1 & 0\\ 0 & -1\\ 1 & 2\end{bmatrix}.$

Wyznaczyć macierz X, jeżeli $\mathrm{A}\,\mathrm{X}=2\,\mathrm{X}+11\,\mathrm{B}$.

Zadanie 8

Dane są macierze: $A=\begin{bmatrix}-1 & 0 & 2\\ 0 & 1 & 0\\ 1 & -1 & 0\end{bmatrix},\ B=\begin{bmatrix}-1 & 0 & 1\\ 1 & 3 & 0\end{bmatrix}.$

Wyznaczyć macierz X, jeżeli $\mathrm{X}\,\mathrm{A}=10\,\mathrm{B}+3\,\mathrm{X}$.

Odpowiedzi

1. a) 15, b) $\begin{bmatrix}-1 & 5 & 3\\ 1 & 2 & 1\\ 3 & 1 & -2\end{bmatrix}$, c) $\begin{bmatrix}-\frac{1}{3} & \frac{1}{3} & -\frac{1}{3}\\ \frac{13}{15} & -\frac{7}{15} & \frac{16}{15}\\ -\frac{1}{15} & \frac{4}{15} & -\frac{7}{15}\end{bmatrix}$.

2. a) $\begin{bmatrix} 8 & -1 \\ 26 & 16 \end{bmatrix}$, b) $\begin{bmatrix} -5 & -13 \\ 23 & 29 \end{bmatrix}$, c) $\begin{bmatrix} 4 & -3 \\ 12 & 25 \end{bmatrix}$, d) $\begin{bmatrix} 7 & 7 \\ 21 & 28 \end{bmatrix}$, e) $\begin{bmatrix} 4 & 1 \\ 9 & 25 \end{bmatrix}$.

3. $\begin{bmatrix} 9 & 55 & -13 \\ -10 & -24 & 14 \end{bmatrix}$.

4. $\begin{bmatrix} 7 & 0 \\ 12 & 9 \\ 0 & -3 \end{bmatrix}$.

5. $\begin{bmatrix} -7 & -10 \\ -6 & 0 \end{bmatrix}$.

6. $\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$.

7. $\begin{bmatrix} 3 & 2 \\ -4 & 1 \\ -5 & -7 \end{bmatrix}$.

8. $\begin{bmatrix} 2 & 1 & -2 \\ -3 & -14 & -2 \end{bmatrix}$

8.3. Układy równań liniowych

Tabela 8.4

POLECENIA	OPIS
linsolve(u,X**)**	rozwiązanie układu równań liniowych, gdzie u oznacza listę równań, natomiast X - listę niewiadomych [1)]
coefmatrix(u,X**)**	macierz współczynników układu u
augcoefmatrix(u,X**)**	macierz współczynników układu u z dołączoną kolumną wyrazów wolnych pomnożoną przez -1.
echelon(M**)**	patrz uwaga 2 na końcu tego podrozdziału
triangularize(M**)**	jw.

[1)] Polecenie **linsolve** i jego składnię możemy też uzyskać, korzystając z formularza w **Menu-Równania/Rozwiąż układ równań liniowych**, gdzie podajemy ilość równań, równania oraz wymieniamy zmienne, oddzielając je przecinkami.

ZADANIA PRZYKŁADOWE

Przykład 1

Sprawdzimy działanie polecenia **linsolve** dla trzech układów równań liniowych w przypadku, gdy: a) układ posiada dokładnie jedno rozwiązanie, b) układ jest sprzeczny, c) układ ma nieskończenie wiele rozwiązań.

a) $\begin{cases} x + y - 3z = 2 \\ x + 2y - z = 1 \\ 2x + 3y - z = 3 \end{cases}$, b) $\begin{cases} x_1 + 2x_2 + 3x_3 = 2 \\ 3x_1 + 3x_2 + 10x_3 = 1 \\ x_1 + 5x_2 + 2x_3 = 3 \end{cases}$, c) $\begin{cases} x - y - z + 2t = -1 \\ x + 2y - 4z - t = 2 \\ 2x + y - 5z + t = 1 \end{cases}$.

a)

```
(%i1) linsolve([x+y-3*z=2, x+2*y-z=1, 2*x+3*y-z=3], [x,y,z]);
```

(%o1) $[x=3, y=-1, z=0]$

Układ posiada dokładnie jedno rozwiązanie postaci $x=3$, $y=-1$, $z=0$.

b)

```
(%i2) uklad:[x1+2*x2+3*x3=2, 3*x1+3*x2+10*x3=1, x1+5*x2+2*x3=3]$
```

```
(%i3) linsolve(uklad,[x1,x2,x3]);
```

(%o3) [ ]

Układ jest sprzeczny (świadczą o tym puste nawiasy [ ]).

c)

```
(%i4) u:[x-y-z+2*t=-1, x+2*y-4*z-t=2, 2*x+y-5*z+t=1]$
```

```
(%i5) linsolve(u,[x,y,z,t]);
solve: dependent equations eliminated: (3)
```

(%o5) $[x=2\ \%r2-\%r1, y=\%r2+\%r1+1, z=\%r2, t=\%r1]$

Rozważany układ równań posiada więc nieskończenie wiele rozwiązań zależnych od dwóch parametrów %r1 oraz %r2. Rozwiązania tego układu są postaci $x = 2\%r2 - \%r1$, $y = \%r2 + \%r1 + 1$, $z = \%r2$, $t = \%r1$, gdzie %r1, %r2 należą do $\mathbb{R}$. Możemy też rozwiązania te przedstawić w nieco innej formie. Najpierw zadeklarujemy dwa parametry, a następnie rozwiążemy układ względem zmiennych x i y (równocześnie podstawiając oznaczenia parametrów p).

```
(%i6) p:[z=z[0],t=t[0]];
```

(%o6) $[z=z_0, t=t_0]$

```
(%i7) r:linsolve(u,[x,y]),p;
solve: dependent equations eliminated: (3)
```

(%o7) $[x=2\ z_0-t_0, y=z_0+t_0+1]$

Połączymy teraz listy r i p, korzystając z polecenia **append**, oraz wyświetlimy postać pełnego rozwiązania, stosując polecenie **map**, oraz anonimową funkcję zadeklarowaną jako wyrażenie **lambda** do elementów podanej listy (patrz też rozdział 21).

```
(%i6) r_u:append(r,p)$
```

```
(%i7) map(lambda([i],print(r_u[i])),[1,2,3,4])$
```

$x=2\ z_0-t_0$

$y=z_0+t_0+1$

$z=z_0$

$t=t_0$

Możemy też inaczej ustalić zmienne i parametry np.

```
(%i8) p:[y=y[0],z=z[0]]$  r:linsolve(u,[x,t]),p$  r_u:append(r,p)$
      map(lambda([i],print(r_u[i])),[1,2,3,4])$
solve: dependent equations eliminated: (3)
```

$x = 3\, z_0 - y_0 + 1$

$t = -z_0 + y_0 - 1$

$y = y_0$

$z = z_0$

Przykład 2

Rozwiążemy układ równań liniowych u (zdefiniowany poniżej w Maximie), stosując twierdzenie Cramera.

```
(%i1) u:[3*x1-x2+2*x3=8,
         2*x1+x2-3*x3=-5,
         4*x1+2*x2-x3=0]$
```

Wprowadzamy wektor równań liniowych, przypisując go do zmiennej u.

```
(%i2) A:coefmatrix(u,[x1,x2,x3]);
```

(%o2) $\begin{bmatrix} 3 & -1 & 2 \\ 2 & 1 & -3 \\ 4 & 2 & -1 \end{bmatrix}$

Tworzymy macierz A współczynników przy niewiadomych.

```
(%i3) B:augcoefmatrix(u,[x1,x2,x3])$
      B:columnop(B,4,4,2);
```

(%o4) $\begin{bmatrix} 3 & -1 & 2 & 8 \\ 2 & 1 & -3 & -5 \\ 4 & 2 & -1 & 0 \end{bmatrix}$

Tworzymy macierz rozszerzoną B, która powstaje z A przez dołączenie do niej kolumny wyrazów wolnych. Stosujemy polecenie **columnop** do zmiany znaku dołączonej kolumny.

Wyznaczamy rozwiązanie układu równań, stosując wzory Cramera, i prezentujemy postać rozwiązania, podobnie jak w poprzednim przykładzie, korzystając z polecenia **map** i wyrażenia **lambda**.

```
(%i5) print('W,"=det",A,"=",W:determinant(A))$
```

$W = \det \begin{bmatrix} 3 & -1 & 2 \\ 2 & 1 & -3 \\ 4 & 2 & -1 \end{bmatrix} = 25$

```
(%i6) map(lambda([i],
                  print(  Wx[i],"=det",
                          k:submatrix(columnswap(B,i,4),4),"=",
                          wk:determinant(k),",        ",
                          x[i]=Wx[i]/'W,"=",
                          wk/W)
                               ),[1,2,3])$
```

$$Wx_1 = \det\begin{bmatrix} 8 & -1 & 2 \\ -5 & 1 & -3 \\ 0 & 2 & -1 \end{bmatrix} = 25, \qquad x_1 = \frac{Wx_1}{W} = 1$$

$$Wx_2 = \det\begin{bmatrix} 3 & 8 & 2 \\ 2 & -5 & -3 \\ 4 & 0 & -1 \end{bmatrix} = -25, \qquad x_2 = \frac{Wx_2}{W} = -1$$

$$Wx_3 = \det\begin{bmatrix} 3 & -1 & 8 \\ 2 & 1 & -5 \\ 4 & 2 & 0 \end{bmatrix} = 50, \qquad x_3 = \frac{Wx_3}{W} = 2$$

Układ posiada zatem dokładnie jedno rozwiązanie postaci

$$x_1 = 1, x_2 = -1, x_3 = 2.$$

Aby wygenerować rozwiązania **innego układu cramerowskiego** o trzech zmiennych wystarczy jedynie zmodyfikować wektor równań u.

Przykład 3

Stosując twierdzenie Kroneckera-Capellego, rozwiążemy wprowadzone poniżej w Maximie trzy układy równań liniowych, gdy: a) układ posiada dokładnie jedno rozwiązanie, b) układ jest sprzeczny, c) układ ma nieskończenie wiele rozwiązań.

a)

```
(%i1) ordergreat(x1,x2,x3)$
```

Porządkujemy zmienne (patrz tabela 4.1).

```
(%i2) u:[3*x1+x2=1, 2*x1+x2=3,4*x1+x2=-1]$
```

```
(%i3) map(lambda([i],print(u[i])),[1,2,3])$
```

$3\,x1 + x2 = 1$

$2\,x1 + x2 = 3$

$4\,x1 + x2 = -1$

Wyświetlamy postać układu równań.

```
(%i4) A:coefmatrix(u,[x1,x2]);
```

$$(\%o4)\quad \begin{bmatrix} 3 & 1 \\ 2 & 1 \\ 4 & 1 \end{bmatrix}$$

Tworzymy macierz A współczynników przy niewiadomych.

```
(%i5) B:augcoefmatrix(u,[x1,x2])$
      B:columnop(B,3,3,2);
```

$$(\%o6)\quad \begin{bmatrix} 3 & 1 & 1 \\ 2 & 1 & 3 \\ 4 & 1 & -1 \end{bmatrix}$$

Tworzymy macierz rozszerzoną B, która powstaje z A przez dołączenie do niej kolumny wyrazów wolnych (patrz przykład 2).

```
(%i7) [il_rownan,il_niewiadomych]:matrix_size(A);
(%o7) [3,2]
```

```
(%i8) rz_A:rank(A);  rz_B:rank(B);
(%o8) 2
(%o9) 2
```

Wczytujemy wymiary macierzy A do odpowiednio nazwanych zmiennych. Wyznaczamy rzędy macierzy A i B.

Stosujemy twierdzenie Kroneckera-Capellego sformułowane w postaci poniższej konstrukcji warunkowej.

```
(%i10) if rz_A#rz_B then
           print("Układ jest sprzeczny.")
       elseif rz_A=il_niewiadomych then
           print("Układ ma dokładnie jedno rozwiązanie.")
       else print("Układ ma nieskończenie wiele rozwiązań zależnych od",
                 il_niewiadomych-rz_A,"parametrów.")$
Układ ma dokładnie jedno rozwiązanie.
```

Znajdujemy to rozwiązanie, korzystając z polecenia **linsolve**.

```
(%i11) linsolve(u,[x1,x2]);
solve: dependent equations eliminated: (3)
(%o11) [x1 = -2, x2 = 7]
```

Wynik powyższy odczytujemy następująco: trzecie równanie jako zależne od pozostałych zostało wyeliminowane.
Układ posiada dokładnie jedno rozwiązanie postaci $x_1 = -2$, $x_2 = 7$.

b)
Postępujemy podobnie jak w podpunkcie a).

```
(%i1) ordergreat(x1,x2,x3,x4)$
```

```
(%i2) u:[x1+3*x2-x3+x4=2, 2*x1-x2+x3-x4=-2,4*x1-9*x2+5*x3-5*x4=3]$
```

```
(%i3) w:map(lambda([i],print(u[i])),[1,2,3])$
```

$$x1 + 3\,x2 - x3 + x4 = 2$$
$$2\,x1 - x2 + x3 - x4 = -2$$
$$4\,x1 - 9\,x2 + 5\,x3 - 5\,x4 = 3$$

```
(%i4) A:coefmatrix(u,[x1,x2,x3,x4])$
      B:augcoefmatrix(u,[x1,x2,x3,x4])$
      B:columnop(B,5,5,2);
```

$$(\%o6)\ \begin{bmatrix} 1 & 3 & -1 & 1 & 2 \\ 2 & -1 & 1 & -1 & -2 \\ 4 & -9 & 5 & -5 & 3 \end{bmatrix}$$

Tym razem wyświetlamy tylko macierz B. Macierz A jest wczytana do pamięci.
Ponieważ mamy cztery niewiadome, więc modyfikujemy znak piątej (ostatniej) kolumny.

```
(%i7) [il_rownan,il_niewiadomych]:matrix_size(A);
(%o7) [3,4]
```

```
(%i8) rz_A:rank(A);  rz_B:rank(B);
(%o8) 2
(%o9) 3
```

Poniżej odpowiedź wygenerowana przez konstrukcję warunkową (tę samą, co w podpunkcie a)).

```
(%i10) if rz_A#rz_B then
             print("Układ jest sprzeczny.")
       elseif rz_A=il_niewiadomych then
             print("Układ ma dokładnie jedno rozwiązanie.")
       else print("Układ ma nieskończenie wiele rozwiązań zależnych od",
                  il_niewiadomych-rz_A,"parametrów.")$
Układ jest sprzeczny.
```

c)
Podobnie jak w poprzednich podpunktach.

```
(%i1) ordergreat(x1,x2,x3)$
      u:[x1+x2-3*x3=2, -x1+2*x2+x3=3,-2*x1+7*x2=11]$
      map(lambda([i],print(u[i])),[1,2,3])$
```

$x1 + x2 - 3\ x3 = 2$

$-x1 + 2\ x2 + x3 = 3$

$7\ x2 - 2\ x1 = 11$

```
(%i4) A:coefmatrix(u,[x1,x2,x3])$
      B:augcoefmatrix(u,[x1,x2,x3])$B:columnop(B,4,4,2);
```

(%o6) $\begin{bmatrix} 1 & 1 & -3 & 2 \\ -1 & 2 & 1 & 3 \\ -2 & 7 & 0 & 11 \end{bmatrix}$

```
(%i7) [il_rownan,il_niewiadomych]:matrix_size(A);
(%o7) [3,3]
```

```
(%i8) rz_A:rank(A);  rz_B:rank(B);
(%o8) 2
(%o9) 2
```

```
(%i10) if rz_A#rz_B then
             print("Układ jest sprzeczny.")
       elseif rz_A=il_niewiadomych then
             print("Układ ma dokładnie jedno rozwiązanie.")
       else print("Układ ma nieskończenie wiele rozwiązań zależnych od",
                 il_niewiadomych,"-",rz_A,"parametrów.")$
Układ ma nieskończenie wiele rozwiązań zależnych od 3 - 2 parametrów.
```

Z powyższych rozważań wynika, że układ ma nieskończenie wiele rozwiązań zależnych od jednego parametru. Znajdujemy te rozwiązania. Niewiadomymi będą x_1 oraz x_2, natomiast jako parametr przyjmiemy $x_3 = t_0$, $t_0 \in \mathbb{R}$.

```
(%i11) r:linsolve(u,[x1,x2]),x3=t[0];
solve: dependent equations eliminated: (1)
```

(%o11) $[x1 = \frac{7t_0+1}{3}, x2 = \frac{2t_0+5}{3}]$

Wyświetlamy jeszcze postać rozwiązań układu. Tym razem stosujemy polecenie **display**.

```
(%i12) display(r[1],r[2],x3=t[0])$
```

$$x1 = \frac{7t_0+1}{3}$$

$$x2 = \frac{2t_0+5}{3}$$

$$x3 = t_0$$

Uwaga 1

Jeśli do rozwiązania układu równań liniowych stosujemy metodę eliminacji Gaussa, to warto zapoznać się z poleceniami **triangularize** oraz **echelon,** by móc zweryfikować własne obliczenia.

```
(%i1) u:[2*x-y-z=0,x+y+z=3,x-y+2*z=5]$
```

Wprowadzenie układu równań.

```
(%i2) solve(u, [x,y,z]);
```

(%o2) $[[x=1, y=0, z=2]]$

Rozwiązanie układu za pomocą polecenia **solve**.

```
(%i3) Au:augcoefmatrix(u,[x,y,z]);
```

(%o3) $\begin{bmatrix} 2 & -1 & -1 & 0 \\ 1 & 1 & 1 & -3 \\ 1 & -1 & 2 & -5 \end{bmatrix}$

Macierz współczynników tego układu.

```
(%i4) triangularize(Au);
```

(%o4) $\begin{bmatrix} 2 & -1 & -1 & 0 \\ 0 & 3 & 3 & -6 \\ 0 & 0 & 9 & -18 \end{bmatrix}$

Przekształcenie rozważanego układu do postaci:

$$\begin{aligned} 2x - y - z &= 0 \\ 3y + 3z - 6 &= 0 \\ 9z - 18 &= 0 \end{aligned}$$

```
(%i5) echelon(Au);
```

(%o5) $\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 1 & 1 & -2 \\ 0 & 0 & 1 & -2 \end{bmatrix}$

$$\begin{aligned} x - \frac{1}{2}y - \frac{1}{2}z &= 0 \\ y + z - 2 &= 0 \\ z - 2 &= 0 \end{aligned}$$

Uwaga 2

Polecenia **triangularize** oraz **echelon** generują macierz trójkątną górną (taką jak przy metodzie eliminacji, gdy wykonuje się operacje elementarne na wierszach). Jest jednak jedna różnica: w przypadku zastosowania **echelon** otrzymujemy macierz, której pierwsze niezerowe wyrazy w każdym wierszu są równe 1.
Taką postać łatwiej przekształcić, by uzyskać ostateczne rozwiązanie (porównaj w powyższym przykładzie (%o4) oraz (%o5)).

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Rozwiązać układy równań:

a) $\begin{cases} 3x - y + 2z = 8 \\ 2x + y - 3z = -5 \\ 4x + 2y - z = 0 \end{cases}$, b) $\begin{cases} 3x - y + z = 1 \\ 2x + y - z = -1 \\ 2x - 3y + 3z = 3 \end{cases}$, c) $\begin{cases} x_1 - 2x_2 = 3 \\ 2x_1 + x_3 = 2 \\ 3x_2 - x_3 = 1 \\ x_1 + x_2 + x_3 = 9 \end{cases}$.

Zadanie 2

Rozwiązać układy równań stosując twierdzenie Cramera:

a) $\begin{cases} 2x_1 - x_2 + 3x_3 = -3 \\ 3x_1 + 2x_2 + x_3 = 6 \\ 4x_1 - x_2 + 2x_3 = 0 \end{cases}$, b) $\begin{cases} x_1 + x_2 - x_4 = 0 \\ x_2 + x_3 = 1 \\ x_1 + x_3 = 3 \\ x_2 + 2x_4 = -1 \end{cases}$.

Zadanie 3

Rozwiązać układy równań stosując twierdzenie Kroneckera-Capellego:

a) $\begin{cases} 8x_1 + x_2 = 6 \\ 2x_1 + x_3 = 8 \\ x_2 + 2x_3 = 1 \end{cases}$, b) $\begin{cases} 3x_1 - 2x_2 + 4x_3 + 5x_4 = 3 \\ 4x_1 - 3x_2 + 3x_3 + 4x_4 = 5 \\ -x_1 - 6x_3 - 7x_4 = 1 \end{cases}$, c) $\begin{cases} 3x_1 + 2x_3 = 2 \\ x_2 + 2x_3 = 4 \\ 9x_1 - 3x_2 = 5 \end{cases}$.

Odpowiedzi

1. a) $x = 1, y = -1, z = 2$, b) $x = 0, y = \%r1 - 1, z = \%r1$, gdzie $\%r1 \in \mathbb{R}$, c) układ sprzeczny.
2. a) $x_1 = 1, x_2 = 2, x_3 = -1$, b) $x_1 = 1, x_2 = -1, x_3 = 2, x_4 = 0$.
3. a) $x_1 = 1.75, x_2 = -8, x_3 = 4.5$, b) $x_1 = -7t_0 - 6s_0 - 1, x_2 = -8t_0 - 7s_0 - 3$, $x_3 = s_0, x_4 = t_0$, gdzie $s_0, t_0 \in \mathbb{R}$, c) układ sprzeczny.

8.4. Macierze i wyznaczniki funkcyjne

Tabela 8.5

POLECENIA	OPIS
hessian(f, x**)**	macierz Hessego [1] - macierz kwadratowa, której elementy są postaci *h*[i,j]=diff(*f*(*x*), *x*[i],1,*x*[j],1), *x*=[*x*[1],*x*[2],…,*x*[n]]
wronskian([f_1,..., f_n], x**)**	macierz Wrońskiego [2] - macierz kwadratowa, której elementy są postaci *h*[i,j]=diff(*f*[j], *x*,i-1)
jacobian([f_1,..., f_m], x**)**	macierz Jacobiego [3] - macierz prostokątna, której elementy są postaci *h*[i,j]=diff(*f*[i](*x*), *x*[j]), *x*=[*x*[1],*x*[2],…,*x*[n]]
jacobian([f(x)], x**)**	$[f'_{x1}, f'_{x2}, \dots, f'_{xn}]$ (w tym szczególnym przypadku jest to gradient funkcji f)

[1] Macierz oznaczana często $H(f)(x)$, ma zastosowanie w optymalizacji.
[2] Polecenie wymaga wczytania pakietu **functs.** Wyznacznik jest stosowany m.in. w równaniach i układach równań różniczkowych do badania liniowej niezależności rozwiązań, natomiast macierz jest nazywana też macierzą fundamentalną.
[3] Stosowany do funkcji wielu zmiennych, funkcji uwikłanych, całek wielokrotnych oraz układów równań różniczkowych.

Uwaga 1

Hesjanem, **wrońskianem**, **jakobianem** nazywać będziemy wyznaczniki odpowiednio macierzy: Hessego, Wrońskiego, Jacobiego. Dla macierzy Jacobiego, oczywiście, nie zawsze istnieje jej wyznacznik.

ZADANIA PRZYKŁADOWE

Przykład 1

Przedstawimy symbolicznie macierz Hessego funkcji $f = f(x_1, x_2)$.

Najpierw wprowadzimy zależność funkcyjną.

```
(%i1) depends(f,[x[1],x[2]]);
```

(%o1) $[f(x_1, x_2)]$

Zastosujemy dwa sposoby zapisu funkcji f: skrócony w (%i2) oraz pełny w (%i3). Drugi sposób nie wymaga deklaracji **depends**.

```
(%i2) hessian(f,[x[1],x[2]]);
```

(%o2) $$\begin{bmatrix} \frac{d^2}{d x_1^2} f & \frac{d^2}{d x_1 \, d x_2} f \\ \frac{d^2}{d x_1 \, d x_2} f & \frac{d^2}{d x_2^2} f \end{bmatrix}$$

```
(%i3) hessian(f(x[1],x[2]),[x[1],x[2]]);
```

(%o3) $$\begin{bmatrix} \frac{d^2}{d x_1^2} f(x_1,x_2) & \frac{d^2}{d x_1 \, d x_2} f(x_1,x_2) \\ \frac{d^2}{d x_1 \, d x_2} f(x_1,x_2) & \frac{d^2}{d x_2^2} f(x_1,x_2) \end{bmatrix}$$

Przykład 2

Rozważmy funkcję $f(x,y,z) = x^3 - 3xy^2 - yz + z$.
a) Wyznaczymy gradient funkcji f (używając funkcji **jacobian**) oraz jej punkty stacjonarne.
b) Wyznaczymy macierz Hessego oraz zbadamy, czy jest dodatnio określona w drugim punkcie stacjonarnym.

a)

```
(%i1) f(x,y,z):=x^3-3*x*y^2-y*z+z$
```

```
(%i2) grad:jacobian([f(x,y,z)],[x,y,z]);
      solve(grad[1],[x,y,z]);
```

(%o2) $\begin{bmatrix} 3x^2-3y^2 & -z-6xy & 1-y \end{bmatrix}$

(%o3) $[[x=-1, y=1, z=6], [x=1, y=1, z=-6]]$

b)

```
(%i4) H:hessian(f(x,y,z),[x,y,z])$
```

```
(%i5) submatrix(2,3,H,2,3);
      Delta[1]=determinant(submatrix(2,3,H,2,3)),rozw[2];
```

(%o5) $\begin{bmatrix} 6x \end{bmatrix}$

(%o6) $\Delta_1 = 6$

```
(%i7) submatrix(3,H,3);
      Delta[2]=determinant(submatrix(3,H,3)),rozw[2];
```

(%o7) $\begin{bmatrix} 6x & -6y \\ -6y & -6x \end{bmatrix}$

(%o8) $\Delta_2 = -72$

```
(%i9) H;
      Delta[3]=determinant(H),rozw[2];
```

(%o9) $\begin{bmatrix} 6x & -6y & 0 \\ -6y & -6x & -1 \\ 0 & -1 & 0 \end{bmatrix}$

(%o10) $\Delta_3 = -6$

Macierz H nie jest więc dodatnio określona w punkcie (1,1,-6). Aby macierz H była dodatnio określona, wyznaczniki $\Delta_1, \Delta_2, \Delta_3$ powinny być większe od zera.

Uwaga 2

Jeśli definiujemy więcej niż jeden wyznacznik funkcyjny bądź dokonujemy zmian przy określaniu (przypisywaniu) funkcji oraz zmiennych, trzeba pamiętać o stosowaniu poleceń: **kill(**zmienna**)** oraz **reset()**.

Przykład 3

Wyznaczymy jakobiany podanych przekształceń.

a)

(%i1) `x(r,phi):=r*cos(phi);`
`y(r,phi):=r*sin(phi);`

(%o1) $x(r,\phi):=r\cos(\phi)$

(%o2) $y(r,\phi):=r\sin(\phi)$

(%i3) `jacobian([x(r,phi),y(r,phi)],[r,phi]);`

(%o3) $\begin{bmatrix} \cos(\phi) & -\sin(\phi)\, r \\ \sin(\phi) & \cos(\phi)\, r \end{bmatrix}$

(%i4) `determinant(%);`
`trigsimp(%);`

(%o4) $\sin(\phi)^2\, r+\cos(\phi)^2\, r$

(%o5) r

b)

(%i1) `x(r,phi):=a*r*cos(phi);`
`y(r,phi):=b*r*sin(phi);`

(%o1) $x(r,\phi):=a\, r\cos(\phi)$

(%o2) $y(r,\phi):=b\, r\sin(\phi)$

(%i3) `jacobian([x(r,phi),y(r,phi)],[r,phi]);`

(%o3) $\begin{bmatrix} a\cos(\phi) & -a\sin(\phi)\, r \\ b\sin(\phi) & b\cos(\phi)\, r \end{bmatrix}$

(%i4) `determinant(%);`
`trigsimp(%);`

(%o4) $a\, b\sin(\phi)^2\, r+a\, b\cos(\phi)^2\, r$

(%o5) $a\, b\, r$

c)

(%i1) `x(u,v):=u+v;`
`y(u,v):=u*v;`

(%o1) $x(u,v):=u+v$

(%o2) $y(u,v):=u\, v$

(%i3) `jacobian([x(u,v),y(u,v)],[u,v]);`

(%o3) $\begin{bmatrix} 1 & 1 \\ v & u \end{bmatrix}$

(%i4) `determinant(%);`

(%o4) $u-v$

d)

(%i1) `x(u,v):=u^2+v^2;`
`y(u,v):=2*u*v;`

(%o1) $x(u,v):=u^2+v^2$

(%o2) $y(u,v):=2\, u\, v$

(%i3) `jacobian([x(u,v),y(u,v)],[u,v]);`

(%o3) $\begin{bmatrix} 2\, u & 2\, v \\ 2\, v & 2\, u \end{bmatrix}$

(%i4) `determinant(%);`

(%o4) $4\, u^2-4\, v^2$

Przykład 4

Wyznaczymy wrońskian pary funkcji $f(t)=\sqrt{t}$, $g(t)=\frac{1}{t}$ dla $t>0$.

(%i1) `f(t):=sqrt(t)$ g(t):=1/t$`

(%i3) `load(functs)$`
`wronskian(['f(t), 'g(t)],t);`

(%o4) $\begin{bmatrix} f(t) & g(t) \\ \frac{d}{d\, t} f(t) & \frac{d}{d\, t} g(t) \end{bmatrix}$

```
(%i5) ev(%,nouns);determinant(%);
```

$$(\%o5)\quad \begin{bmatrix} \sqrt{t} & \frac{1}{t} \\ \frac{1}{2\sqrt{t}} & -\frac{1}{t^2} \end{bmatrix}$$

$$(\%o6)\quad -\frac{3}{2\,t^{3/2}}$$

Przykład 5

Sprawdzimy, że funkcje $y_1(x) = e^x$, $y_2(x) = xe^x$ są rozwiązaniami równania liniowego jednorodnego $y'' - 2y' + y = 0$, a następnie, że tworzą one układ fundamentalny na zbiorze $\mathbb{R}$, tzn. że wrońskian pary funkcji y_1, y_2 jest różny od zera dla każdego $x \in \mathbb{R}$.

```
(%i1) RLJ:'diff(y,x,2)-2*'diff(y,x)+y=0;
```

$$(\%o1)\quad \frac{d^2}{d\,x^2}\,y - 2\left(\frac{d}{d\,x}\,y\right) + y = 0$$

```
(%i2) y1:exp(x)$
      RLJ,y=y1;%,nouns;
```

$$(\%o3)\quad \frac{d^2}{d\,x^2}\,\%e^x - 2\left(\frac{d}{d\,x}\,\%e^x\right) + \%e^x = 0$$

$$(\%o4)\quad 0 = 0$$

```
(%i5) y2:x*exp(x)$
      RLJ,y=y2;%,expand,nouns;
```

$$(\%o6)\quad \frac{d^2}{d\,x^2}\left(x\,\%e^x\right) - 2\left(\frac{d}{d\,x}\left(x\,\%e^x\right)\right) + x\,\%e^x = 0$$

$$(\%o7)\quad 0 = 0$$

```
(%i8) load(functs)$
      W:wronskian([y1,y2],x);
      determinant(W),expand;
```

$$(\%o9)\quad \begin{bmatrix} \%e^x & x\,\%e^x \\ \%e^x & x\,\%e^x + \%e^x \end{bmatrix}$$

$$(\%o10)\quad \%e^{2\,x}$$

Wczytujemy pakiet **functs**.
Budujemy macierz Wrońskiego W dla funkcji y_1 i y_2.
Ponieważ $det(W) = e^{2x} > 0$ dla $x \in \mathbb{R}$, więc y_1 oraz y_2 tworzą układ fundamentalny na $\mathbb{R}$.

Rozwiązanie ogólne rozważanego równania jest więc postaci:

$$y_o = C_1\, e^x + C_2\, xe^x,\ C_1, C_2 \in \mathbb{R}.$$

Uwaga 3

W rozdziale 18 (przykład 8 oraz przykład 9) rozwiązania równań liniowych jednorodnych II rzędu o stałych współczynnikach wyznaczamy, stosując polecenie **ode2** albo, metodą „krok po kroku", budując odpowiednie równania charakterystyczne.

9. Granice i ciągłość funkcji

Tabela 9.1

POLECENIA	OPIS
limit(f(x),x,x0)	$\lim\limits_{x\to x_0} f(x)$
limit(f(x),x,x0,minus)	$\lim\limits_{x\to x_0^-} f(x)$
limit(f(x),x,x0,plus)	$\lim\limits_{x\to x_0^+} f(x)$
limit(f(x),x,inf)	$\lim\limits_{x\to\infty} f(x)$
limit(f(x),x,minf)	$\lim\limits_{x\to -\infty} f(x)$

Tabela 9.2

KOMUNIKATY PROGRAMU	OPIS
infinity	nieskończoność zespolona (w zbiorze liczb rzeczywistych granica nie istnieje)
ind	granica funkcji nie istnieje, a funkcja jest ograniczona
und	granica funkcji nie istnieje, a funkcja jest nieograniczona

ZADANIA PRZYKŁADOWE

Przykład 1

Obliczymy granice:

a) $\lim\limits_{x\to 2} \frac{\sin(2x-4)}{2-x}$, b) $\lim\limits_{x\to 5^+} \frac{\ln(x-5)}{x-3}$, c) $\lim\limits_{x\to -\infty} \frac{e^{3-x}}{2-x^5}$, d) $\lim\limits_{x\to 1} \cos\frac{1}{1-x}$.

a)

```
(%i1) 'limit(sin(2*x-4)/(2-x),x,2)=limit(sin(2*x-4)/(2-x),x,2);
```

(%o1) $\lim\limits_{x\to 2} \frac{\sin(2x-4)}{2-x} = -2$

b)

```
(%i2) 'limit(log(x-5)/(x-3),x,5,plus)=limit(log(x-5)/(x-3),x,5,plus);
```

(%o2) $\lim\limits_{x\to 5+} \frac{\log(x-5)}{x-3} = -\infty$

c)

```
(%i3) 'limit(%e^(3-x)/(2-x^5),x,minf)=limit(%e^(3-x)/(2-x^5),x,minf);
```

(%o3) $\lim\limits_{x \to -\infty} \dfrac{\%e^{3-x}}{2-x^5} = \infty$

d) Granica w tym przypadku nie istnieje, ale funkcja jest ograniczona.

```
(%i4) 'limit(cos(1/(1-x)),x,1)=limit(cos(1/(1-x)),x,1);
```

(%o4) $\lim\limits_{x \to 1} \cos\left(\dfrac{1}{1-x}\right) = ind$

Przykład 2

Obliczymy granice jednostronne funkcji f w punkcie x_0, jeżeli:

a) $f(x) = \frac{3x-1}{9-x^2}$, $x_0 = 3$; b) $f(x) = \frac{e^{\frac{1}{4-x}}}{x+2}$, $x_0 = 4$.

a) W tym przypadku granice jednostronne są różne, więc granica funkcji nie istnieje, o czym Maxima informuje nas komunikatem **infinity** (patrz tabela 9.2).

```
(%i1) 'limit((3*x-1)/(9-x^2),x,3,minus)=limit((3*x-1)/(9-x^2),x,3,minus);
'limit((3*x-1)/(9-x^2),x,3,plus)=limit((3*x-1)/(9-x^2),x,3,plus);
'limit((3*x-1)/(9-x^2),x,3)=limit((3*x-1)/(9-x^2),x,3);
```

(%o1) $\lim\limits_{x \to 3-} \dfrac{3x-1}{9-x^2} = \infty$

(%o2) $\lim\limits_{x \to 3+} \dfrac{3x-1}{9-x^2} = -\infty$

(%o3) $\lim\limits_{x \to 3} \dfrac{3x-1}{9-x^2} = infinity$

b) W tym przypadku granice jednostronne są różne, więc granica funkcji nie istnieje, o czym Maxima informuje nas komunikatem **und** (patrz tabela 9.2).

```
(%i4) 'limit((%e^(1/(4-x))/(x+2)),x,4,minus)=limit((%e^(1/(4-x))/(x+2)),x,4,minus);
'limit((%e^(1/(4-x))/(x+2)),x,4,plus)=limit((%e^(1/(4-x))/(x+2)),x,4,plus);
'limit((%e^(1/(4-x))/(x+2)),x,4)=limit((%e^(1/(4-x))/(x+2)),x,4);
```

(%o4) $\lim\limits_{x \to 4-} \dfrac{\%e^{\frac{1}{4-x}}}{x+2} = \infty$

(%o5) $\lim\limits_{x \to 4+} \dfrac{\%e^{\frac{1}{4-x}}}{x+2} = 0$

(%o6) $\lim\limits_{x \to 4} \dfrac{\%e^{\frac{1}{4-x}}}{x+2} = und$

Przykład 3

Zbadamy ciągłość funkcji f w punkcie x_0 i naszkicujemy jej wykres, jeżeli:

a) $f(x) = \begin{cases} x + e^{1+\frac{1}{x}} & \text{dla} \quad -1 \le x < 0, \\ (1-2x)\operatorname{arctg} x & \text{dla} \quad 0 \le x \le 1, \end{cases} \qquad x_0 = 0;$

b) $f(x) = \begin{cases} \ln x & \text{dla} \quad 0 < x \le 1, \\ \frac{x-\sqrt{x-1}}{x-1} & \text{dla} \quad 1 < x \le 3, \end{cases} \qquad x_0 = 1;$

c) $f(x) = \begin{cases} x^2 - x & \text{dla} \quad 0 \le x \le 2, \\ e^{\frac{1}{4-x^2}} & \text{dla} \quad 2 < x \le 5, \end{cases} \qquad x_0 = 2.$

Będziemy definiować funkcje sklejane, więc skorzystamy z konstrukcji warunkowej **if** (patrz tabela 21.1).

a)

```
(%i1) f(x):=if (-1<=x and x<0 ) then x+exp(1+1/x)
              elseif (0<=x and x<=1) then (1-2*x)*atan(x)$
```

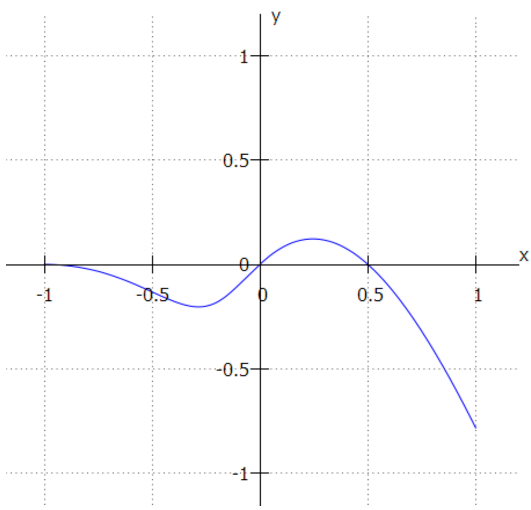
Dziedziną rozważanej funkcji jest przedział $[-1,1]$. Obliczymy wartość funkcji w punkcie $x_0 = 0$, a następnie jej granicę lewostronną w tym punkcie (granica prawostronna jest równa wartości funkcji).

```
(%i2) 'f(0)=f(0);
(%o2) f(0)=0
```

```
(%i3) 'limit(x+exp(1+1/x) ,x,0,minus)=limit(x+exp(1+1/x) ,x,0,minus);
```

(%o3) $\lim\limits_{x \to 0-} \%e^{1/x+1} + x = 0$

```
plot2d( f, [x,-1.2,1.2],[y,-1.2,1.2],
        [axes, true], [box, false],
        grid2d, [yx_ratio, 1],
        [axes, solid],
        [xtics, -1, .5,1],
        [ytics, -1,0.5,1],
        [label, ["x", 1.2, 0.05],
        ["y", 0.05, 1.2]]
        )$
```



Funkcja f jest więc ciągła w punkcie $x_0 = 0$.

b)

```
(%i1) f(x):=if (0<x and x<=1 ) then log(x)
              elseif (1<x and x<=3) then (x-sqrt(x-1))/(x-1)$
```

Dziedziną rozważanej funkcji jest przedział $(0,3]$. Aby zbadać jej ciągłość w punkcie $x_0 = 1$, obliczymy jej wartość, a następnie granicę prawostronną w tym punkcie (granica lewostronna jest równa wartości funkcji).

```
(%i2) 'f(1)=f(1);
```

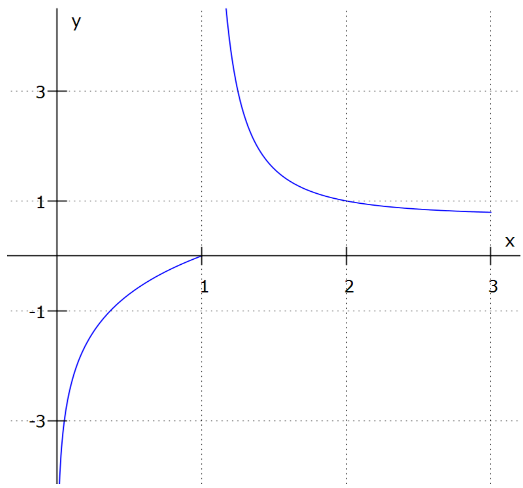
(%o2) $f(1)=0$

```
(%i3) 'limit((x-sqrt(x-1))/(x-1),x,1,plus)=limit((x-sqrt(x-1))/(x-1),x,1,plus);
```

(%o3) $\lim\limits_{x\to 1+} \dfrac{x-\sqrt{x-1}}{x-1} = \infty$

Ponieważ granica prawostronna w punkcie $x_0 = 1$ jest niewłaściwa, to funkcja f jest nieciągła w tym punkcie. Poniżej ilustracja graficzna tego faktu.

```
plot2d(f,[x,-0.2,3.2],[y,-4,4],
       [axes,true],[box,false],
       grid2d, [yx_ratio, 1],
       [axes, solid],
       [xtics, 1, 1,4]],
       [ytics, -3,2,3]],
       [label,["x", 3.1, 0.3],
              ["y", 0.1, 4]]
       )$
```



c) W tym przypadku, dziedziną funkcji f jest przedział $[0,5]$.

```
(%i1) f(x):=if (0<=x and x<=2 ) then x^2-x
              elseif (2<x and x<=5) then exp(1/(4-x^2))$
```

```
(%i2) 'f(2)=f(2);
```

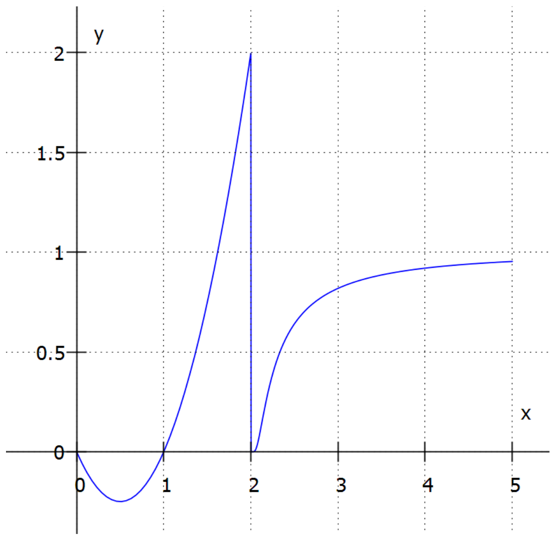
(%o2) $f(2)=2$

```
(%i3) 'limit(exp(1/(4-x^2)),x,2,plus)=limit(exp(1/(4-x^2)),x,2,plus);
```

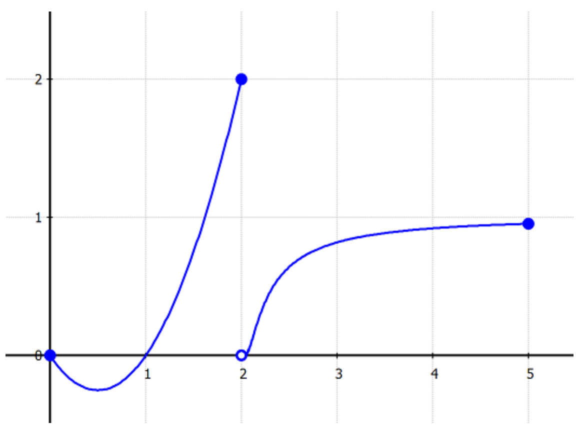
(%o3) $\lim\limits_{x\to 2+} \%e^{\frac{1}{4-x^2}} = 0$

Z porównania wartości funkcji f oraz granicy prawostronnej w punkcie $x_0 = 2$ wynika, że funkcja f jest nieciągła w tym punkcie.
Poniżej dwa rysunki przedstawiające wykresy funkcji f. Na pierwszym z nich pojawia się pionowy odcinek w punkcie nieciągłości. Do narysowania drugiego wykresu funkcji f zostały zastosowane opcje pakietu **draw**, co pozwoliło na bardziej precyzyjny szkic.

```
plot2d(f,[x,-1,5.5],[y,-0.5,2.25],
       [axes,true],[box,false],
       grid2d, [yx_ratio, 1],
       [axes, solid],
       [xtics, 0, 1,5],
       [ytics, 0,0.5,2],
       [label,["x", 5.1, 0.2],
              ["y", 0.2, 2.1]]
       )$
```




```
load(draw)$
draw2d(grid=true,
    xrange=[-0.5,5.5],
    yrange=[-0.5,2.5],
    xaxis_width =2,
    yaxis_width =2,
    xaxis_type =solid,
    yaxis_type =solid,
    xaxis=true,yaxis=true,
    xtics_axis=true,
    xtics={1,2,3,4,5},
    ytics_axis=true,
    ytics={0,1,2},
    line_width=2,
explicit(x^2-x,x,0,2),
explicit(exp(1/(4-x^2)),x,2,5),
    point_size=1.5,
    point_type=filled_circle,
points([0,2,5,2],[f(0),f(2),f(5),0]),
    point_size=0.7,color=white,
points([2],[0])  )$
```



ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Obliczyć granice, o ile istnieją, lub stwierdzić brak istnienia granic:

a) $\lim\limits_{x\to 4} \frac{\operatorname{tg}(4x-16)}{x-4}$, b) $\lim\limits_{x\to 3^-} \frac{\log(3-x)}{1-x}$, c) $\lim\limits_{x\to -\infty} \frac{e^{1-2x}}{x^3-1}$, d) $\lim\limits_{x\to 1} \frac{x-2}{x-x^2}$, e) $\lim\limits_{x\to 1^-} \frac{x-2}{x-x^2}$,

f) $\lim\limits_{x\to 1^+} \frac{x-2}{x-x^2}$, g) $\lim\limits_{x\to -2} \frac{e^{\frac{1}{x^2+2x}}}{x+1}$, h) $\lim\limits_{x\to -2^-} \frac{e^{\frac{1}{x^2+2x}}}{x+1}$, i) $\lim\limits_{x\to -2^+} \frac{e^{\frac{1}{x^2+2x}}}{x+1}$, j) $\lim\limits_{x\to 2} \sin\frac{x}{x-2}$.

Zadanie 2

Zbadać ciągłość funkcji f w punkcie x_0 i naszkicować jej wykres, jeżeli:

a) $f(x) = \begin{cases} (1-3x)\arcsin x & \text{dla} \quad -0.5 \le x \le 0, \\ e^{1-\frac{x+1}{x}} & \text{dla} \quad 0 < x \le 0.5, \end{cases} \qquad x_0 = 0;$

b) $f(x) = \begin{cases} \frac{4}{\pi}\operatorname{arctg} x & \text{dla} \quad -2 \le x \le 1, \\ \frac{\sqrt{2x-1}-\sqrt{3x-2}}{x-1} & \text{dla} \quad 1 < x \le 2, \end{cases} \qquad x_0 = 1;$

c) $f(x) = \begin{cases} e^{\frac{1}{x-3}} & \text{dla} \quad 0 \le x < 3, \\ x - x^2 & \text{dla} \quad 3 \le x \le 4, \end{cases} \qquad x_0 = 3.$

Odpowiedzi

1. a) 4, b) ∞, c) $-\infty$, d) nie istnieje, e) $-\infty$, f) ∞, g) nie istnieje, h) $-\infty$, i) 0, j) nie istnieje.
2. a) funkcja ciągła; b) funkcja nieciągła, $f(1) = 1$, $\lim\limits_{x\to 1^+} f(x) = -0.5$;

   c) funkcja nieciągła, $f(3) = -6$, $\lim\limits_{x\to 3^-} f(x) = 0$.

10. Rachunek różniczkowy funkcji jednej zmiennej

Pochodną funkcji f w punkcie x_0 oznaczamy symbolem

$$f'(x_0) \quad \text{lub} \quad \frac{df}{dx}(x_0) \quad \text{lub} \quad \frac{d}{dx}f(x)\Big|_{x=x_0}.$$

Pochodną rzędu n-tego funkcji f w punkcie x_0 oznaczamy symbolem

$$f^{(n)}(x_0) \quad \text{lub} \quad \frac{d^n f}{dx^n}(x_0) \quad \text{lub} \quad \frac{d^n}{dx^n}f(x)\Big|_{x=x_0},$$

w szczególności pochodną rzędu drugiego funkcji f w punkcie x_0 oznaczamy również symbolem $f''(x_0)$.

Tabela 10.1

POLECENIA	OPIS
diff(f(x),x**)**	pochodna rzędu pierwszego funkcji f
diff(f(x),x,n**)**	pochodna rzędu n-tego funkcji f, gdzie n =1,2,3,…
at('diff(f(x),x),x=x0**)**	$f'(x_0)$

ZADANIA PRZYKŁADOWE

Przykład 1

Obliczymy pochodną funkcji $f(x) = 2x^3 + \sqrt{4x-7} - 10\,\mathrm{arctg}\,x$ w punkcie $x = 2$.

Polecenie **at**('diff(),…) pozwala nam na symboliczne przedstawienie obliczanej pochodnej, natomiast polecenie **ev** na stopniowe przekształcanie.

```
(%i1) f(x):=2*x^3+sqrt(4*x-7)-10*atan(x)$
(%i2) p:at('diff(f(x),x),x=2)$
(%i3) p=ev(p,diff);
```

(%o3) $\frac{d}{d\,x}\left(-10\,\mathrm{atan}(x)+\sqrt{4\,x-7}+2\,x^3\right)\Big|_{x=2} = -\frac{10}{x^2+1}+\frac{2}{\sqrt{4\,x-7}}+6\,x^2\Big|_{x=2}$

```
(%i4) p=ev(p,nouns);
```

(%o4) $\frac{d}{d\,x}\left(-10\,\mathrm{atan}(x)+\sqrt{4\,x-7}+2\,x^3\right)\Big|_{x=2} = 24$

Wykonanie wszystkich instrukcji dla p (patrz tabela 2.2).

Przykład 2

Obliczymy drugą pochodną funkcji $f(x) = \frac{1}{x^2+1} + \operatorname{arctg} x + \ln(x^2+1)$ i zapiszemy ją w postaci iloczynowej.

(%i1) f(x):=1/(x^2+1)+atan(x)+log(x^2+1);

(%o1) $f(x):=\frac{1}{x^2+1}+\operatorname{atan}(x)+\log(x^2+1)$

(%i2) diff(f(x),x,2);

(%o2) $\frac{2}{x^2+1}-\frac{4x^2}{(x^2+1)^2}-\frac{2x}{(x^2+1)^2}-\frac{2}{(x^2+1)^2}+\frac{8x^2}{(x^2+1)^3}$

(%i3) factor(%);

(%o3) $-\frac{2(x-1)x(x^2+2x-1)}{(x^2+1)^3}$

Przykład 3

Obliczymy pochodną funkcji $f(x) = e^{\alpha x}\cos^2(\beta x)$, a następnie zdefiniujemy funkcję $p1$ jako pochodną funkcji f przedstawioną w postaci iloczynowej. Na końcu wyznaczymy wartość $p1(0)$.

(%i1) f(x):=%e^(alpha*x)*cos(beta*x)^2$

(%i2) diff(f(x),x,1);

(%o2) $\alpha \,\%e^{\alpha x}\cos(\beta x)^2-2\beta\,\%e^{\alpha x}\cos(\beta x)\sin(\beta x)$

(%i3) p1(x):=''(factor(diff(f(x),x,1)));

(%o3) $p1(x):=-\%e^{\alpha x}\cos(\beta x)(2\beta\sin(\beta x)-\alpha\cos(\beta x))$

(%i4) p1(0);

(%o4) α

Przykład 4

Wyznaczymy ekstrema lokalne i przedziały monotoniczności funkcji
$f(x) = \frac{e^x}{x^2-1}$.

Podobnie jak w poprzednim przykładzie, najpierw definiujemy funkcję f, a następnie jej pochodną $p1$ przekształconą do postaci iloczynowej.

(%i1) f(x):=%e^x/(x^2-1)$

(%i2) p1(x):=''(factor(diff(f(x),x,1)));

(%o2) $p1(x):=\frac{(x^2-2x-1)\%e^x}{(x-1)^2(x+1)^2}$

Dziedziną funkcji f oraz jej pochodnej jest zbiór $\mathbb{R} \setminus \{-1,1\}$.
Korzystając z instrukcji **solve** wyznaczamy punkty stacjonarne tzn. miejsca zerowe pochodnej.

```
(%i3) solve(p1(x)=0);
(%o3) [x=1-sqrt(2),x=sqrt(2)+1]
```

Sprawdzamy teraz, gdzie pochodna jest ujemna, a gdzie dodatnia. Zauważamy, że jednym z czynników pochodnej jest e^x. Wyrażenie to, dla dowolnego x z dziedziny, przyjmuje wartości dodatnie. Badając zatem znak pochodnej, możemy ten czynnik pominąć. Do rozwiązywania nierówności wymiernych wykorzystujemy pakiet **solve_rat_ineq** oraz instrukcję o tej samej nazwie.

```
(%i4) load(solve_rat_ineq)$
```

```
(%i5) solve_rat_ineq((x^2-2*x-1)/(((x-1)^2)*(x+1)^2)<0);
(%o5) [[x>1-sqrt(2),x<1],[x>1,x<sqrt(2)+1]]
```

```
(%i6) solve_rat_ineq((x^2-2*x-1)/(((x-1)^2)*(x+1)^2)>0);
(%o6) [[x<-1],[x>-1,x<1-sqrt(2)],[x>sqrt(2)+1]]
```

Powyższe wyniki odczytujemy następująco:

$$f'(x) < 0 \Leftrightarrow x \in (1-\sqrt{2},1) \cup (1,1+\sqrt{2})$$

$$f'(x) > 0 \Leftrightarrow x \in (-\infty,-1) \cup (-1,1-\sqrt{2}) \cup (1+\sqrt{2},+\infty)$$

Zatem funkcja f jest malejąca na przedziałach: $\left(1-\sqrt{2},1\right)$, $(1,1+\sqrt{2})$ oraz rosnąca na przedziałach: $(-\infty,-1)$, $\left(-1,1-\sqrt{2}\right)$, $(1+\sqrt{2},+\infty)$.
Ponadto w punkcie $x = 1-\sqrt{2}$ funkcja f osiąga maksimum lokalne, a w punkcie $x = 1+\sqrt{2}$ minimum lokalne.

Przykład 5

Wyznaczymy ekstrema lokalne i przedziały monotoniczności funkcji $f(x) = \frac{x}{\ln x - 1}$.

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

```
(%i2) p1(x):=''(factor(diff(f(x),x,1)));
```

$$\text{(\%o2) } p1(x) := \frac{\log(x)-2}{(\log(x)-1)^2}$$

Dziedziną funkcji f oraz jej pochodnej jest zbiór $(0,e) \cup (e,+\infty)$.
Wyznaczamy teraz punkty stacjonarne.

```
(%i3) solve(p1(x)=0);
(%o3) [x=%e^2]
```

Funkcja $p1$ ma więc jedno miejsce zerowe, którego wartość przybliżona wynosi

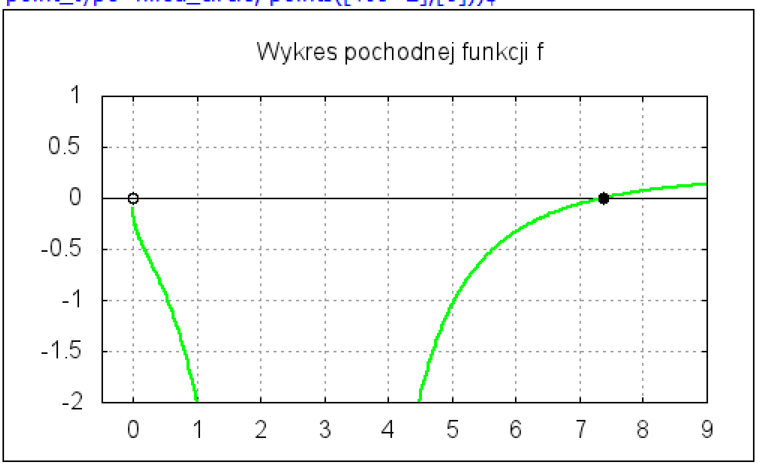
```
(%i4) %,numer;
(%o4) [x = 7.38905609893065]
```

W celu zbadania warunku wystarczającego istnienia ekstremum lokalnego funkcji f wykorzystujemy wykres pochodnej.

```
(%i5) load(draw)$
```

```
(%i6) wxdraw2d(
      xrange=[-0.5,9], yrange=[-2,1], grid=true,
      title="Wykres pochodnej funkcji f",
      xaxis=true, xaxis_type=solid,
      color=green, line_width=2,
      explicit(p1(x),x,0,9),
      color=black, point_size=1.25,
      point_type=circle, points([0],[0]),
      point_type=filled_circle, points([%e^2],[0]))$
```

(%t6)

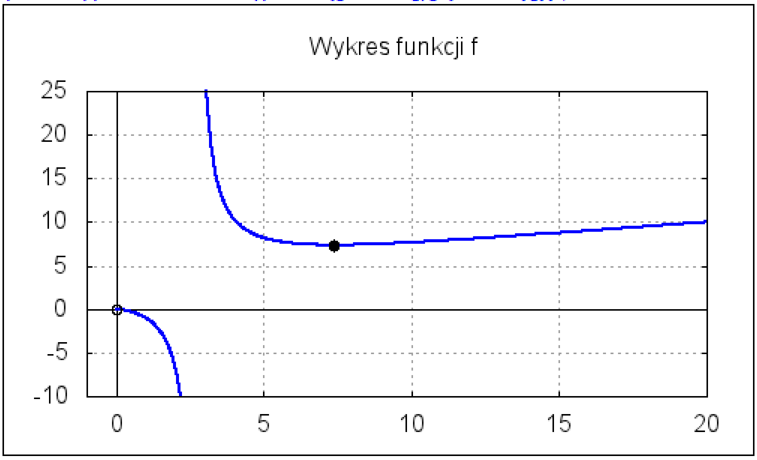


W sąsiedztwie punktu $x = e^2$ pochodna zmienia znak z "$-$" na "$+$", więc funkcja f posiada minimum lokalne w tym punkcie.
Z powyższego wykresu odczytujemy też, że pochodna przyjmuje wartości ujemne dla $x \in (0, e) \cup (e, e^2)$ oraz wartości dodatnie dla $x \in (e^2, \infty)$. Wynika stąd, że funkcja f jest malejąca na przedziałach: $(0, e)$, (e, e^2) oraz jest rosnąca na przedziale (e^2, ∞).

Rysujemy teraz wykres funkcji f.

```
(%i7) wxdraw2d(
      xrange=[-1,20], yrange=[-10,25], grid=true,
      title="Wykres funkcji f",
      xaxis=true, xaxis_type=solid, yaxis=true, yaxis_type=solid,
      color=blue, line_width=2,
      explicit(f(x),x,0,20),
      color=black, point_size=1.25,
      point_type=circle,points([0],[0]),
      point_type=filled_circle,points([%e^2],[f(%e^2)]))$
```

(%t7)



Przykład 6

Wyznaczymy ekstrema lokalne funkcji

$$f(x) = 4(x^2 + 1)\operatorname{arctg} x - \pi x^2 - 4x,$$

wykorzystując drugi warunek wystarczający istnienia ekstremum lokalnego.

Najpierw definiujemy funkcję f oraz jej pochodną $p1$ (dziedziną obu funkcji jest zbiór liczb rzeczywistych), a następnie wyznaczamy punkty stacjonarne funkcji f.

```
(%i1) f(x):=4*(x^2+1)*atan(x)-%pi*x^2-4*x$
      p1(x):=''(diff(f(x),x,1));
```

(%o2) $\mathrm{p1}(x) := 8\, x\, \mathrm{atan}(x) - 2\,\pi\, x$

```
(%i3) solve([p1(x)=0], [x]);
```

(%o3) $[x = 1, x = 0]$

Obliczamy teraz drugą pochodną funkcji f (oznaczamy ją przez $p2$) oraz badamy jej znak w każdym z punktów stacjonarnych.

```
(%i4) p2(x):=''(diff(f(x),x,2));
```

(%o4) $\mathrm{p2}(x) := 8\,\mathrm{atan}(x) + \dfrac{8\,x}{x^2+1} - 2\,\pi$

```
(%i5) 'p2(0)=p2(0);
      'p2(1)=p2(1);
(%o5) p2(0) = -2 π
(%o6) p2(1) = 4
```

Na podstawie drugiego warunku wystarczającego istnienia ekstremum lokalnego wnioskujemy, że funkcja f osiąga w punkcie $x = 0$ maksimum lokalne, gdyż $p2(0) < 0$, natomiast w punkcie $x = 1$ osiąga minimum lokalne, gdyż $p2(1) > 0$. Obliczamy wartości funkcji f w tych punktach.

```
(%i7) fmax(0)=f(0); fmin(1)=f(1);
(%o7) fmax(0) = 0
(%o8) fmin(1) = π - 4
```

Przykład 7

Wyznaczymy najmniejszą i największą wartość funkcji $f(x) = x\sqrt{32 - x^2}$ na przedziale $\left[-4\sqrt{2}, 3\right]$.

Przedział $\left[-4\sqrt{2}, 3\right]$ jest zawarty w dziedzinie naturalnej funkcji f.

```
(%i1) f(x):=x*sqrt(32-x^2);
      p1(x):=''(factor(diff(f(x),x,1)));
```

(%o1) $f(x) := x\sqrt{32 - x^2}$

(%o2) $p1(x) := -\dfrac{2(x-4)(x+4)}{\sqrt{32 - x^2}}$

```
(%i3) solve([p1(x)=0], [x]);
(%o3) [x = -4, x = 4]
```

Otrzymujemy dwa rozwiązania równania $p1(x) = 0$. Liczba -4 należy do przedziału $\left(-4\sqrt{2}, 3\right]$ (dziedziny funkcji $p1$), natomiast liczba 4 nie należy do tego przedziału. Sprawdzenie takie (w trudniejszych rachunkowo przypadkach) można wykonać, badając wartość logiczną zdań (jak poniżej).

```
(%i4) (x>-4*sqrt(2) and x<=3),x=-4,pred;
(%o4) true
```

```
(%i5) (x>-4*sqrt(2) and x<=3),x=4,pred;
(%o5) false
```

Aby znaleźć najmniejszą i największą wartość funkcji f na podanym przedziale, obliczamy wartość tej funkcji w punkcie $x = -4$ oraz na krańcach przedziału tzn. w punktach $x = -4\sqrt{2}$ oraz $x = 3$.
Pomocniczo szkicujemy wykres funkcji f.

```
(%i6) 'f(-4)=f(-4);
      'f(-4*sqrt(2))=f(-4*sqrt(2));
      'f(3)=f(3);
```

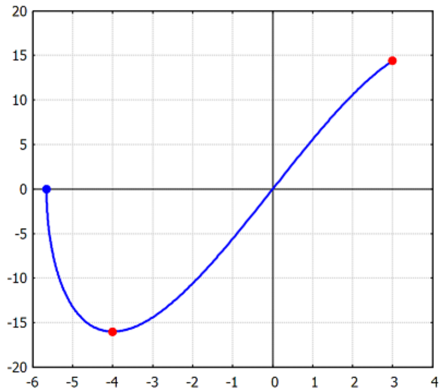
(%o6) $f(-4)=-16$

(%o7) $f(-2^{5/2})=0$

(%o8) $f(3)=3\sqrt{23}$

```
(%i1) load(draw)$
```

```
(%i2) draw2d(xrange=[-6,4],
         yrange=[-20,20],
         grid=true,
         title=sconcat("f(x)=", f(x)),
         xaxis=true,xtics=1,
         xaxis_type=solid,
         yaxis=true,
         yaxis_type=solid,
         color=blue,line_width=2,
      explicit(f(x),x,-4*sqrt(2),3),
         point_type=7,
      points([-4*sqrt(2)],[0]),
         point_type=7,color=red,
      points([-4,3],[f(-4),f(3)]))$
```



Wartość najmniejsza $f_{min}(-4) = -16$, wartość największa $f_{max}(3) = 3\sqrt{23}$.

Przykład 8

Wyznaczymy punkty przegięcia oraz przedziały wklęsłości i wypukłości funkcji $f(x) = \frac{x-1}{x^2-2x+2}$.

```
(%i1) f(x):=(x-1)/(x^2-2*x+2);
```

(%o1) $f(x):=\frac{x-1}{x^2-2x+2}$

Zauważamy, że mianownik funkcji f nie ma pierwiastków rzeczywistych. Dziedziną funkcji f oraz jej pochodnych jest więc zbiór liczb rzeczywistych.

```
(%i2) solve(denom(f(x)));
```

(%o2) $[x=1-\%i, x=\%i+1]$

denom(w) wybiera mianownik wyrażenia wymiernego w.

Przedstawiamy teraz drugą pochodną funkcji f w postaci iloczynowej, a następnie znajdujemy jej miejsca zerowe.

```
(%i3) p2(x):=''(factor(diff(f(x),x,2)));
```

(%o3) $p2(x):=\frac{2(x-1)(x^2-2x-2)}{(x^2-2x+2)^3}$

```
(%i4) solve(p2(x)=0);
(%o4) [x=1-√3,x=√3+1,x=1]
```

Punkty $x = 1 - \sqrt{3}$, $x = 1$, $= 1 + \sqrt{3}$ mogą być punktami przegięcia.
Korzystamy teraz z warunku wystarczającego istnienia punktu przegięcia.
Do rozwiązania odpowiednich nierówności wymiernych będziemy potrzebować pakietu **solve_rat_ineq**.

```
(%i5) load(solve_rat_ineq)$
```

```
(%i6) n1:p2(x)>0;
      solve_rat_ineq(n1);
```

$$(\%o6)\ \frac{2(x-1)(x^2-2x-2)}{(x^2-2x+2)^3} > 0$$

$$(\%o7)\ [[x>1-\sqrt{3},x<1],[x>\sqrt{3}+1]]$$

```
(%i8) n2:p2(x)<0;
      solve_rat_ineq(n2);
```

$$(\%o8)\ \frac{2(x-1)(x^2-2x-2)}{(x^2-2x+2)^3} < 0$$

$$(\%o9)\ [[x<1-\sqrt{3}],[x>1,x<\sqrt{3}+1]]$$

Powyższe wyniki odczytujemy następująco:

$$f''(x) > 0 \Leftrightarrow x \in (1-\sqrt{3},1) \cup (\,1+\ \sqrt{3},+\infty)$$
$$f''(x) < 0 \Leftrightarrow x \in (-\infty,1-\sqrt{3}) \cup (1{,}1+\ \sqrt{3})$$

Zatem funkcja f jest wypukła na przedziałach: $(1-\sqrt{3},1)$, $(\,1+\ \sqrt{3},+\infty)$ oraz wklęsła na przedziałach: $\left(-\infty,1-\sqrt{3}\right)$, $(1{,}1+\ \sqrt{3})$.
Ponieważ druga pochodna zmienia znak w sąsiedztwie punktów $x = 1 - \sqrt{3}$, $x = 1$, $x = 1 + \sqrt{3}$, punkty te są punktami przegięcia funkcji f.

Obliczamy wartości funkcji w punktach przegięcia (symbolicznie i numerycznie). Do uproszczenia wyników przydatne będzie polecenie **radcan**.

```
(%i10) f(1-sqrt(3)),radcan;
       f(1);
       f(1+sqrt(3)),radcan;
```

$$(\%o10)\ -\frac{\sqrt{3}}{4}$$

$$(\%o11)\ 0$$

$$(\%o12)\ \frac{\sqrt{3}}{4}$$

```
(%i13) 'f(1-sqrt(3))=f(1-sqrt(3.0));
       'f(1)=f(1);
       'f(1+sqrt(3))=f(1+sqrt(3.0));
```

$$(\%o13)\ f(1-\sqrt{3}) = -0.43301270189221$$

$$(\%o14)\ f(1)=0$$

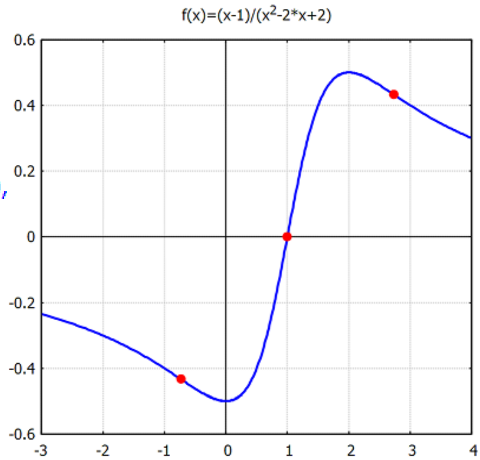
$$(\%o15)\ f(\sqrt{3}+1)=0.43301270189221$$

Zanim narysujemy wykres funkcji f, deklarujemy wektory odciętych i rzędnych punktów przegięcia.

```
(%i16) p:[1-sqrt(3),1,1+sqrt(3)]$
       fp:map(f,p)$
```

```
(%i18) load(draw)$
```

```
(%i19) draw2d(
           xrange=[-3,4],
           yrange=[-0.6,0.6],
           grid=true,
           title=sconcat("f(x)=",f(x)),
           xaxis=true,xtics=1,
           xaxis_type=solid,
           yaxis=true,
           yaxis_type=solid,
           line_width=2,
       explicit(f(x),x,-3,4),
           point_type=7,
           color=red,
       points(p,fp))$
```



Przykład 9

Wyznaczymy równania asymptot wykresu funkcji $f(x) = 6\operatorname{arctg} x - 5x$.

```
(%i1) f(x):=6*atan(x)-5*x$
```

Dziedziną funkcji f jest zbiór $\mathbb{R}$, zatem funkcja ta nie posiada asymptoty pionowej. Obliczymy teraz granice funkcji w $+\infty$ oraz w $-\infty$.

```
(%i2) limit(f(x), x, inf);
(%o2) -∞
```

```
(%i3) limit(f(x), x, minf);
(%o3) ∞
```

Z powyższego wnioskujemy, że funkcja f nie posiada asymptoty poziomej. Szukamy następnie asymptoty ukośnej w $+\infty$ wśród funkcji postaci $y = a_1 x + b_1$, gdzie współczynniki a_1, b_1 znajdujemy, obliczając odpowiednie granice. Analogicznie postępujemy w $-\infty$.

```
(%i4) a1:limit(f(x)/x, x, inf);
(%o4) -5
```

```
(%i5) b1:limit(f(x)-a1*x, x, inf);
(%o5) 3 π
```

```
(%i6) y=a1*x+b1;
(%o6) y =3 π -5 x
```

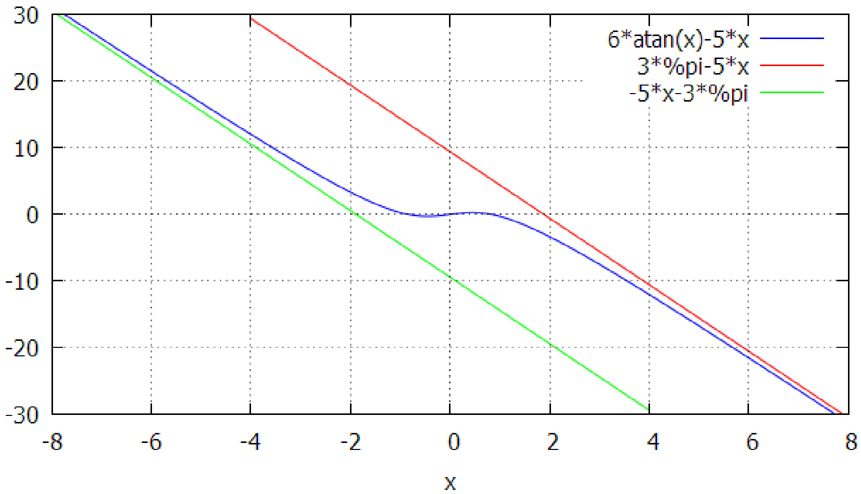
```
(%i7) a2:limit(f(x)/x, x, minf);
(%o7) -5
```

```
(%i8) b2:limit(f(x)-a2*x, x, minf);
(%o8) -3 π
```

```
(%i9) y=a2*x+b2;
(%o9) y = -5 x -3 π
```

Narysujemy teraz wykres funkcji f wraz z wykresami jej asymptot.

```
(%i10) plot2d([ f(x), a1*x+b1, a2*x+b2], [x,-8,8], [y,-30,30], [yx_ratio, 0.5])$
```



Przykład 10

Zbadamy przebieg zmienności funkcji $f(x) = \frac{2x^3+6x^2+6x+1}{(x+1)^2}$ i naszkicujemy jej wykres.

```
(%i1) f(x):=(2*x^3+6*x^2+6*x+1)/(x+1)^2$
```

Dziedziną funkcji f jest zbiór $D = (-\infty, -1) \cup (-1, \infty)$. Obliczamy granice funkcji f na krańcach dziedziny.

```
(%i2) limit(f(x), x, inf);
(%o2) ∞
```

Funkcja f nie posiada asymptoty poziomej.

```
(%i3) limit(f(x), x, minf);
(%o3) -∞
```

```
(%i4) limit(f(x), x, -1, plus);
(%o4) -∞
```

```
(%i5) limit(f(x), x, -1, minus);
(%o5) -∞
```

Wynika stąd, że funkcja f posiada asymptotę pionową obustronną o równaniu $x = -1$.

Badamy dalej istnienie asymptot ukośnych funkcji f, obliczając granice:

```
(%i6) a:limit(f(x)/x, x, inf)$
      b:limit(f(x)-a*x, x, inf)$
      y=a*x+b;
(%o8) y =2 x +2
```

```
(%i9) a:limit(f(x)/x, x, minf)$
      b:limit(f(x)-2*x, x, minf)$
      y=a*x+b;
(%o11) y =2 x +2
```

Prosta $y = 2x + 2$ jest więc asymptotą ukośną funkcji f w $+\infty$ oraz w $-\infty$.

W kolejnym etapach:

a) badamy pierwszą pochodną funkcji f, aby wyznaczyć ekstrema lokalne oraz przedziały monotoniczności,

b) badamy drugą pochodną, aby sprawdzić istnienie punktów przegięcia oraz wyznaczyć przedziały wypukłości i wklęsłości.

Pochodne, podobnie jak w poprzednich przykładach, definiujemy w postaci iloczynowej oraz wczytujemy pakiet do rozwiązywania nierówności wymiernych.

```
(%i12) p1(x):=''(factor(diff(f(x),x,1)));
```

(%o12) $p1(x):=\frac{2(x+2)(x^2+x+1)}{(x+1)^3}$

Dziedzina pierwszej oraz drugiej pochodnej jest taka sama jak dziedzina funkcji.

```
(%i13) p2(x):=''(factor(diff(f(x),x,2)));
```

(%o13) $p2(x):=-\frac{6}{(x+1)^4}$

```
(%i14) load(solve_rat_ineq)$
```

a)

```
(%i15) realroots(p1(x)=0);
```

(%o15) $[x=-2]$

Rozwiązań tego równania poszukujemy w zbiorze liczb rzeczywistych.

```
(%i16) p1(x)>0;
       solve_rat_ineq(p1(x)>0);
```

(%o16) $\frac{2(x+2)(x^2+x+1)}{(x+1)^3}>0$

(%o17) $[[x<-2],[x>-1]]$

```
(%i18) p1(x)<0;
       solve_rat_ineq(p1(x)<0);
```

(%o18) $\frac{2(x+2)(x^2+x+1)}{(x+1)^3}<0$

(%o19) $[[x>-2,x<-1]]$

```
(%i20) 'f(-2)=f(-2);
```

(%o20) $f(-2)=-3$

Na podstawie powyższych wyników wnioskujemy, że funkcja f rośnie na przedziałach: $(-\infty,-2)$, $(-1,+\infty)$ oraz maleje na przedziale $(-2,-1)$. Ponadto w punkcie $x=-2$ funkcja f osiąga maksimum lokalne i $f(-2)=-3$.

b)

Druga pochodna w całej swojej dziedzinie przyjmuje wartości ujemne.

```
(%i21) p2(x)<0;p2(x)<0,pred;
```

(%o21) $-\frac{6}{(x+1)^4}<0$

(%o22) true

Znajdujemy miejsca zerowe funkcji f, a następnie szkicujemy jej wykres. Interesują nas, oczywiście, tylko pierwiastki rzeczywiste.
Stosując polecenie **solve**, otrzymujemy wszystkie pierwiastki (również zespolone) w formie symbolicznej. Dodanie **radcan** i **expand** pozwala przekształcić je do prostszej postaci. Następnie wybieramy pierwiastek rzeczywisty i przybliżamy go z dokładnością 3 cyfr znaczących.

```
(%i23) solve(f(x)=0),radcan,expand;
```

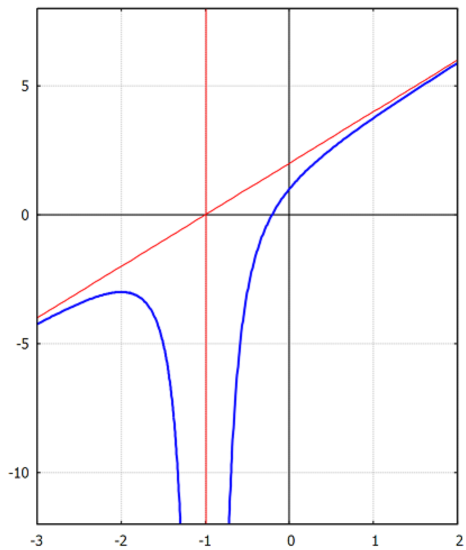
(%o23) $[x=-\frac{\sqrt{3}\,\%i}{2^{4/3}}-\frac{1}{2^{4/3}}-1, x=\frac{\sqrt{3}\,\%i}{2^{4/3}}-\frac{1}{2^{4/3}}-1, x=\frac{1}{2^{1/3}}-1]$

```
(%i24) %[3],bfloat,fpprec=3;
```

(%o24) $x=-2.06\mathrm{b}{-1}$

```
(%i25) load(draw)$
```

```
(%i26) draw2d(
        dimensions=[450,600],
        xrange=[-3,2],
        yrange=[-12,8],
        grid=true,
        xaxis=true,
        xaxis_type=solid,
        yaxis=true,
        yaxis_type=solid,
        line_width=2,
    explicit(f(x),x,-4,3),
        line_width=1,
        color=red,
    explicit(2*x+2,x,-4,3),
    implicit(x=-1,x,-4,3,y,-12,8)
            )$
```



Przykład 11

Pudełko w kształcie prostopadłościanu powinno mieć pojemność 36 cm^3. Jego dno ma być prostokątem, którego stosunek długości boków będzie wynosić 2. Dno i ściany pudełka mają być pokryte wewnątrz farbą. Jakie powinny być wymiary pudełka, by zużyć jak najmniej farby?

Z warunków zadania mamy, że $x \cdot 2x \cdot z = 36$, gdzie x, $2x$, z są długościami boków prostopadłościanu. Z powyższej równości wyznaczamy zmienną z i wstawiamy ją do wzoru na pole powierzchni P ścian i dna pudełka.

```
(%i1) x*(2*x)*z=36$
```

```
(%i2) %/(2*x^2);
```

(%o2) $z=\frac{18}{x^2}$

```
(%i3) x*(2*x)+2*x*z+2*(2*x)*z,%$
```

```
(%i4) P(x):=''(%);
```

(%o4) $P(x):=2x^2+\frac{108}{x}$

Dziedziną funkcji P (zgodnie z warunkami zadania) jest zbiór $D = (0, \infty)$.

Wyznaczamy miejsca zerowe pochodnej funkcji P.

```
(%i5) realroots(diff(P(x),x,1));
(%o5) [x=3]
```

Wynika stąd, że jest tylko jeden punkt stacjonarny $x = 3$. Badamy znak drugiej pochodnej funkcji P w tym punkcie.

```
(%i6) diff(P(x),x,2);%,x=3;
```

(%o6) $\frac{216}{x^3}+4$

(%o7) 12

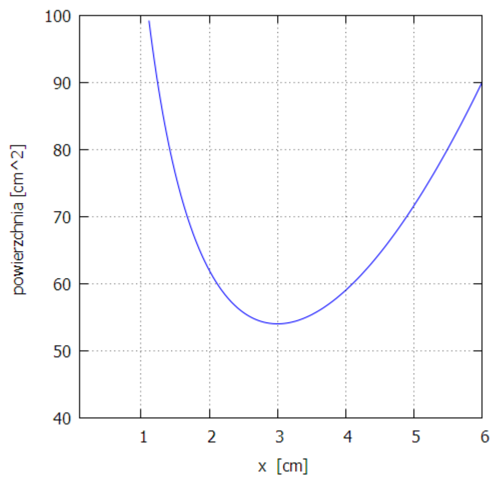
Ponieważ $P''(3) = 12 > 0$, więc funkcja P posiada minimum lokalne w tym punkcie. Wyznaczamy teraz wymiary pudełka dla $x = 3$.

```
(%i8) z=18/x^2,x=3;
(%o8) z=2
```

Zatem najmniej farby wykorzystamy do pokrycia takiego pudełka, którego dno ma wymiary 3 cm i 6 cm, natomiast wysokość jest równa 2 cm.

Poniższy rysunek przedstawia zależność funkcyjną powierzchni do pomalowania (funkcja P) od długości krótszego boku podstawy prostopadłościanu (zmienna x). Rysunek ten pozwala odczytać przybliżoną wartość powierzchni minimalnej.

```
(%i9) plot2d(P(x),
        [x,.1,6],[y,40,100],
        grid2d, [yx_ratio, 1],
        [xtics,0, 1,6],
        [ytics, 0,10,100],
        [xlabel, " x  [cm]"],
        [ylabel,
        "powierzchnia [cm^2] "]
           )$
```



Przykład 12

Dana jest funkcja popytu $D(p) = pe^{-p^2+1}$ w zależności od ceny p produktu. Jak zmieni się popyt na rozważany produkt, jeżeli cenę $p_0 = 2$ *zł* podniesiemy o 1%?

Korzystamy z interpretacji ekonomicznej elastyczności funkcji popytu określonej wzorem

$$E_p D(P) = \frac{p}{D(p)} D'(p).$$

Elastyczność funkcji popytu określa, w przybliżeniu, o ile procent zmieni się popyt, gdy cena p wzrośnie o 1%. Aby wyznaczyć elastyczność, definiujemy funkcję popytu oraz jej pochodną.

```
(%i1) D(p):=p*%e^(-p^2+1);
```

(%o1) $D(p) := p\,\%e^{-p^2+1}$

```
(%i2) D1(p):=''(factor(diff(D(p),p,1)));
```

(%o2) $D1(p) := -(2\,p^2-1)\,\%e^{1-p^2}$

```
(%i3) E(p):=(p/D(p))*(D1(p));
```

(%o3) $E(p) := \frac{p}{D(p)} D1(p)$

```
(%i4) p0:2$ 'E('p0)=E(p0);
```

(%o5) $E(p0) = -7$

Wynik oznacza, że przy cenie produktu 2 *zł* jej podwyższenie o 1% spowoduje spadek popytu o 7% (a jej obniżenie o 1% spowoduje wzrost popytu o 7%).

Przykład 13

Znając funkcję kosztów całkowitych

$$K(x) = 0.1x^3 - 0.12x^2 + 70x + 20\,000,$$

gdzie x jest liczbą wytworzonych jednostek, wyznaczymy wielkość produkcji, przy której jednostkowe koszty produkcji są minimalne. Ile wynoszą te koszty?

Funkcja kosztów przeciętnych (jednostkowych) jest określona równaniem

$$k(x) = \frac{K(x)}{x}$$

W celu wyznaczenia wielkości produkcji, przy której jednostkowe koszty produkcji są minimalne, definiujemy funkcję kosztów przeciętnych i obliczamy jej pierwszą oraz drugą pochodną.

```
(%i1) k(x):=(1/10*x^3-12/100*x^2+70*x+20000)/x;
      k1(x):=''(radcan(diff(k(x),x)));
      k2(x):=''(radcan(diff(k(x),x,2)));
```

(%o1) $k(x) := \frac{\frac{1}{10}x^3 - \frac{12}{100}x^2 + 70\,x + 20000}{x}$

(%o2) $k1(x) := \frac{5\,x^3 - 3\,x^2 - 500000}{25\,x^2}$

(%o3) $k2(x) := \frac{x^3 + 200000}{5\,x^3}$

Następnie wyznaczamy punkty stacjonarne. Interesują nas jedynie pierwiastki rzeczywiste (a dokładniej dodatnie) pochodnej $k1$ i ich przybliżenia całkowite.

```
(%i4) realroots(k1(x)),bfloat,fpprec=3;
(%o4) [x =4.66b1]
```

```
(%i5) round(rhs(%[1]));
(%o5) 47
```

```
(%i6) k2(%),numer;
(%o6) 0.58527108636814
```

Druga pochodna w punkcie stacjonarnym ma wartość dodatnią, więc jest to minimum.

Wnioskujemy stąd, że jednostkowe koszty produkcji są najmniejsze, gdy wielkość produkcji wynosi 47 sztuk. Liczymy teraz jednostkowe koszty produkcji przy wielkości produkcji równej 47 sztuk.

```
(%i7) k(47.0);
(%o7) 710.791914893617
```

Zatem najmniejsze koszty przeciętne (optimum techniczne) wynoszą $k_{min} = 711$ *zł* i są osiągane dla wielkości produkcji $x = 47$ sztuk.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Obliczyć: a) $f'(x)$, gdy $f(x) = \frac{\ln x}{\ln x+1}$, b) $f^{(4)}(x)$, gdy $f(x) = e^{3x}(4x^2 + 1)$.

Zadanie 2

Obliczyć:

a) $\frac{d^2}{dx^2} f(x) \Big|_{x=0}$, gdy $f(x) = e^{-x^2} \operatorname{arctg}\left(\frac{x}{3}\right)$, b) $f^{(3)}\left(\frac{\pi}{3}\right)$, gdy $f(x) = \sin^3(2x)$.

Zadanie 3

Wyznaczyć punkty stacjonarne funkcji:

a) $f(x) = \frac{(x+1)^3}{x^2-2x}$, b) $f(x) = \frac{e^{2x}}{x-x^2}$,

c) $f(x) = \ln^3 x - 3\ln x$, d) $f(x) = \sqrt{1-x^2} - 0.5 \arcsin x$.

Zadanie 4

Wyznaczyć ekstrema lokalne (o ile istnieją) oraz przedziały monotoniczności funkcji:

a) $f(x) = \frac{3-x^2}{x^3}$, b) $f(x) = \frac{e^{-2x}}{x-1}$,

c) $f(x) = \frac{x}{1-\ln x}$, d) $f(x) = 2\operatorname{arctg}^2 x - \pi \operatorname{arctg} x + 1$.

Zadanie 5

Wyznaczyć ekstrema lokalne funkcji wykorzystując drugi warunek wystarczający istnienia ekstremum lokalnego:

a) $f(x) = 3x^5 - 25x^3 + 60x$, b) $f(x) = 3\arcsin x - 4x\sqrt{1-x^2}$.

Zadanie 6

Wyznaczyć najmniejszą i największą wartość funkcji na wskazanym przedziale:

a) $f(x) = (x^2 - 2)\sqrt{4 - x^2}$, $x \in [-1,1]$,

b) $f(x) = 3\ln^2 x - \ln^3 x$, $x \in [e, e^3]$,

c) $f(x) = (2\cos x - 1)^2$, $x \in \left[\frac{\pi}{6}, \pi\right]$,

d) $f(x) = x - 2\sin x + 2\cos x - 2\cos^2 x$, $x \in \left[0, \frac{\pi}{4}\right]$.

Zadanie 7

Wyznaczyć punkty przegięcia (o ile istnieją) oraz przedziały wklęsłości i wypukłości funkcji:

a) $f(x) = \frac{(x-1)^3}{x^2-2x}$, b) $f(x) = (1 - 2x^2)\, e^{2x}$,

c) $f(x) = x - \operatorname{arctg} x$, d) $f(x) = x(2\ln x - \ln^2 x)$.

Zadanie 8

Wyznaczyć równania asymptot wykresu funkcji:

a) $f(x) = \frac{x^2}{x-3}$, b) $f(x) = \frac{e^{-x}}{x^2-4}$, c) $f(x) = 4\operatorname{arctg} x - x$,

d) $f(x) = 1 - \frac{\ln x}{x}$, e) $f(x) = x - e^{\frac{1}{x}}$.

Zadanie 9

Zbadać przebieg zmienności funkcji:

a) $f(x) = \frac{x^3}{1-x^2}$, b) $f(x) = xe^{\frac{1}{x}}$, c) $f(x) = (2x^2 - x + 1)e^{-x}$,

d) $f(x) = \frac{x}{\ln x}$, e) $f(x) = \ln^2 x - \ln x - 1$, f) $f(x) = (x-5)\sqrt{x^2+1}$.

Zadanie 10

Który z prostokątów o polu $S = 100$ cm^2 ma najmniejszy obwód?

Zadanie 11

Liczbę 200 przedstawiono w postaci sumy dwóch składników tak, że iloraz sumy kwadratów tych składników przez iloczyn tych składników jest najmniejszy. Ile wynosi wartość tego ilorazu?

Zadanie 12

Z kawałka drutu o długości 90 cm należy zbudować model prostopadłościanu o podstawie kwadratowej i największej objętości. Jakie powinny być wymiary tego prostopadłościanu?

Zadanie 13

Pole powierzchni całkowitej prostopadłościanu o podstawie kwadratowej jest równe 12. Jakie powinny być wymiary tego prostopadłościanu, by jego objętość była największa?

Zadanie 14

Funkcja popytu na pomidory w zależności od ceny p jest określona wzorem $D(p) = pe^{-3p^2}$. Jak zmieni się popyt na pomidory, jeżeli cenę $p_0 = 5\ zł$ podniesiemy o 1% ?

Zadanie 15

Funkcja popytu na gruszki w zależności od ceny p jest określona wzorem $D(p) = 100e^{\sqrt{100-p^2}}$. Jak zmieni się popyt na gruszki, jeżeli cenę $p_0 = 6\ zł$ podniesiemy o 1% ?

Zadanie 16

Dla poniższych funkcji kosztów całkowitych K wyznaczyć wielkość produkcji, przy której jednostkowe koszty produkcji są minimalne i podać, ile wynoszą te koszty.

a) $K(x) = 0.02x^3 - 5x^2 + 30x + 180\,000$,
b) $K(x) = 0.005x^3 - 6x^2 + 3000x + 52\,000$.

Odpowiedzi

1. a) $\frac{1}{x(\ln x+1)^2}$, b) $27(12x^2 + 32x + 19)\, e^{3x}$.
2. a) 0, b) -48.
3. a) $x = -1$, $x = 3 - \sqrt{7}$, $x = 3 + \sqrt{7}$; b) $x = 1 - \frac{\sqrt{2}}{2}$, $x = 1 + \frac{\sqrt{2}}{2}$; c) $x = e^{-1}$, $x = e$; d) $x = -0.5$.
4. a) funkcja jest rosnąca na przedziałach: $(-\infty, -3)$, $(3, \infty)$ oraz malejąca na przedziałach: $(-3, 0)$, $(0, 3)$, maksimum lokalne w punkcie $x = -3$, minimum lokalne w punkcie $x = 3$; b) funkcja jest rosnąca na przedziale $(-\infty, 0.5)$ oraz malejąca na przedziałach: $(0.5, 1)$, $(1, \infty)$, maksimum lokalne w punkcie $x = 0.5$; c) funkcja jest rosnąca na przedziałach: $(0, e)$, (e, e^2) oraz malejąca na przedziale (e^2, ∞), maksimum lokalne w punkcie $x = e^2$; d) funkcja jest malejąca na przedziale $(-\infty, 1)$ oraz rosnąca na przedziale $(1, \infty)$, minimum lokalne w punkcie $x = 1$.
5. a) maksimum lokalne w puncie $x = -2$ oraz w punkcie $x = 1$, minimum lokalne w punkcie $x = -1$ oraz w punkcie $x = 2$; b) maksimum lokalne w puncie $x = -\frac{\sqrt{2}}{4}$ oraz minimum lokalne w punkcie $x = \frac{\sqrt{2}}{4}$.
6. a) $\min\limits_{[-1,1]} f(x) = f(0) = -4$, $\max\limits_{[-1,1]} f(x) = f(-1) = f(1) = -\sqrt{3}$;
   b) $\min\limits_{[e,e^3]} f(x) = f(1) = 0$, $\max\limits_{[e,e^3]} f(x) = f(e^2) = 4$; c) $\min\limits_{\left[\frac{\pi}{6},\pi\right]} f(x) = f\left(\frac{\pi}{3}\right) = 0$, $\max\limits_{\left[\frac{\pi}{6},\pi\right]} f(x) = f(\pi) = 9$; d) $\min\limits_{\left[0,\frac{\pi}{4}\right]} f(x) = f\left(\frac{\pi}{6}\right) \approx -0.24$, $\max\limits_{\left[0,\frac{\pi}{4}\right]} f(x) = f(0) = 0$.
7. a) funkcja jest wklęsła na przedziałach: $(-\infty, 0)$, $(1,2)$ oraz wypukła na przedziałach: $(0, 1)$, $(2, \infty)$, $x = 1$ – punkt przegięcia; b) funkcja jest wklęsła na przedziałach: $(-\infty, -2)$, $(0, \infty)$ oraz wypukła na przedziale $(-2, 0)$, $x = -2$, $x = 0$ – punkty przegięcia;

c) funkcja jest wklęsła na przedziale $(-\infty, 0)$ oraz wypukła na przedziale $(0, \ \infty)$, $x = 0$ – punkt przegięcia; d) funkcja jest wypukła na przedziale $(0, 1)$ oraz wklęsła na przedziale $(1, \ \infty)$, $x = 1$ – punkt przegięcia.

8. a) $x = 3$ asymptota pionowa obustronna, $y = x + 3$ asymptota ukośna przy $x \to \pm\infty$; b) $x = -2$ asymptota pionowa obustronna, $x = 2$ asymptota pionowa obustronna, $y = 0$ asymptota pozioma przy $x \to \infty$; c) $y = -x + 2\pi$ asymptota ukośna przy $x \to \infty$, $y = -x - 2\pi$ asymptota ukośna przy $x \to -\infty$; d) $x = 0$ asymptota pionowa prawostronna, $y = 1$ asymptota pozioma przy $x \to \infty$; e) $x = 0$ asymptota pionowa prawostronna, $y = x - 1$ asymptota ukośna przy $x \to \pm\infty$.

9. a) $x = -1$ asymptota pionowa obustronna, $x = 1$ asymptota pionowa obustronna, $y = -x$ asymptota ukośna przy $x \to \pm\infty$, funkcja jest malejąca na przedziałach: $(-\infty, -2)$, $(2, \infty)$ oraz rosnąca na przedziałach: $(-2, -1)$, $(-1, 1)$, $(1, \ 2)$, minimum lokalne w punkcie $x = -2$, maksimum lokalne w punkcie $x = 2$, funkcja jest wypukła na przedziałach: $(-\infty, -1)$, $(0, 1)$ oraz wklęsła na przedziałach: $(-1, 0)$, $(1, \ \infty)$, $x = 0$ – punkt przegięcia; b) $x = 0$ asymptota pionowa prawostronna, $y = x + 1$ asymptota ukośna przy $x \to \pm\infty$, funkcja jest rosnąca na przedziałach: $(-\infty, 0)$, $(1, \infty)$ oraz malejąca na przedziale $(0, 1)$, minimum lokalne w punkcie $x = 1$, funkcja jest wklęsła na przedziale $(-\infty, 0)$ oraz wypukła na przedziale $(0, \infty)$; c) $y = 0$ asymptota pozioma przy $x \to \infty$, funkcja jest malejąca na przedziałach: $(-\infty, 0.5)$, $(2, \infty)$ oraz rosnąca na przedziale $(0.5, 2)$, minimum lokalne w punkcie $x = 0.5$, maksimum lokalne w punkcie $x = 2$, funkcja jest wypukła na przedziałach: $(-\infty, 1)$, $(3.5, \infty)$ oraz wklęsła na przedziale $(1, 3.5)$, $x = 1$, $x = 3.5$ – punkty przegięcia; d) $x = 1$ asymptota pionowa obustronna, funkcja jest malejąca na przedziałach: $(0,1)$, $(1, e)$ oraz rosnąca na przedziale (e, ∞), minimum lokalne w punkcie $x = e$, funkcja jest wklęsła na przedziałach: $(0, 1)$, (e^2, ∞) oraz wypukła na przedziale $(1, e^2)$, $x = e^2$ – punkt przegięcia; e) $x = 0$ asymptota pionowa prawostronna, funkcja jest malejąca na przedziale $\left(0, \sqrt{e}\right)$ oraz rosnąca na przedziale $\left(\sqrt{e}, \ \infty\right)$, minimum lokalne w punkcie $x = \sqrt{e}$, funkcja jest wypukła na przedziale $\left(0, \sqrt{e^3}\right)$ oraz wklęsła na przedziale $\left(\sqrt{e^3}, \infty\right)$, $x = \sqrt{e^3}$ – punkt przegięcia; f) funkcja jest rosnąca na przedziałach: $\left(-\infty, \frac{5-\sqrt{17}}{4}\right)$, $\left(\frac{5+\sqrt{17}}{4}, \infty\right)$ oraz malejąca na przedziale $\left(\frac{5-\sqrt{17}}{4}, \frac{5+\sqrt{17}}{4}\right)$, maksimum lokalne w punkcie $x = \frac{5-\sqrt{17}}{4}$, minimum lokalne w punkcie $x = \frac{5+\sqrt{17}}{4}$, funkcja jest wklęsła na przedziale $(-\infty, 1)$ oraz wypukła na przedziale $(1, \ \infty)$, $x = 1$ – punkt przegięcia.

10. Kwadrat o długości boku 10 cm.

11. 2.

12. 7.5 cm, 7.5 cm, 7.5 cm.

13. $\sqrt{2}$, $\sqrt{2}$, $\sqrt{2}$.

14. Popyt spadnie o 14%.

15. Popyt spadnie o 4.5%.

16. a) $x = 219$ sztuk, $k_{min} = 716\ zł$; b) $x = 614$ sztuk, $k_{min} = 1286\ zł$.

11. Ciągi liczbowe i funkcyjne

Tabela 11.1

POLECENIA	OPIS
a[n]:=	definicja ciągu a_n
a .. b	ciąg liczb od a do b wypisanych co 1 [2)]
a .. r .. b	ciąg liczb od a do b wypisanych co r [2)]
arithmetic(a,r,n)	n-ty wyraz ciągu arytmetycznego o pierwszym wyrazie a i różnicy r [1)]
arithsum(a,r,n)	suma ciągu arytmetycznego n-wyrazowego o pierwszym wyrazie a i różnicy r [1)]
geometric(a,q,n)	n-ty wyraz ciągu geometrycznego o pierwszym wyrazie a i ilorazie q [1)]
geosum(a,q,n)	suma ciągu geometrycznego n-wyrazowego o pierwszym wyrazie a i ilorazie q [1)]
makelist(w,i,L)	w wyrażeniu w, i zmienia się wg elementów listy L
makelist(w,i,ip,ik,k)	tym razem i zmienia się od ip do ik z krokiem k [3)]
sort(L,p)	porządkuje listę L wg reguły (relacji) p

[1)] Funkcje dostępne po wczytaniu pakietu **functs**.
[2)] Funkcje dostępne po wczytaniu pakietu **integer_sequence**.
[3)] Polecenie **makelist** ma jeszcze kilka innych alternatywnych składni.

ZADANIA PRZYKŁADOWE

Przykład 1

Obliczymy trzeci wyraz i sumę trzech pierwszych wyrazów ciągu arytmetycznego o wyrazie początkowym $a_1 = 1$ i różnicy $r = 2$.

```
(%i1) load(functs)$
```

Wczytujemy pakiet **functs**.

```
(%i2) a[n]=arithmetic (a, r, n);
```

(%o2) $a_n = (n-1)\,r + a$

```
(%i3) a[3]=arithmetic (1, 2, 3);
```

(%o3) $a_3 = 5$

```
(%i4) S[n]=arithsum (a, r, n);
```

(%o4) $S_n = n\left(\frac{(n-1)\,r}{2} + a\right)$

```
(%i5) S[3]=arithsum (1, 2, 3);
```

(%o5) $S_3 = 9$

```
(%i6) S[3]=arithmetic (1, 2, 1)+arithmetic (1, 2, 2)+arithmetic (1,2, 3);
```
(%o6) $S_3 = 9$

Przykład 2

Obliczymy trzeci wyraz i sumę trzech pierwszych wyrazów ciągu geometrycznego o wyrazie początkowym $a_1 = 1$ i ilorazie q = 2.

```
(%i1) load(functs)$
```

```
(%i2) geometric (a, q, n);
```
(%o2) $a\,q^{n-1}$

```
(%i3) a[3]=geometric (1, 2, 3);
```
(%o3) $a_3 = 4$

```
(%i4) S[n]=geosum (a, q, n);
```
(%o4) $S_n = \frac{a\,(1-q^n)}{1-q}$

```
(%i5) S[3]=geosum (1, 2, 3);
```
(%o5) $S_3 = 7$

```
(%i6) S[3]=geometric  (1, 2, 1)+geometric  (1, 2, 2)+geometric  (1,2, 3);
```
(%o6) $S_3 = 7$

Przykład 3

Zbadamy monotoniczność i obliczymy granicę ciągu o wyrazie ogólnym

$$a_n = \frac{4n+3}{1-2n}$$

```
(%i1) a[n]:=(4*n+3)/(1-2*n);
```
(%o1) $a_n := \frac{4\,n+3}{1-2\,n}$

```
(%i2) 'a[n+1]=a[n+1];
```
(%o2) $a_{n+1} = \frac{4\,(n+1)+3}{1-2\,(n+1)}$

```
(%i3) 'a[n+1]-'a[n]=factor(a[n+1]-a[n]);
```
(%o3) $a_{n+1} - a_n = \frac{10}{(2\,n-1)(2\,n+1)}$

Ponieważ $a_{n+1} - a_n > 0$ dla $n \in \mathbb{N}$, więc ciąg ten jest rosnący.

```
(%i4) 'limit(a[n], n, inf)=limit(a[n], n, inf);
```

(%o4) $\lim_{n\to\infty} \frac{4n+3}{5-2n} = -2$

Przykład 4

Dany jest ciąg o wyrazie ogólnym $a_n = \frac{n!}{n^n}$. Obliczymy kilka jego wyrazów, zbadamy jego monotoniczność oraz obliczymy granice: $\lim_{n\to\infty} a_n$, $\lim_{n\to\infty} \frac{a_{n+1}}{a_n}$.

```
(%i1) a[n]:=n!/n^n;
```

(%o1) $a_n := \frac{n!}{n^n}$

```
(%i2) a[1];
```

(%o2) 1

```
(%i3) a[5];
```

(%o3) $\frac{24}{625}$

```
(%i4) makelist(display(a[i]),i,[1,2,5,10])$
```

$a_1 = 1$

$a_2 = \frac{1}{2}$

$a_5 = \frac{24}{625}$

$a_{10} = \frac{567}{1562500}$

```
(%i5) assume(n>=1);
```

(%o5) $[n >= 1]$

```
(%i6) 'a[n+1]/'a[n]=a[n+1]/a[n];
      minfactorial(%);
```

(%o6) $\frac{a_{n+1}}{a_n} = \frac{n^n (n+1)^{-n-1} (n+1)!}{n!}$

(%o7) $\frac{a_{n+1}}{a_n} = \frac{n^n}{(n+1)^n}$

```
(%i8) is (rhs(%)<1);
```

(%o8) true

Ponieważ dla ciągu (a_n) o wyrazach dodatnich iloraz $\frac{a_{n+1}}{a_n}$ jest mniejszy niż 1, więc ciąg ten jest malejący.

```
(%i9) 'limit(a[n], n, inf)=limit(a[n], n, inf);
```

(%o9) $\lim\limits_{n\to\infty} \frac{n!}{n^n} = 0$

```
(%i10) 'limit('a[n+1]/'a[n],n, inf)='limit(n^n/(n+1)^n, n, inf);
       'limit('a[n+1]/'a[n],n, inf)=limit(n^n/(n+1)^n, n, inf);
```

(%o10) $\lim\limits_{n\to\infty} \frac{a_{n+1}}{a_n} = \lim\limits_{n\to\infty} \frac{n^n}{(n+1)^n}$

(%o11) $\lim\limits_{n\to\infty} \frac{a_{n+1}}{a_n} = \%e^{-1}$

Przykład 5

Rozważmy ciąg funkcyjny o wyrazie ogólnym $f_n(x) = n \sin\frac{x}{n}$, $x \in \mathbb{R}$, $n \in \mathbb{N}$. Naszkicujemy wykresy czterech pierwszych wyrazów tego ciągu.

```
(%i1) f[n](x) := n*sin(x/n)$
```

```
(%i2) 'f[3](x)=f[3](x);
```

(%o2) $f_3(x) = 3 \sin\left(\frac{x}{3}\right)$

```
(%i3) 'f[3](3*%pi)=f[3](3*%pi);
```

(%o3) $f_3(3\pi) = 0$

```
(%i4) makelist(display(f[n](x)),n,1,4)$
```

$f_1(x) = \sin(x)$

$f_2(x) = 2 \sin\left(\frac{x}{2}\right)$

$f_3(x) = 3 \sin\left(\frac{x}{3}\right)$

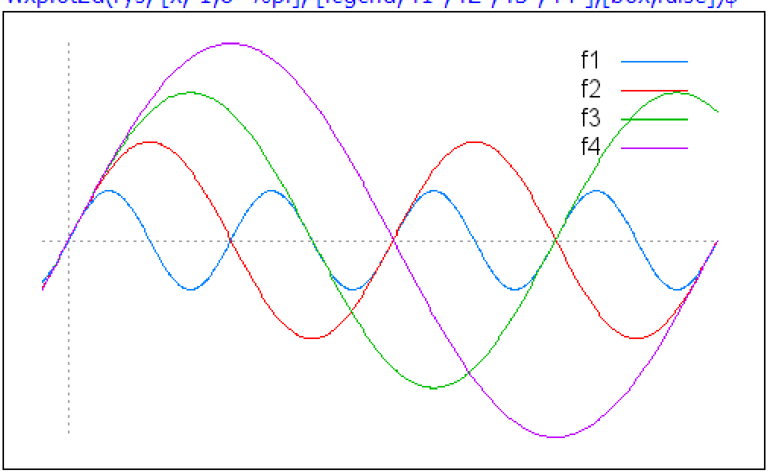
$f_4(x) = 4 \sin\left(\frac{x}{4}\right)$

```
(%i5) rys:f[n](x),n=[1,2,3,4];
```

(%o5) $[\sin(x), 2 \sin\left(\frac{x}{2}\right), 3 \sin\left(\frac{x}{3}\right), 4 \sin\left(\frac{x}{4}\right)]$

```
(%i6) wxplot2d(rys, [x,-1,8*%pi], [legend,"f1","f2","f3","f4"],[box,false])$
```

(%t6)



Przykład 6

Obliczymy przyszłą wartość FV_n kwoty $PV = 1000$ *zł* przy oprocentowaniu rocznym $i = 2\%$ (prostym i złożonym) po 5 latach.

```
(%i1) load(functs)$
```

oprocentowanie proste

– ciąg arytmetyczny o różnicy

$$r = i \cdot PV$$

```
(%i2) arithmetic(PV,i*PV,n+1),factor;
(%o2) (i n+1) PV
(%i3) kill(FV)$
(%i4) FV[n]:=(i*n+1)*PV$
(%i5) FV[1];
(%o5) (i+1) PV
```

Wartość po roku:

```
(%i6) %, PV=1000,i=0.02;
(%o6) 1020.0
(%i7) FV[5];
(%o7) (5 i+1) PV
```

Wartość po pięciu latach:

```
(%i8) %, PV=1000,i=0.02;
(%o8) 1100.0
```

oprocentowanie złożone

– ciąg geometryczny o ilorazie

$$q = 1 + i$$

```
(%i9) geometric(PV,i+1,n+1),factor;
(%o9) (i+1)^n PV
(%i10) kill(FV)$
(%i11) FV[n]:=(i+1)^n*PV$
(%i12) FV[1];
(%o12) (i+1) PV
```

Wartość po roku:

```
(%i13) %, PV=1000,i=0.02;
(%o13) 1020.0
(%i14) FV[5];
(%o14) (i+1)^5 PV
```

Wartość po pięciu latach:

```
(%i15) %, PV:1000,i:0.02;
(%o15) 1104.0808032
```

Po 5 latach wartość przyszła 1000 *zł* przy oprocentowaniu prostym wzrosła o 100 *zł*, a przy oprocentowaniu złożonym o 104,08 *zł*, czyli o 4,08 *zł* więcej.

Uwaga 1

Do obliczania wartości pieniądza w czasie można również wykorzystać funkcje ekonomiczne znajdujące się w pakiecie **finance**.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Zbadać monotoniczność ciągów i obliczyć ich granice:

a) $a_n = \frac{2n}{n+5}$, b) $b_n = \frac{n}{n^2+1}$, c) $c_n = \frac{n^2}{n+3}$, d) $d_n = \frac{n!}{n+1}$.

Zadanie 2

Obliczyć granice ciągów:

a) $\lim\limits_{n\to\infty} \frac{2\cdot 3^n - 6^{n+1}}{6^n+2}$, c) $\lim\limits_{n\to\infty} \sqrt[n]{2n^3-1}$, e) $\lim\limits_{n\to\infty} \left(\frac{n^2}{n^2+3}\right)^{n^2}$,

b) $\lim\limits_{n\to\infty} \frac{2n^2+3}{1-5n^2}$, d) $\lim\limits_{n\to\infty} \left(3n - \sqrt{9n^2+n}\right)$, f) $\lim\limits_{n\to\infty} \left(\frac{1}{n} - 1\right)^n$.

Zadanie 3

Obliczyć przyszłą wartość FV_n kwoty $PV = 20\,000$ *zł* przy oprocentowaniu rocznym $i = 3\%$ (prostym i złożonym) po 4 latach.

Odpowiedzi

1. a) rosnący, 2; b) malejący, 0; c) rosnący, ∞; d) rosnący, ∞.
2. a) -6, b) $-\frac{2}{5}$, c) 1, d) $\frac{1}{6}$ e) e^{-3}, f) nie istnieje.
3. $22\,400$ *zł* (oprocentowanie proste), $22\,510{,}18$ *zł* (oprocentowanie złożone).

12. Szeregi liczbowe i potęgowe

Tabela 12.1

POLECENIA	OPIS
sum(w(k),k,k0,k1**)**	suma wyrażeń postaci $w(k)$ dla k zmieniającego się od $k0$ do $k1$ z krokiem 1 (często daje jedynie zapis symboliczny szeregu)
sum(…), **simpsum**	opcja upraszczająca **sum(…)** np. gdy sumowanie jest do nieskończoności lub do zmiennej
simplify_sum(sum(…)**)**	najbardziej zaawansowane upraszczanie sum [1)]
powerseries(f(x), x, a**)**	szereg Taylora funkcji f w punkcie a
niceindices(…)	indeksowanie postaci i, j, k, l, m, n [2)]
taylor(f(x),x,a,n**)**	wielomian Taylora dla funkcji f stopnia n w punkcie a
S, revert	szereg S odwrotnie uporządkowany [3)]
sum is divergent	komentarz oznaczający, że szereg jest rozbieżny

[1)] Polecenie dostępne po wczytaniu pakietu **simplify_sum**.
[2)] Dotyczy m.in. poleceń: **sum**, **product**, **powerseries**.
[3)] Opcja dostępna po wczytaniu pakietu **revert**.

ZADANIA PRZYKŁADOWE

Przykład 1

Obliczymy sumy:

a) $\sum_{k=1}^{5} 4^k$, b) $\sum_{k=1}^{n} (2k-1)$, c) $\sum_{k=1}^{\infty} \frac{1}{2^k}$, d) $\sum_{k=0}^{\infty} 3\left(\frac{4}{5}\right)^k$, e) $\sum_{k=0}^{n} \binom{n}{k}$.

a) 4+16+64+ ... +1024

```
(%i1) display(sum(4^k, k, 1, 5))$
```

$$\sum_{k=1}^{5} 4^k = 1364$$

b) 1+3+5+7+ ... +2n-1

```
(%i2) display(sum(2*k-1, k, 1, n)), simpsum$
```

$$\sum_{k=1}^{n} 2\,k - 1 = n^2$$

Zauważmy, że Maxima nie zapisuje sumowanych składników w nawiasach. Niemniej jednak powyższy zapis dotyczy sumy $\sum_{k=1}^{n}(2k-1)$.

c) 1/2+1/4+1/8+ ...

```
(%i3) display(sum(1/2^k, k, 1, inf)), simpsum$
```

$$\sum_{k=1}^{\infty}\frac{1}{2^k}=1$$

d) szereg geometryczny: a=3, q=4/5

```
(%i4) display(sum(3*(4/5)^k, k, 0, inf)), simpsum$
```

$$\sum_{k=0}^{\infty}3\left(\frac{4}{5}\right)^k=15$$

e) dwumian Newtona

```
(%i5) display(sum(binomial(n, k), k, 0, n)), simpsum$
```

$$\sum_{k=0}^{n}\binom{n}{k}=2^n$$

```
(%i6) s(n):=''(sum(binomial(n, k), k, 0, n));
```

(%o6) $s(n):=\sum_{k=0}^{n}\binom{n}{k}$

```
(%i7) s(3)=ev(s(3),simpsum);
```

(%o7) $\sum_{k=0}^{3}\binom{3}{k}=8$

Przykład 2

Zbadamy zbieżność szeregów:

a) $\sum_{k=2}^{\infty}\frac{4^k}{3^k}$, b) $\sum_{k=1}^{\infty}\frac{1}{k}$, c) $\sum_{k=1}^{\infty}\frac{1}{k^2}$, d) $\sum_{k=0}^{\infty}\frac{1}{k!}$, e) $\sum_{k=1}^{\infty}\frac{1}{k(k+1)}$.

a) 16/9+64/27+256/81+...

```
(%i1) 'sum((4/3)^k, k, 2, inf); sum((4/3)^k, k, 2, inf), simpsum;
```

(%o1) $$\sum_{k=2}^{\infty} \frac{4^k}{3^k}$$

```
sum: sum is divergent.
 -- an error. To debug this try: debugmode(true);
```

b) szereg harmoniczny rzędu 1: 1+1/2+1/3+1/4+...

```
(%i3) sum(1/k, k, 1, inf);
      sum(1/k, k, 1, inf), simpsum;
      sum(1/k, k, 1, 1000), simpsum,numer;
      sum(1/k, k, 1, 100000), simpsum,numer;
```

(%o3) $$\sum_{k=1}^{\infty} \frac{1}{k}$$

```
sum: sum is divergent.
 -- an error. To debug this try: debugmode(true);
(%o5) 7.485470860550343
(%o6) 12.09014612986328
```

c) szereg harmoniczny rzędu 2: 1+1/4+1/9+1/16+...

```
(%i7) display(sum(1/k^2, k, 1, inf)), simpsum$
```

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

d) 1+1+1/2+1/6+1/24+...

```
(%i8) load("simplify_sum")$
```

```
(%i9) sum(1/k!,k,0,inf)=simplify_sum(sum(1/k!,k,0,inf));
```

(%o9) $$\sum_{k=0}^{\infty} \frac{1}{k!} = \%e$$

e) (1-1/2)+(1/2-1/3)+(1/3-1/4)+...

```
(%i10) 1/k/(k+1)=partfrac(1/k/(k+1),k,1,inf);
       sum(%,k,1,inf);
       simplify_sum(%);
```

(%o10) $\frac{1}{k\,(k+1)}=\frac{1}{k}-\frac{1}{k+1}$

(%o11) $\sum_{k=1}^{\infty}\frac{1}{k\,(k+1)}=\sum_{k=1}^{\infty}\frac{1}{k}-\frac{1}{k+1}$

(%o12) $1=1$

Przykład 3

Wyznaczymy sumy szeregów potęgowych

$$\sum_{k=0}^{\infty} x^k \text{ oraz } \sum_{k=0}^{\infty} kx^k.$$

```
(%i1) assume(abs(x)<1);
```

(%o1) $[|x|<1]$

```
(%i2) sum(x^k, k, 0, inf)=ev(sum(x^k, k, 0, inf),simpsum);
```

(%o2) $\sum_{k=0}^{\infty} x^k=\frac{1}{1-x}$

```
(%i3) diff(%,x,1);
```

(%o3) $\sum_{k=0}^{\infty} k\,x^{k-1}=\frac{1}{(1-x)^2}$

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

(%o4) $x\sum_{k=0}^{\infty} k\,x^{k-1}=\frac{x}{(1-x)^2}$

Przykład 4

Wyznaczymy zbiór tych $x \in \mathbb{R}$, dla których zbieżny jest szereg:

a) $\sum_{n=0}^{\infty}\frac{n!}{2^n}x^n$, b) $\sum_{n=0}^{\infty}\frac{3^n}{n}(x+1)^n$.

a)

```
(%i1) sum(n!/2^n*x^n, n, 1, inf);
```

$$\text{(\%o1)} \quad \sum_{n=1}^{\infty} \frac{n!\,x^n}{2^n}$$

```
(%i2) a[n]:=n!/2^n;
```

$$\text{(\%o2)} \quad a_n := \frac{n!}{2^n}$$

```
(%i3) a[n+1];
```

$$\text{(\%o3)} \quad 2^{-n-1}(n+1)!$$

```
(%i4) 'a[n+1]/'a[n]=a[n+1]/a[n];
```

$$\text{(\%o4)} \quad \frac{a_{n+1}}{a_n} = \frac{(n+1)!}{2\,n!}$$

```
(%i5) minfactorial(%);
```

$$\text{(\%o5)} \quad \frac{a_{n+1}}{a_n} = \frac{n+1}{2}$$

```
(%i6) limit(%,n,inf);
```

$$\text{(\%o6)} \quad \lim_{n \to \infty} \frac{a_{n+1}}{a_n} = \infty$$

Promień zbieżności jest równy zero, zatem szereg potęgowy jest zbieżny tylko dla $x = 0$.

b)

```
(%i1) a[n]:=3^n/n;
```

$$\text{(\%o1)} \quad a_n := \frac{3^n}{n}$$

```
(%i2) f(x):=a[n]*(x+1)^n;
```

$$\text{(\%o2)} \quad f(x) := a_n (x+1)^n$$

```
(%i3) sum(f(x), n, 1, inf);
```

$$\text{(\%o3)} \quad \sum_{n=1}^{\infty} \frac{3^n (x+1)^n}{n}$$

```
(%i4) %,x+1=t;
```

$$\text{(\%o4)} \quad \sum_{n=1}^{\infty} \frac{3^n t^n}{n}$$

```
(%i5) 'a[n]^(1/n)=a[n]^(1/n);
```

$$\text{(\%o5)} \quad a_n^{1/n} = \frac{3}{n^{1/n}}$$

```
(%i6) limit(%,n,inf);
```

$$\text{(\%o6)} \quad \lim_{n \to \infty} a_n^{1/n} = 3$$

```
(%i7) r:1/%;
```

$$\text{(\%o7)} \quad \frac{1}{\lim_{n \to \infty} a_n^{1/n}} = \frac{1}{3}$$

Promień zbieżności dla t wynosi $r = \frac{1}{3}$, zatem szereg potęgowy jest zbieżny dla każdego $t \in \left(-\frac{1}{3}, \frac{1}{3}\right)$, czyli

$$-\frac{1}{3} < t < \frac{1}{3}.$$

Stąd

$$-\frac{1}{3} < x + 1 < \frac{1}{3},$$

co daje

$$-\frac{4}{3} < x < -\frac{2}{3}.$$

Szereg $\sum_{n=0}^{\infty} \frac{3^n}{n} (x+1)^n$ jest więc zbieżny co najmniej dla $x \in \left(-\frac{4}{3}, -\frac{2}{3}\right)$. Sprawdzimy teraz zbieżność rozważanego szeregu na krańcach tego przedziału. Zajmiemy się zatem szeregami liczbowymi.

Niech $x = -\frac{4}{3}$.

```
(%i8) sum(f(x), n, 1, inf);
      %,x=-4/3;
```

$$(\%o8)\quad \sum_{n=1}^{\infty} \frac{3^n (x+1)^n}{n}$$

$$(\%o9)\quad \sum_{n=1}^{\infty} \frac{(-1)^n}{n}$$

Uzyskaliśmy szereg naprzemienny. Spełnia on założenia kryterium Leibniza, gdyż:

Niech $x = -\frac{2}{3}$.

```
(%i10) sum(f(x), n, 1, inf);
       %,x=-2/3;
```

$$(\%o10)\quad \sum_{n=1}^{\infty} \frac{3^n (x+1)^n}{n}$$

$$(\%o11)\quad \sum_{n=1}^{\infty} \frac{1}{n}$$

Uzyskaliśmy szereg harmoniczny rzędu 1, czyli rozbieżny.

```
(%i12) kill(a[n])$ a[n]:=1/n;
```

$$(\%o13)\quad a_n := \frac{1}{n}$$

```
(%i14) 'limit(a[n], n, inf)=limit(a[n],n,inf);
```

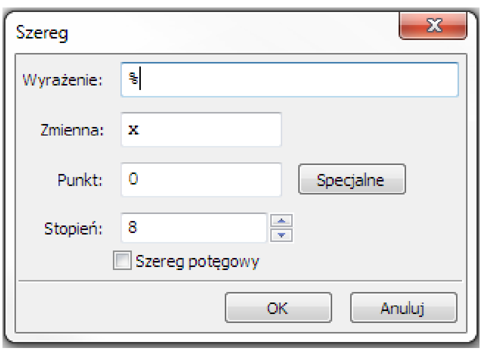
$$(\%o14)\quad \lim_{n \to \infty} \frac{1}{n} = 0$$

```
(%i15) a[n+1]-a[n]=ratsimp(a[n+1]-a[n]);
```

$$(\%o15)\quad \frac{1}{n+1} - \frac{1}{n} = -\frac{1}{n^2+n}$$

Reasumując, szereg $\sum_{n=0}^{\infty} \frac{3^n}{n} (x+1)^n$ jest zbieżny dla $x \in \left[-\frac{4}{3}, -\frac{2}{3}\right)$.

Uwaga 1



W celu rozwinięcia funkcji w szereg potęgowy należy wybrać **MENU-RRC/Rozwiń w szereg**. Maxima poda wtedy kilka początkowych wyrazów rozwinięcia. Jeśli zaznaczymy opcję **Szereg potęgowy**, Maxima wyświetli postać szeregu.
Formularz dostępny też jest z palety **Podstawowa Matematyka**.

Przykład 5

Rozwiniemy w szereg Taylora funkcje:

a) $f(x) = e^x$, b) $f(x) = \sin x$, c) $f(x) = \frac{1}{1+x}$, d) $f(x) = \ln(x+1)$.

a)

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

```
(%i2) f(x)=niceindices(powerseries(f(x), x, 0));
```

(%o2) $\%e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$

```
(%i3) %,x=1;
```

(%o3) $\%e = \sum_{i=0}^{\infty} \frac{1}{i!}$

```
(%i4) sum(1/i!, i, 0, 3),numer;
      sum(1/i!, i, 0, 5),numer;
      sum(1/i!, i, 0, 8),numer;
(%o4) 2.666666666666666
(%o5) 2.716666666666666
(%o6) 2.718278769841270
```

Porównujemy przybliżenia liczby *e* uzyskane przez sumowanie czterech, sześciu oraz dziewięciu pierwszych wyrazów szeregu (%o3)

```
(%i7) %e,numer;
(%o7) 2.718281828459045
```

oraz wartość numeryczną sumy dostępną w Maximie.

Naszkicujemy wykres funkcji $f(x) = e^x$ oraz kolejne jej przybliżenia wielomianami stopni: 2, 5, 7. Zauważmy, że, zwiększając stopień wielomianu, otrzymujemy coraz lepsze przybliżenie funkcji f na coraz większym przedziale.

```
(%i8) t1:taylor(f(x), x, 0, 2);
      t2:taylor(f(x), x, 0, 5);
      t3:taylor(f(x), x, 0, 7);
```

(%o8)/T/ $1 + x + \frac{x^2}{2} + \ldots$

(%o9)/T/ $1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \ldots$

(%o10)/T/ $1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \frac{x^7}{5040} + \ldots$

```
(%i11) plot2d([f(x),t1,t2,t3],[x,-4,2],[y,-1,3],[yx_ratio, 0.5],
                 [legend,"f(x)=exp(x)","t1","t2","t3"],
                 [style,[lines,2],[lines,1],[lines,1],[lines,1]],
                 [gnuplot_preamble,"set key top center;"]  )$
```

Rysunek zamieszczamy na końcu przykładu.

Uwaga 2

Dzięki możliwości rozwinięcia funkcji $f(x) = e^x$ w szereg Taylora możemy znaleźć wartość całki oznaczonej gęstości rozkładu Gaussa: $\int_0^1 \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, dx$.

```
(%i1) integrate(1/(sqrt(2*%pi))*exp(-(x^2)/2), x, 0, 1);
```

(%o1) $\dfrac{\operatorname{erf}\left(\frac{1}{\sqrt{2}}\right)}{2}$

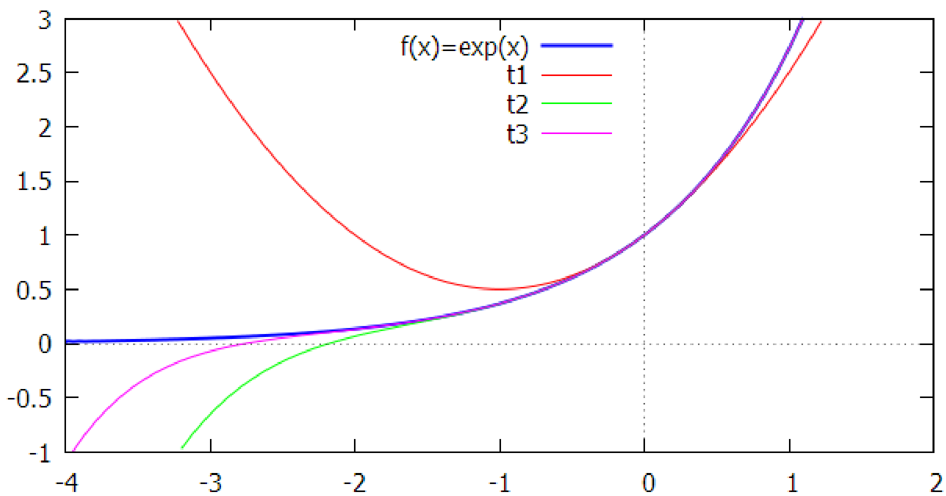
```
(%i2) taylor(1/(sqrt(2*%pi))*exp(-(x^2)/2), x, 0, 6);
```

(%o2)/T/ $\dfrac{1}{\sqrt{2}\sqrt{\pi}}-\dfrac{x^2}{\sqrt{2}^3\sqrt{\pi}}+\dfrac{x^4}{\sqrt{2}^7\sqrt{\pi}}-\dfrac{x^6}{3\sqrt{2}^9\sqrt{\pi}}+\ldots$

```
(%i3) integrate(%, x, 0, 1);
      %,numer;
```

(%o3) $\dfrac{479}{35\,2^{9/2}\sqrt{\pi}}$

(%o4) 0.34123812912908



b)

```
(%i1) f(x):=sin(x)$
      f(x)=niceindices(powerseries(f(x), x, 0));
```

(%o2) $\sin(x)=\sum_{i=0}^{\infty}\dfrac{(-1)^i x^{2i+1}}{(2i+1)!}$

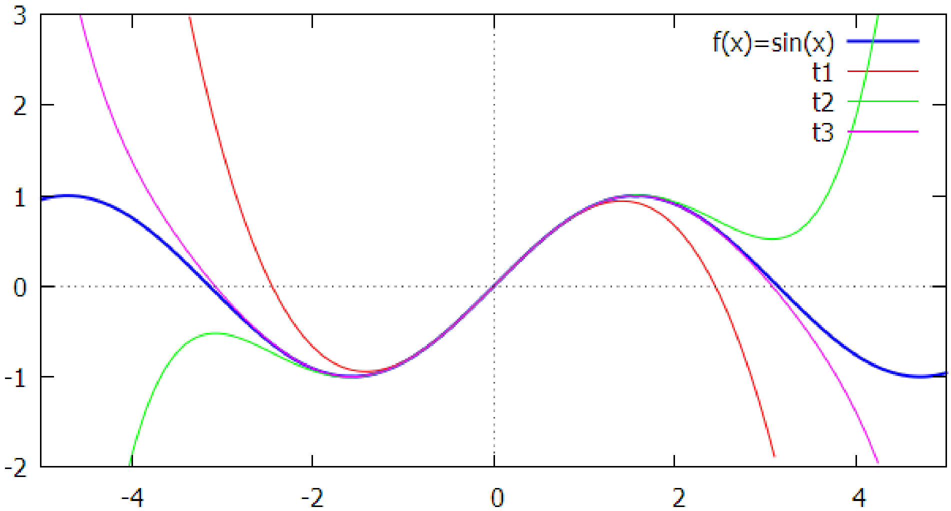
```
(%i3) t1:taylor(f(x), x, 0, 3);
      t2:taylor(f(x), x, 0, 5);
      t3:taylor(f(x), x, 0, 8);
```

(%o3)/T/ $x-\dfrac{x^3}{6}+\ldots$

(%o4)/T/ $x-\dfrac{x^3}{6}+\dfrac{x^5}{120}+\ldots$

(%o5)/T/ $x-\dfrac{x^3}{6}+\dfrac{x^5}{120}-\dfrac{x^7}{5040}+\ldots$

```
(%i6) plot2d([f(x),t1,t2,t3],[x,-5,5],[y,-2,3],[yx_ratio, 0.5],
                    [legend,"f(x)=sin(x)","t1","t2","t3"],
                    [style,[lines,2],[lines,1],[lines,1],[lines,1]]  )$
```



c)

```
(%i1) f(x):=1/(1+x)$
      f(x)=niceindices(powerseries(f(x), x, 0));
      f(x)=taylor(f(x), x, 0, 6);
```

(%o2) $\frac{1}{x+1}=\sum_{i=0}^{\infty}(-1)^i x^i$

(%o3)/T/ $\frac{1}{x+1}=1-x+x^2-x^3+x^4-x^5+x^6+\ldots$

Pamiętamy jednak, że równości (%o2), (%o3) zachodzą jedynie dla x spełniających warunek

$$|x|<1.$$

Mówimy w tym przypadku o szeregu geometrycznym zbieżnym.

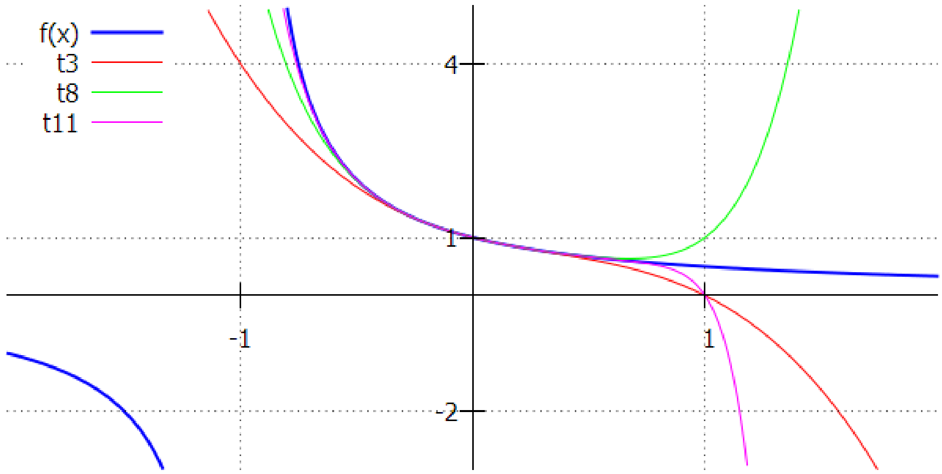
```
(%i4) load("revert")$
      taylor(f(x), x, 0, 6);%, revert;
```

(%o5)/T/ $1-x+x^2-x^3+x^4-x^5+x^6+\ldots$

(%o6)/R/ $x^6-x^5+x^4-x^3+x^2-x+1$

Zauważamy, patrząc na poniższy rysunek, że zwiększając stopień wielomianu otrzymujemy coraz lepsze przybliżenie funkcji f, ale tylko na przedziale $(-1,1)$.

```
(%i10) plot2d([f(x),t3,t6,t11],[x,-2,2],[y,-3,5],[yx_ratio, 0.5],grid2d,
                    [legend,"f(x)","t3","t8","t11"],[axes,solid],[box,false],
                    [xtics, -1,2,1], [ytics, -2,3,4],
                    [style,[lines,2],[lines,1],[lines,1],[lines,1]],
                    [gnuplot_preamble,"set key top left;"]  )$
```



d)

```
(%i1) f(x):=log(x+1)$
      ln(x+1)=niceindices(powerseries(f(x), x, 0));
      changevar(%,i=n+1,n,i);
```

(%o2) $\ln(x+1) = -\sum_{i=1}^{\infty} \frac{(-1)^i x^i}{i}$

(%o3) $\ln(x+1) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{n+1}$

Stosując **changevar**, możemy zmienić nazwy indeksów i zakresy sumowania. To samo polecenie służy również do zamiany zmiennych w całkach (patrz tabela 12.1).

Rozwinięcie funkcji $f(x) = \ln(x+1)$ w szereg potęgowy możemy uzyskać przez całkowanie wyraz po wyrazie rozwinięcia funkcji wymiernej $g(x) = \frac{1}{x+1}$ (patrz podpunkt c)). Promienie zbieżności obu szeregów są jednakowe. W poprzednim podpunkcie ustaliliśmy zbieżność na przedziale $(-1,1)$, natomiast równości $(\%o2)$ i $(\%o3)$ zachodzą przy założeniu, że $x \in (-1,1]$. Dla $x = 1$ otrzymujemy szereg naprzemienny (zbieżny na podstawie kryterium Leibniza), którego sumy częściowe pozwalają przybliżać liczbę $\ln 2$.

```
(%i4) (%o3), x=1;
```

(%o4) $\ln(2) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n+1}$

```
(%i6) 'sum((-1)^n/(n+1),n,0,100);
```

(%o6) $\sum_{n=0}^{100} \frac{(-1)^n}{n+1}$

```
(%i5) log(2),numer;
(%o5) 0.69314718055994
```

```
(%i7) %,nouns,numer;
(%o7) 0.6980731694092
```

Przykład 6

Obliczymy przyszłą wartość po 5 latach ciągu wpłat po $R = 100$ *zł* płatnych z dołu przy oprocentowaniu rocznym $i = 2\%$.

Renta płatna z dołu, $n = 5$, $R = 100$, $i = 0{,}02$. Wartość przyszła jest sumą n skapitalizowanych wpłat postaci $R(1+i)^k$, $k = 0, \dots, n-1$.

```
(%i1) S(R,i,n):=sum(R*(1+i)^k, k, 0, n-1);
```

(%o1) $S(R,i,n) := \sum_{k=0}^{n-1} R(1+i)^k$

```
(%i2) 'sum(100*(1+0.02)^k, k, 0, 4)=S(100,0.02,5);
```

(%o2) $100 \sum_{k=0}^{4} 1.02^k = 520.404016$

Wartość przyszła rozważanego ciągu wpłat wynosi zatem $520{,}40$ *zł*.

Uwaga 3

Na zakończenie przedstawimy symbolicznie szereg Maclaurina, czyli szereg Taylora dla $x = 0$.

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

(%o1) $\sum_{i=0}^{\infty} \frac{x^i \left(\left. \frac{d^i}{d x^i} f(x) \right|_{x=0} \right)}{i!}$

```
(%i2) taylor(f(x), x, 0, 3);
```

(%o2)/T/ $f(0) + \left(\left. \frac{d}{dx} f(x) \right|_{x=0} \right) x + \frac{\left(\left. \frac{d^2}{dx^2} f(x) \right|_{x=0} \right) x^2}{2} + \frac{\left(\left. \frac{d^3}{dx^3} f(x) \right|_{x=0} \right) x^3}{6} + \dots$

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Obliczyć n-tą sumę częściową szeregów:

a) $\sum_{n=0}^{\infty}\left(\frac{1}{3}\right)^n$, b) $\sum_{n=1}^{\infty}(-3)^n$, c) $\sum_{n=0}^{\infty} n$, d) $\sum_{n=0}^{\infty}(-1)^n$.

Zadanie 2

Zbadać zbieżność szeregów:

a) $\sum_{n=0}^{\infty}\frac{n^2}{2^n}$, b) $\sum_{n=0}^{\infty}\frac{n^n}{n!}$, c) $\sum_{n=0}^{\infty}\frac{3n-1}{2n^2+1}$, d) $\sum_{n=1}^{\infty}\cos\frac{1}{n}$

Zadanie 3

Wyznaczyć zbiór tych $x \in \mathbb{R}$, dla których zbieżny jest szereg:

a) $\sum_{n=0}^{\infty}(2x)^{n+1}$, b) $\sum_{n=0}^{\infty}\frac{x^n}{(2n)!}$, c) $\sum_{n=0}^{\infty}\frac{(4x+1)^n}{n+1}$, d) $\sum_{n=0}^{\infty}\frac{n!}{n+2}(5x-1)^{n+2}$.

Zadanie 4

Rozwinąć w szereg Taylora:

a) $f(x) = \frac{1}{1-x}$, b) $f(x) = \frac{1}{1+x^2}$, c) $f(x) = \operatorname{arctg} x$, d) $f(x) = x\cos x$.

Zadanie 5

Wykorzystując szeregi potęgowe, znaleźć przybliżenie liczby:

a) $\ln 3$, b) π, c) $f(x) = \cos 2$, d) $f(x) = \operatorname{tg} 3$.

Zadanie 6

Obliczyć przyszłą wartość po 4 latach ciągu wpłat po R= $20\,000\ zł$ płatnych z góry (wpłaty następują na początku każdego roku, nie na końcu) przy oprocentowaniu rocznym $i = 3\%$.

Odpowiedzi

1. a) $-\frac{3}{2}\left(\frac{1}{3^{n+1}}-1\right)$, b) $-\frac{(-3)^{n+1}+3}{4}$, c) $\frac{n^2+n}{2}$, d) $-\frac{(-1)^{n+1}+1}{2}$.
2. a) zbieżny, b) rozbieżny, c) rozbieżny, d) rozbieżny.
3. a) $\left(-\frac{1}{2},0\right)$, b) $\mathbb{R}$, c) $[-2,0)$, d) $\frac{1}{5}$.
4. a) $\sum_{n=0}^{\infty}x^n$, b) $\sum_{n=0}^{\infty}(-1)^n x^{2n}$, c) $\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}x^{2n+1}$, d) $\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}x^{2n+1}$.
5. a) 1.09861228866811, b) 3.141592653589793, c) -0.41614683654714, d) -0.14254654307428.
6. 86 182,80 zł.

13. Szeregi Fouriera

Szeregiem trygonometrycznym nazywamy szereg postaci:

$$\frac{a_0}{2}+\sum_{n=1}^{\infty}\left(a_n\cos\frac{n\pi x}{t}+b_n\sin\frac{n\pi x}{t}\right),$$

gdzie t jest dowolnie ustaloną dodatnią liczbą rzeczywistą, natomiast $a_0, a_n, b_n \in \mathbb{R}$.
Szeregiem Fouriera funkcji f całkowalnej na przedziale $[-t,t]$ nazywamy szereg trygonometryczny, w którym współczynniki a_n i b_n określone są wzorami Eulera - Fouriera:

$$a_n=\frac{1}{t}\int_{-t}^{t} f(x)\cos\frac{n\pi x}{t}dx,\;\; n=0,1,2,\ldots\;\;,\qquad b_n=\frac{1}{t}\int_{-t}^{t} f(x)\sin\frac{n\pi x}{t}dx,\;\; n=1,2,\ldots$$

Mówimy, że funkcja f spełnia warunki Dirichleta na przedziale $[-t,t]$, gdy:

- ma ona na tym przedziale skończoną liczbę punktów nieciągłości jedynie pierwszego rodzaju i w każdym punkcie nieciągłości wartość funkcji jest równa średniej arytmetycznej granic prawo i lewostronnej funkcji w tym punkcie,
- w punktach $-t$ i t wartości funkcji są równe średniej arytmetycznej granic funkcji: prawostronnej w punkcie $-t$ i lewostronnej w punkcie t,
- przedział $[-t,t]$ można podzielić na skończoną liczbę podprzedziałów tak, by na każdym z tych przedziałów otwartych funkcja f była ciągła i monotoniczna.

Jeżeli funkcja f spełnia na przedziale $[-t,t]$ warunki Dirichleta, to jest ona sumą swojego szeregu Fouriera.

Tabela 13.1

POLECENIA	OPIS
fourier(f,x,p**)**	podaje współczynniki rozwinięcia w szereg Fouriera funkcji f zdefiniowanej na przedziale $[-p,p]$
fourcos(f,x,p**)**	podaje współczynniki rozwinięcia w szereg Fouriera według cosinusów funkcji f zdefiniowanej na przedziale $[0,p]$
foursin(f,x,p**)**	podaje współczynniki rozwinięcia w szereg Fouriera według sinusów funkcji f zdefiniowanej na przedziale $[0,p]$

POLECENIA	OPIS
foursimp(l)	upraszcza listę *l* współczynników rozwinięcia w szereg Fouriera: $\sin(n\pi)$ do wartości 0 oraz $\cos(n\pi)$ do wartości $(-1)^n$
fourexpand(l,x,p,limit**)**	rozwinięcie w szereg Fouriera na podstawie listy współczynników *l* [1)]
totalfourier(f,x,p)	rozwinięcie w szereg Fouriera funkcji f zdefiniowanej na przedziale $[-p,p]$

[1)] W przypadku gdy zmienna *limit* jest liczbą naturalną, otrzymujemy sumę częściową szeregu Fouriera, natomiast gdy przyjmuje wartość *inf* , otrzymujemy sam szereg.

ZADANIA PRZYKŁADOWE

Przykład 1

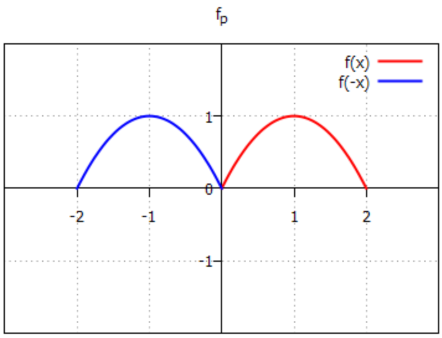
Rozważmy funkcję $f(x) = 2x - x^2$ dla $x \in [0,2]$. Narysujemy parzyste oraz nieparzyste przedłużenie funkcji f na przedział $[-2,2]$.

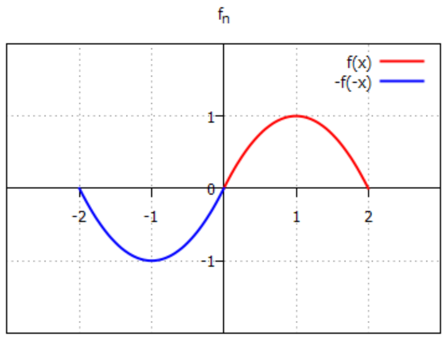
Parzyste i nieparzyste przedłużenie funkcji f oznaczymy odpowiednio f_p i f_n . Funkcje te możemy zdefiniować następująco:

$$f_p(x) = \begin{cases} f(-x), & -2 \le x < 0, \\ f(x), & 0 \le x \le 2, \end{cases} \qquad f_n(x) = \begin{cases} -f(-x), & -2 \le x < 0, \\ f(x), & 0 \le x \le 2, \end{cases}$$

Najpierw zadeklarujemy funkcję f, a następnie, korzystając z poleceń pakietu **draw**, narysujemy funkcje f_p oraz f_n.

```
(%i1)   f(x):=if x>=0 and x<=2  then 2*x-x^2$

(%i3)   load(draw)$
        set_draw_defaults( terminal=wxt, dimensions=[800,300], grid=true,
                           proportional_axes=xy, xaxis=true, yaxis=true,
                           xlabel=" ", ylabel=" ", xtics_axis=true, ytics_axis=true,
                           xaxis_type=solid, yaxis_type=solid,
                           xtics={-2,-1,1,2}, ytics={-1,0,1}, line_width=2 )$
(%i6)   g1:gr2d( xrange=[-3,3], yrange=[-2,2], title="f_p",
                key="f(x)", color=red, explicit(f(x),x,0,2),
                key="f(-x)", color=blue, explicit(f(-x),x,-2,0))$
        g2:gr2d( xrange=[-3,3], yrange=[-2,2], title="f_n",
                key="f(x)", color=red, explicit(f(x),x,0,2),
                key="-f(-x)", color=blue, explicit(-f(-x),x,-2,0))$
        draw(columns=2,g1,g2)$
```





Uwaga 1

Obie funkcje f_p oraz f_n spełniają warunki Dirichleta na przedziale $[-2,2]$ (są więc rozwijalne w szereg Fouriera na tym przedziale).

Przykład 2

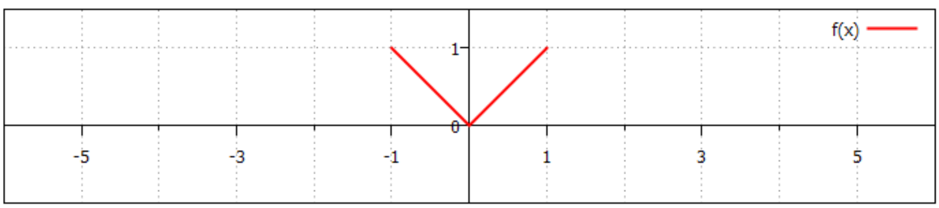
Rozważmy funkcję $f(x) = |x|$ dla $x \in [-1,1]$. Narysujemy okresowe przedłużenie funkcji f na przedział $[-5,5]$.

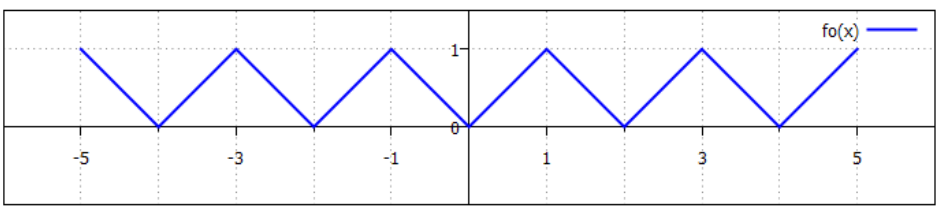
Zaczniemy od zdefiniowania funkcji f oraz jej przedłużenia f_o.

```
(%i1) f(x):=if x>=-1 and x<=1 then abs(x)$
```

```
(%i1) fo(x):=1/%pi*acos(cos(%pi*x))$
```

Zauważmy, że w drugiej z powyższych definicji użyliśmy funkcji trygonometrycznej cosinus oraz funkcji do niej odwrotnej - arcuscosinus.
Współczynniki $\frac{1}{\pi}$ i π zmieniły odpowiednio amplitudę i okres podstawowy funkcji złożonej $\arccos(\cos(x))$ z wartości 2π na 2 .





Powyższe wykresy otrzymaliśmy korzystając z poleceń i opcji pakietu **draw**.

```
(%i3)   load(draw)$
        set_draw_defaults( terminal=wxt, dimensions=[900,350], grid=[2,2],
                           xrange=[-6,6], yrange=[-1,1.5],
                           proportional_axes=xy, xlabel=" ", ylabel=" ",
                           xaxis=true, yaxis=true, xtics=[-5,2,5], ytics={0,1},
                           xtics_axis=true, ytics_axis=true, line_width=2,
                           xaxis_type=solid, yaxis_type=solid )$
```

```
(%i6)   g1:gr2d( key="f(x)", color=red, explicit(f(x),x,-1,1))$
        g2:gr2d( key="fo(x)", color=blue, explicit(fo(x),x,-5,5) )$
        draw(columns=1,g1,g2)$
```

Uwaga 2

Wykres funkcji f_o nazywany jest w zastosowaniach falą o przebiegu trójkątnym. Oczywiście, funkcja f_o, zarówno na przedziale $[-5{,}5]$ jak i na każdym innym domkniętym przedziale symetrycznym, spełnia warunki Dirichleta.

Przykład 3

Rozważmy szereg trygonometryczny

$$\sum_{n=1}^{\infty} \frac{1}{n} \sin(nx)$$

Narysujemy wykresy kilku pierwszych składników oraz sum częściowych tego szeregu na przedziale $[-5{,}5]$.

Dla wygody rysowania, zdefiniujemy wyrazy szeregu jako funkcję f zmiennej n oraz k - tą sumę częściową s jako funkcję zmiennej k.

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

```
(%i2)   s(k):=sum(f(n),n,1,k),simpsum$
```

Sprawdźmy teraz działanie funkcji s razem z poleceniem **trunc**, które dodaje przy sumach znak plus oraz wielokropek (ustawia też składniki zgodnie z indeksowaniem).

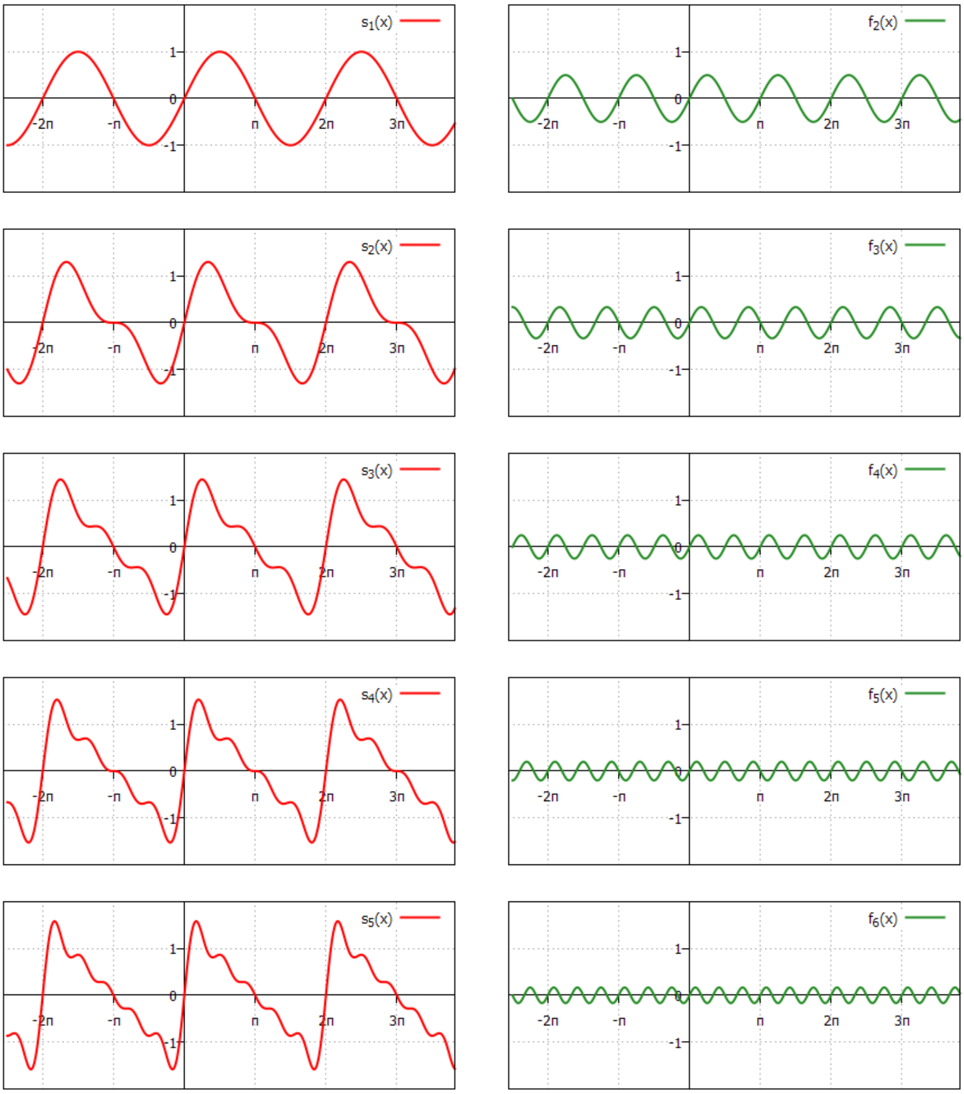
```
(%i3)   trunc(s(4));
```

(%o3) $\sin(x) + \frac{\sin(2\,x)}{2} + \frac{\sin(3\,x)}{3} + \frac{\sin(4\,x)}{4} + \ldots$

Wykresy przedstawimy w dwóch kolumnach: z lewej strony - sumy częściowe, a z prawej - dodawane do tych sum kolejne składniki. Dla obu kolumn będziemy obserwować okresy podstawowe oraz amplitudy.

Zauważmy, że:

$$s_1(x) = f_1(x),\ s_2(x) = s_1(x) + f_2(x),\ s_3(x) = s_2(x) + f_3(x),\ \ \dots$$



Wszystkie sumy mają okres podstawowy równy 2π, a amplitudy coraz większe. Składniki mają natomiast coraz mniejsze amplitudy i coraz mniejszy okres podstawowy (czyli coraz większą częstotliwość).

Skorzytaliśmy tu (podobnie jak w poprzednich przykładach) z pakietu **draw**.

```
(%i5)   load(draw)$
        set_draw_defaults( terminal=wxt, dimensions=[900,1000], grid=[2,2],
                          xrange=[-8,12], yrange=[-2,2],
                          xlabel=" ", ylabel=" ", font_size=12,
                          xtics_axis=true, ytics_axis=true,
                          xaxis_type=solid, yaxis_type=solid,
                          xaxis=true, yaxis=true,
                          xtics={["{/Symbol -2p}",-6.28],
                                 ["{/Symbol -p}",-3.14],
                                 ["{/Symbol p}",3.14],
                                 ["{/Symbol 2p}",6.28],
                                 ["{/Symbol 3p}",9.42]},
                          ytics={-1,0,1}, line_width=2 )$
```

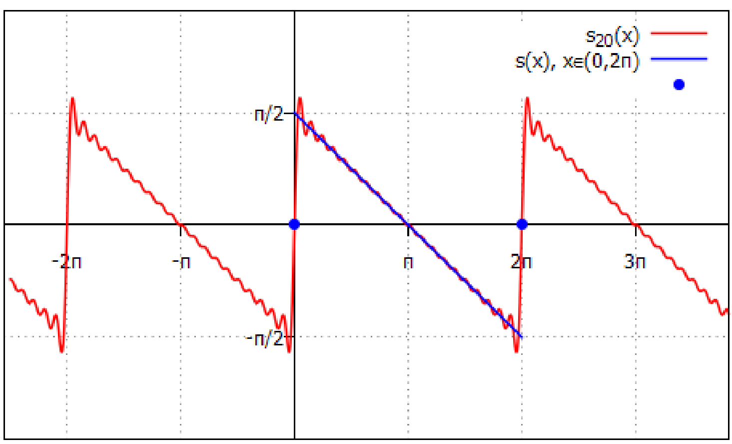
```
(%i8)   fn(i):=gr2d( key=sconcat("f_",{i},"(x)"),
                    color=forest_green, explicit(f(i),x,-2.5*%pi,4*%pi))$
        sk(i):=gr2d( key=sconcat("s_",{i},"(x)"),
                    color=red, explicit(s(i),x,-2.5*%pi,4*%pi))$
        draw(columns=2,
             sk(1),fn(2),sk(2),fn(3),sk(3),fn(4),sk(4),fn(5),sk(5),fn(6))$
```

Nietrudno odgadnąć funkcję, która jest sumą rozważanego szeregu na przedziale $[0,2\pi]$.

$$s(x) = \begin{cases} \dfrac{\pi - x}{2}, & 0 < x < 2\pi, \\ 0, & x \in \{0,2\pi\}, \end{cases}$$

Zdefiniujemy funkcję suma i sprawdzimy dopasowanie wykresów: funkcji oraz 20-tej sumy częściowej szeregu. Punkty na krańcach dołączymy osobno.

```
(%i9)   suma(x):=if x>0 and x<2*%pi then (%pi-x)/2$
```



```
(%i10) sz:draw2d( dimensions=[500,300], yrange=[-3,3], line_width=1.5,
           ytics={["{/Symbol -p/2}",-1.57],
                  ["{/Symbol p/2}",1.57]}, grid=true,
           color=red, key="s_{20}(x)",
           explicit(s(20),x,-2.5*%pi,4*%pi),
           color=blue, key="s(x), x∈(0,2п)",
           explicit(suma(x),x,0,2*%pi),
           key=" ", point_type=filled_circle,
           points([0,2*%pi],[0,0]))$
```

Przykład 4

Podamy współczynniki rozwinięcia w szereg Fouriera funkcji $f(x) = 2 - x$ dla $x \in (-2,2)$. Narysujemy wykresy kilku sum częściowych tego rozwinięcia razem z wykresem rozszerzonej (na przedział domknięty $[-2,2]$) funkcji f.

```
(%i2) f(x):=2-x$ t:2$
```

Najpierw definiujemy funkcję f oraz przypisujemy zmiennej t wartość 2.

```
(%i3) declare(n,integer)$
```

Wprowadzamy deklarację, że zmienna n jest liczbą całkowitą. Wtedy, przy dalszych obliczeniach, otrzymamy, że:

$$\cos(n\pi) = \sin\left(n\pi + \frac{\pi}{2}\right) = (-1)^n,$$

$$\sin(n\pi) = \cos\left(n\pi + \frac{\pi}{2}\right) = 0,$$

```
(%i4) load(fourie)$
(%i5) w:fourier(f(x),x,2)$
```

(w) $a_0 = 2$

(w) $a_n = 0$

(w) $b_n = \dfrac{4(-1)^n}{\pi n}$

Wczytujemy pakiet **fourie**, a następnie korzystamy z polecenia **fourier**, aby wyznaczyć współczynniki rozwinięcia rozważanej funkcji w szereg.

```
(%i9) fourexpand(w,x,t,inf); intosum(%);
```

(%o8) $\dfrac{4\displaystyle\sum_{n=1}^{\infty}\frac{(-1)^n \sin\left(\frac{\pi n x}{2}\right)}{n}}{\pi} + 2$

(%o9) $\left(\displaystyle\sum_{n=1}^{\infty}\frac{4(-1)^n \sin\left(\frac{\pi n x}{2}\right)}{\pi n}\right) + 2$

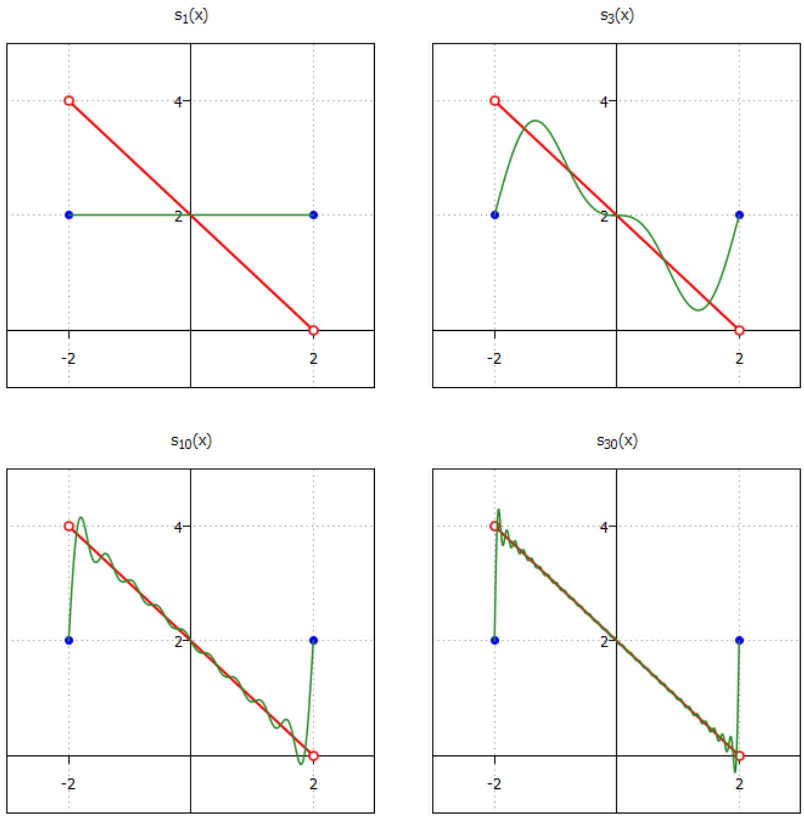
Rozwijamy funkcję f w szereg Fouriera na bazie współczynników w.

Stosujemy polecenie **intosum**, aby stałe 4 oraz π znalazły się pod symbolem sumy.

```
(%i10) s(n):=fourexpand(w,x,t,n)$
```

Definiujemy funkcję s , której wartości, to n-te sumy rozwinięcia w szereg Fouriera.

Po wczytaniu pakietu **draw** i wykonaniu ustawień za pomocą polecenia **set_draw_defaults** narysujemy wykres funkcji f (kolor czerwony) rozszerzonej o wartości 2 na krańcach przedziału (punkty niebieskie) wraz z wykresami następujących sum częściowych: s_1, s_3, s_{10}, s_{30} (kolor zielony).



Funkcja, która jest rozszerzeniem funkcji f (patrz wykresy powyżej) spełnia warunki Dirichleta, zatem jest ona sumą swojego szeregu Fouriera dla $x \in [-2,2]$.

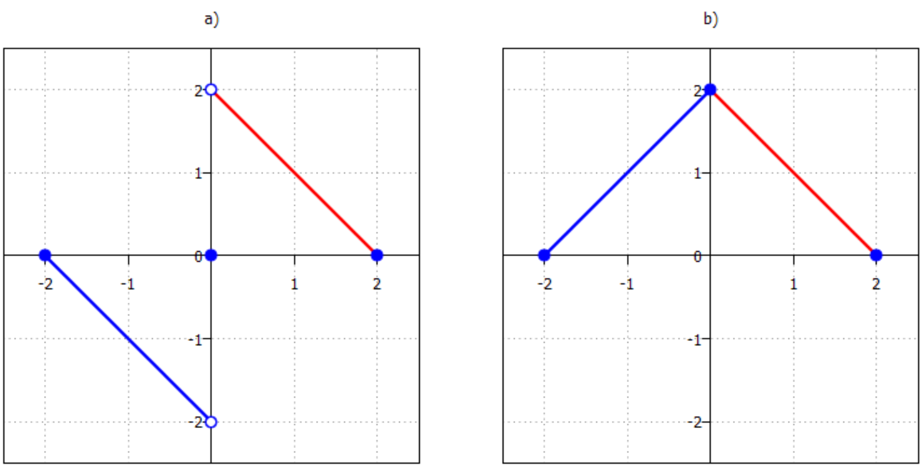
```
(%i12)  load(draw)$
        set_draw_defaults( terminal=wxt, dimensions=[700,700], grid=true,
                           xlabel=" ", ylabel=" ", xaxis=true, yaxis=true,
                           xtics_axis=true, ytics_axis=true, line_width=2,
                           xaxis_type=solid, yaxis_type=solid,
                           xtics={-2,2}, ytics={2,4} )$
(%i14)  gr[i]:=gr2d(xrange=[-3,3], yrange=[-1,5], title=sconcat("s_",{i},"(x)"),
                color=blue, point_type=filled_circle, points([[2,2],[-2,2]]),
                color=red, explicit(f(x),x,-2,2),
                point_size=1.25, points([[2,0],[-2,4]]),
                color=white, point_size=0.75, points([[2,0],[-2,4]]),
                line_width=1.5, color=forest_green, explicit(s(i-1),x,-2,2))$
        draw(columns=2,gr[1],gr[3],gr[10],gr[30])$
```

Przykład 5

Podamy współczynniki rozwinięcia funkcji $f(x) = 2 - x$ dla $x \in (0,2)$ w szereg Fouriera: a) według sinusów, b) według cosinusów.

Najpierw (podobnie jak w przykładzie 1) narysujemy wykres przedłużenia nieparzystego f_a oraz przedłużenia parzystego f_b funkcji f na przedział $[-2,2]$.

```
(%i2)   f(x):=2-x$ t:2$

(%i4)   load(draw)$
        set_draw_defaults( terminal=wxt, dimensions=[800,400],
                           xrange=[-2.5,2.5], yrange=[-2.5,2.5], grid=true,
                           proportional_axes=xy, xlabel=" ", ylabel=" ",
                           xaxis=true, yaxis=true, font_size=14,
                           xtics_axis=true, ytics_axis=true, point_size=1.5,
                           xtics={-2,-1,1,2}, ytics=[-2,1,2], line_width=2.5,
                           xaxis_type=solid, yaxis_type=solid )$

(%i7)   fa:gr2d( title="a)",
                 color=red, explicit(f(x),x,0,2),
                 color=blue, line_type=dots, explicit(-f(-x),x,-2,0),
                 point_type=filled_circle, points([[-2,0],[2,0],[0,0],[0,2],[0,-2]]),
                 color=white, point_size=1, points([[0,2],[0,-2]]))$
        fb:gr2d( title="b)",
                 color=red, explicit(f(x),x,0,2),
                 color=blue, line_type=dots, explicit(f(-x),x,-2,0),
                 point_type=filled_circle, points([[-2,0],[2,0],[0,2]]) )$
        draw(columns=2,fa,fb)$
```



Obie funkcje f_a oraz f_b spełniają warunki Dirichleta, więc są równe swojemu rozwinięciu w szereg Fouriera na przedziale $[-2,2]$.

Wczytamy pakiet **fourie** i zadeklarujemy zakres zmiennej n.

```
(%i11) load(fourie)$
(%i12) declare(n,integer)$
```

Uwaga 3

W dalszej części przykładu 5 i w kolejnych przykładach ustawimy w Preferencjach programu (zakładka styl) czcionkę matematyczną - *Calibri*, aby wyraźniej wyświetlała się liczba π.

Wyznaczymy współczynniki rozwinięcia funkcji f w szereg sinusów (polecenie **foursin**) oraz w szereg cosinusów (polecenie **fourcos**). Na bazie tych współczynników i polecenia **fourexpand** podamy postać szeregów przekształconą poleceniem **intosum**.

```
(%i10) wa:foursin(f(x),x,t)$
```

(wa) $b_n = \frac{4}{\pi n}$

```
(%i12) fourexpand(wa,x,t,inf); intosum(%);
```

(%o11) $\frac{4 \sum_{n=1}^{\infty} \frac{\sin\left(\frac{\pi n x}{2}\right)}{n}}{\pi}$

(%o12) $\sum_{n=1}^{\infty} \frac{4 \sin\left(\frac{\pi n x}{2}\right)}{\pi n}$

```
(%i15) wb:fourcos(f(x),x,t)$ wb:foursimp(wb)$
```

(wb) $a_0 = 1$

(wb) $a_n = \frac{4}{\pi^2 n^2} - \frac{4 (-1)^n}{\pi^2 n^2}$

(wb) $a_0 = 1$

(wb) $a_n = -\frac{4 ((-1)^n - 1)}{\pi^2 n^2}$

```
(%i18) fourexpand(wb,x,t,inf); intosum(%);
```

$$\text{(\%o17)}\quad 1-\frac{4\displaystyle\sum_{n=1}^{\infty}\frac{((-1)^n-1)\cos\left(\frac{\pi n x}{2}\right)}{n^2}}{\pi^2}$$

$$\text{(\%o18)}\quad \left(\sum_{n=1}^{\infty}-\frac{4((-1)^n-1)\cos\left(\frac{\pi n x}{2}\right)}{\pi^2 n^2}\right)+1$$

Na zakończenie tego przykładu zdefiniujemy funkcje F_a i F_b, które pozwolą wyznaczać n-te sumy częściowe szeregów Fouriera. Przedstawimy też ilustrację (w formie animacji) funkcji f_a oraz funkcji f_b wraz ze zmieniającymi się sumami częściowymi.

```
(%i19) Fa(n):=fourexpand(wa,x,t,n)$
(%i20) trunc(Fa(5));
```

$$\text{(\%o20)}\quad \frac{4\sin\left(\frac{\pi x}{2}\right)}{\pi}+\frac{2\sin(\pi x)}{\pi}+\frac{4\sin\left(\frac{3\pi x}{2}\right)}{3\pi}+\frac{\sin(2\pi x)}{\pi}+\frac{4\sin\left(\frac{5\pi x}{2}\right)}{5\pi}+\ldots$$

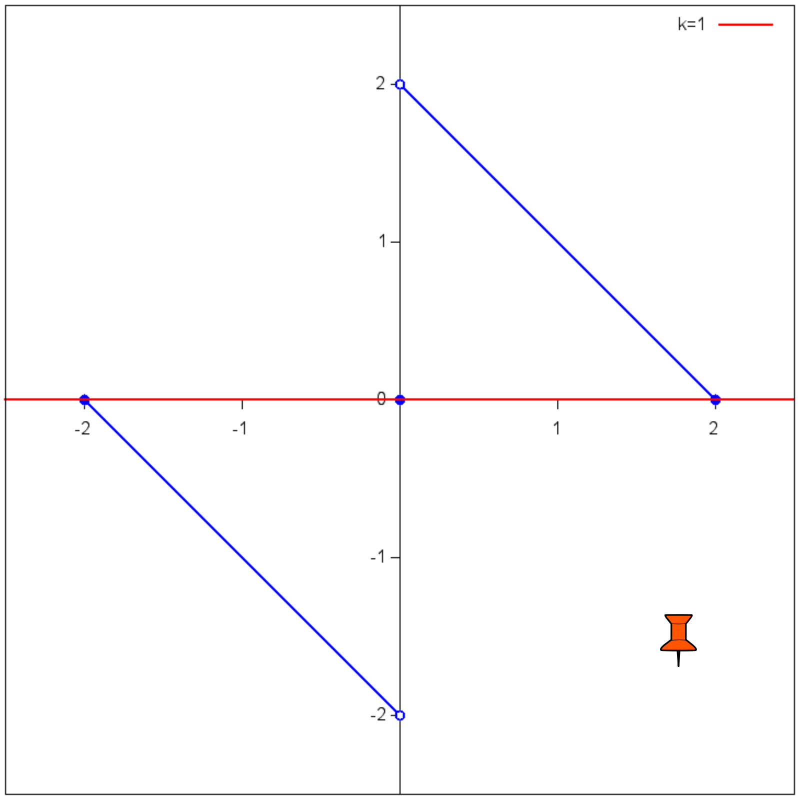
```
(%i21) Fb(n):=fourexpand(wb,x,t,n)$
(%i22) trunc(Fb(5));
```

$$\text{(\%o22)}\quad 1+\frac{8\cos\left(\frac{\pi x}{2}\right)}{\pi^2}+\frac{8\cos\left(\frac{3\pi x}{2}\right)}{9\pi^2}+\frac{8\cos\left(\frac{5\pi x}{2}\right)}{25\pi^2}+\ldots$$

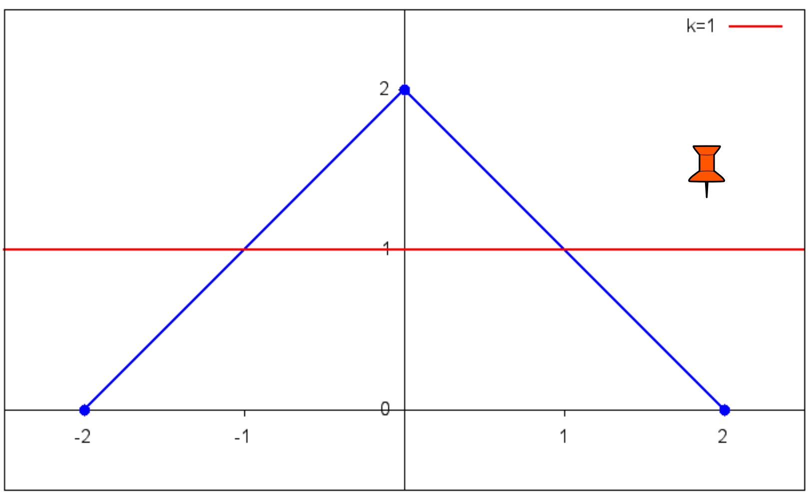
Zanim, za pomocą poleceń **apply**(draw, …), wygenerujemy animowane gify, wyczyścimy wcześniejsze ustawienia domyślne dla pakietu **draw**.

```
(%i23) set_draw_defaults()$
```

a)



b)



```
(%i24)  apply(draw,
           append( [ terminal='animated_gif, delay=70,
                     file_name="E:/Szereg_Fouriera_a",
                     dimensions=[800,800]],
           makelist( gr2d( title="a)", proportional_axes=xy,
                     xrange=[-2.5,2.5], yrange=[-2.5,2.5],
                     font_size=14, point_size=1.5,
                     xlabel=" ", ylabel=" ", line_width=2.5,
                     xaxis=true, yaxis=true,
                     xtics_axis=true, ytics_axis=true,
                     xtics={-2,-1,1,2}, ytics=[-2,1,2],
                     xaxis_type=solid, yaxis_type=solid ,
                     explicit(f(x),x,0,2),
                     color=blue, explicit(-f(-x),x,-2,0),
                     point_type=filled_circle,
                     points([[-2,0],[2,0],[0,0],[0,2],[0,-2]]),
                     color=white, point_size=1,
                     points([[0,2],[0,-2]]),
                     color=red, key=sconcat("k=",k),
                     explicit(Fa(k-1),x,-2.5,2.5)), k,1,10)) )$
```

```
(%i25)  apply(draw,
           append( [ terminal='animated_gif, delay=70,
                     file_name="E:/Szereg_Fouriera_b",
                     dimensions=[800,500]],
           makelist( gr2d( title="b)", proportional_axes=xy,
                     xrange=[-2.5,2.5], yrange=[-0.5,2.5],
                     font_size=14, point_size=1.5,
                     xlabel=" ", ylabel=" ", line_width=2.5,
                     xaxis=true, yaxis=true,
                     xtics_axis=true, ytics_axis=true,
                     xtics={-2,-1,1,2}, ytics=[0,1,2],
                     xaxis_type=solid, yaxis_type=solid ,
                     explicit(f(x),x,0,2),
                     explicit(f(-x),x,-2,0),
                     point_type=filled_circle,
                     points([[-2,0],[2,0],[0,2]]),
                     color=red, key=sconcat("k=",k),
                     explicit(Fb(k-1),x,-2.5,2.5)), k,1,10)) )$
```

Przykład 6

Rozwiniemy w szereg Fouriera funkcję

$$f(x) = \begin{cases} 0, & -\pi < x < 0, \\ x, & 0 \le x < \pi, \\ \frac{\pi}{2}, & x \in \{-\pi, \pi\}. \end{cases}$$

Definiowanie funkcji sklejanych w Maximie jest raczej kłopotliwe. Można zdefiniować funkcję f używając konstrukcji warunkowej **if**:

```
(%i1)   f(x):=if x>-%pi and x<0 then 0
              elseif x>=0 and x<%pi then x
              elseif x=-%pi or x=%pi then %pi/2$
```

Wartości funkcji zostają wtedy podane poprawnie:

```
(%i2)   [f(-%pi),f(-1),f(2)];
```

(%o2) $[\frac{\pi}{2}, 0, 2]$

Pojawia się natomiast problem, gdy chcemy scałkować tak zdefiniowaną funkcję. Zdefiniujemy więc funkcję f inaczej, aby móc ją rozwinąć w szereg Fouriera. Użyjemy w tym celu funkcji charakterystycznej zbioru tj. funkcji, która przyjmuje wartość 1 na podanym zbiorze, a poza nim wartość 0.

```
(%i5)   f(x):=x*charfun(x>0)$
        t:%pi$
        declare(n,integer)$
```

Niestety, w przypadku większości funkcji sklejanych, polecenie **fourier** nie działa. Dlatego wyznaczymy współczynniki a_0, a_n oraz b_n obliczając odpowiednie całki na przedziałach $(-\pi, 0)$ oraz $(0, \pi)$. Zauważmy jednak, że w tym przykładzie wszystkie całki na przedziale $(-\pi, 0)$ są równe 0 i można było je pominąć.

```
(%i8)   w1:a[0]=1/(2*t)*( integrate(f(x), x,-t,0)+integrate(f(x), x,0,t) );
        w2:a[n]=1/t*( integrate(f(x)*cos(n*x), x,-t,0)
                    +integrate(f(x)*cos(n*x), x,0,t) );
        w3:b[n]=1/t*( integrate(f(x)*sin(n*x), x,-t,0)
                    +integrate(f(x)*sin(n*x), x,0,t) );
```

(w1) $a_0 = \frac{\pi}{4}$

(w2) $a_n = \frac{\frac{(-1)^n}{n^2} - \frac{1}{n^2}}{\pi}$

(w3) $b_n = -\frac{(-1)^n}{n}$

Mając wyznaczone współczynniki a_0, a_n oraz b_n, skorzystamy z polecenia **fourexpand**, a następnie przekształcimy otrzymaną sumę.

```
(%i12) load(fourie)$
       fourexpand([w1,w2,w3],x,t,inf)$
       intosum(%)$ sumcontract(%);
```

(%o12) $$\left(\sum_{n=1}^{\infty} \frac{\left(\frac{(-1)^n}{n^2}-\frac{1}{n^2}\right)\cos(n\,x)}{\pi}-\frac{(-1)^n\sin(n\,x)}{n}\right)+\frac{\pi}{4}$$

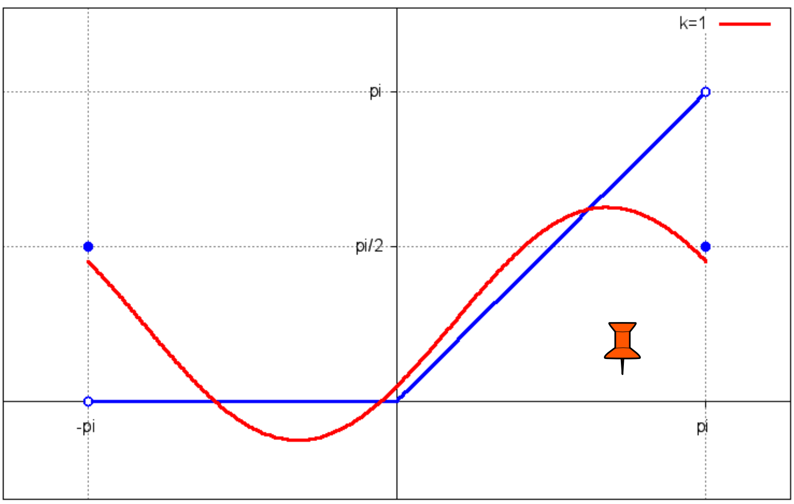
Zdefiniujemy teraz (podobnie jak w poprzednich przykładach) n-tą sumę częściową szeregu i pokażemy, jaka jest postać sumy s_3.

```
(%i13) s(n):=fourexpand([w1,w2,w3],x,t,n)$
```

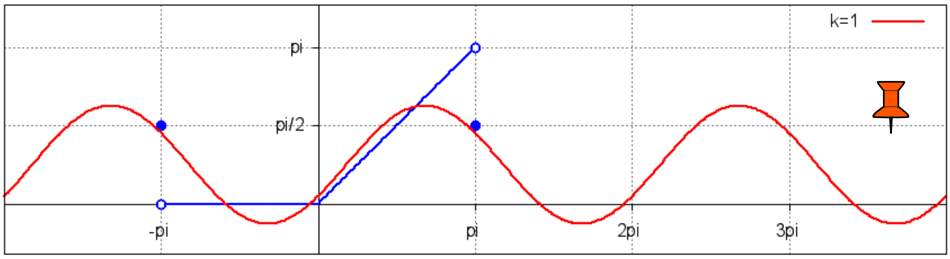
```
(%i14) trunc(s(3));
```

(%o14) $$\frac{\pi}{4}-\frac{2\cos(x)}{\pi}+\sin(x)-\frac{\sin(2\,x)}{2}-\frac{2\cos(3\,x)}{9\,\pi}+\frac{\sin(3\,x)}{3}+\ldots$$

Na poniższych rysunkach widoczna jest funkcja f i wybrane sumy częściowe otrzymanego wyżej szeregu Fouriera. W pierwszym przypadku rozważamy sumy i funkcję f na przedziale $[-\pi,\pi]$, a w drugim - funkcję f i sumy częściowe jej okresowego przedłużenia na przedziale $[-2\pi,4\pi]$.



```
(%i15)  load(draw)$

(%i16)  apply(draw,
            append( [ terminal='animated_gif, delay=90,
                      file_name="E:/Szereg_Fouriera_f_sk",
                      dimensions=[900,500]],
            makelist( gr2d( proportional_axes=xy, grid=true,
                            xrange=[-4,4], yrange=[-1,4],
                            xlabel=" ", ylabel=" ", line_width=3,
                            xtics_axis=true, ytics_axis=true,
                            xaxis_type=solid, yaxis_type=solid,
                            xaxis=true, yaxis=true,
                            xtics={["-pi",-t], ["pi",t]},
                            ytics={["pi/2",t/2], ["pi",t]},
                            explicit(f(x),x,-t,t),
                            point_size =1.5, point_type=filled_circle,
                            points([[-t,%pi/2],[t,%pi/2],[-t,0],[t,%pi]]),
                            color=white, point_size =1,
                            point_type=filled_circle, points([[-t,0],[t,%pi]]),
                            color=red, key=sconcat("k=",k),
                            explicit(s(k),x,-t,t) ), k,[1,3,5,9,13,21])))$

(%i17)  apply(draw,
            append( [ terminal='animated_gif, delay=90,
                      file_name="E:/Szereg_Fouriera_f_sk2",
                      dimensions=[900,500]],
            makelist( gr2d( proportional_axes=xy, grid=true,
                            xrange=[-2*t,4*t], yrange=[-1,4],
                            xlabel=" ", ylabel=" ", line_width=2.5,
                            xtics_axis=true, ytics_axis=true,
                            xaxis_type=solid, yaxis_type=solid,
                            xaxis=true, yaxis=true,
                            xtics={["-pi",-t], ["pi",t], ["2pi",2*t], ["3pi",3*t]},
                            ytics={["pi/2",t/2], ["pi",t]},
                            explicit(f(x),x,-t,t),
                            point_size =1.5, point_type=filled_circle,
                            points([[-t,%pi/2],[t,%pi/2],[-t,0],[t,%pi]]),
                            color=white, point_size =1,
                            point_type=filled_circle, points([[-t,0],[t,%pi]]),
                            color=red, key=sconcat("k=",k),
                            explicit(s(k),x,-2*t,4*t) ), k,[1,3,5,9,13,21])))$
```



Przykład 7

Rozwiniemy w szereg Fouriera funkcję $f(x) = |\sin x|$ dla $x \in [-\pi, \pi]$.

Skorzystamy w tym przykładzie z polecenia **totalfourier**. Zauważmy, że f jest funkcją parzystą, stąd współczynniki $b_n = 0$, dla $n \in \mathbb{N}$.

(%i1) `load(fourie)$`

(%i3) `f(x):=abs(sin(x))$ t:%pi$`

(%i4) `totalfourier(f(x),x,t);`

Współczynniki szeregu Fouriera funkcji f.

(%t4) $a_0 = \frac{2}{\pi}$

(%t5) $a_n = \frac{2\left(\frac{\cos(\pi n)}{2n+2} - \frac{\cos(\pi n)}{2n-2} + \frac{1}{2n+2} - \frac{1}{2n-2}\right)}{\pi}$

(%t6) $b_n = 0$

Uproszczone współczynniki szeregu Fouriera funkcji f.

(%t7) $a_0 = \frac{2}{\pi}$

(%t8) $a_n = -\frac{2((-1)^n + 1)}{\pi(n-1)(n+1)}$

(%t9) $b_n = 0$

Rozwinięcie funkcji f w szereg Fouriera.

(%o9) $\frac{2}{\pi} - \frac{2\sum\limits_{n=1}^{\infty}\frac{((-1)^n+1)\cos(nx)}{(n-1)(n+1)}}{\pi}$

Przykład 8

Korzystając z rozwinięcia w szereg Fouriera funkcji $f(x) = x^2$ dla $x \in [-\pi, \pi]$ wyznaczymy sumę szeregu liczbowego

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

Zauważmy, że funkcja f jest funkcją parzystą oraz spełnia warunki Dirichleta na rozważanym przedziale. Rozwiniemy ją więc w szereg Fouriera wg cosinusów.

```
(%i1) load(fourie)$
```

```
(%i3) f(x):=x^2$
      t:%pi$
```

```
(%i4) w:fourcos(f(x),x,t)$
```

(w) $a_0 = \frac{\pi^2}{3}$

(w) $a_n = \frac{2\left(\frac{\pi^2 \sin(\pi n)}{n} - \frac{2 \sin(\pi n)}{n^3} + \frac{2 \pi \cos(\pi n)}{n^2}\right)}{\pi}$

Tym razem do uproszczenia współczynników zastosujemy polecenie **foursimp**.

```
(%i6) foursimp(%)$
```

(%t6) $a_0 = \frac{\pi^2}{3}$

(%t7) $a_n = \frac{4(-1)^n}{n^2}$

```
(%i8) f(x)=fourexpand(w,x,t,inf);
```

(%o8) $x^2 = 4\left(\sum_{n=1}^{\infty} \frac{(-1)^n \cos(n x)}{n^2}\right) + \frac{\pi^2}{3}$

W szczególności dla $x = \pi$ otrzymamy równość

```
(%i9) %,x=%pi;
```

(%o9) $\pi^2 = 4\left(\sum_{n=1}^{\infty} \frac{1}{n^2}\right) + \frac{\pi^2}{3}$

a po dalszych przekształceniach - oczekiwany wynik.

```
(%i12) %-%pi^2/3;%/4$ rhs(%)=lhs(%);
```

(%o10) $$\frac{2\pi^2}{3}=4\sum_{n=1}^{\infty}\frac{1}{n^2}$$

(%o12) $$\sum_{n=1}^{\infty}\frac{1}{n^2}=\frac{\pi^2}{6}$$

Podobne rozumowanie (dla szeregów potęgowych) przeprowadziliśmy w rozdziale 12 (patrz przykład 5 d) oraz zadanie 5 c)). Również polecenie **sum** podaje pewne sumy (patrz przykład 2 c) w rozdziale 12).

Przykład 9

Korzystając z rozwinięcia w szereg Fouriera funkcji $f(x) = x$ dla $x \in [-\pi, \pi]$ wyznaczymy przybliżoną wartość liczby π.

Zauważmy, że funkcja f jest funkcją nieparzystą, zatem rozwiniemy ją w szereg Fouriera według sinusów. Do otrzymanego rozwinięcia podstawimy $x = \frac{\pi}{2}$.

```
(%i2) f(x):=x$ t:%pi$
```

```
(%i4) load(fourie)$
      w:foursin(f(x),x,t)$
```

(w) $$b_n=\frac{2\left(\frac{\sin(\pi n)}{n^2}-\frac{\pi\cos(\pi n)}{n}\right)}{\pi}$$

```
(%i8) f(x)=fourexpand(w,x,t,inf); %,x=%pi/2; solve(%,%pi)$ %[1];
```

(%o5) $$x=-2\sum_{n=1}^{\infty}\frac{(-1)^n\sin(n\,x)}{n}$$

(%o6) $$\frac{\pi}{2}=-2\sum_{n=1}^{\infty}\frac{(-1)^n\sin\left(\frac{\pi n}{2}\right)}{n}$$

(%o8) $$\pi=-4\sum_{n=1}^{\infty}\frac{(-1)^n\sin\left(\frac{\pi n}{2}\right)}{n}$$

Ponieważ $\sin\left(n\frac{\pi}{2}\right) = 0$ dla n parzystych oraz $\sin\left(n\frac{\pi}{2}\right)$ jest równe 1 albo -1 dla n nieparzystych, to powyższą sumę możemy zapisać w postaci:

```
(%i9)   %pi=4*sum((-1)^(n+1)/(2*n-1),n,1,inf);
```

(%o9) $\pi = 4\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{2n-1}$

Mimo że szereg ten jest dosyć wolno zbieżny, to jest historycznie ważny. Jest pierwszym odkrytym szeregiem, który pozwolił przybliżać liczbę π.

```
(%i12)  S[100] = 4*sum((-1)^(n+1)/(2*n-1), n,1,100), numer;
        S[1000] = 4*sum((-1)^(n+1)/(2*n-1), n,1,1000), numer;
        S[100000] = 4*sum((-1)^(n+1)/(2*n-1), n,1,100000), numer;
```

(%o10) $S_{100} = 3.131592903558554$

(%o11) $S_{1000} = 3.140592653839794$

(%o12) $S_{100000} = 3.14158265358972$

Porównajmy powyższe przybliżenia z wartością numeryczną liczby π dostępną w Maximie.

```
(%i13)  %pi,numer;
(%o13)  3.141592653589793
```

Podobnie, rozwijając funkcję $f(x) = \operatorname{arctg} x$ w szereg Maclaurina (patrz też rozdział 12), mamy

$$\operatorname{arctg} x = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)}x^{2n+1}, \; x \in [-1,1].$$

Podstawiając do obu stron $x = 1$ otrzymujemy

$$\frac{\pi}{4} = \sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)}.$$

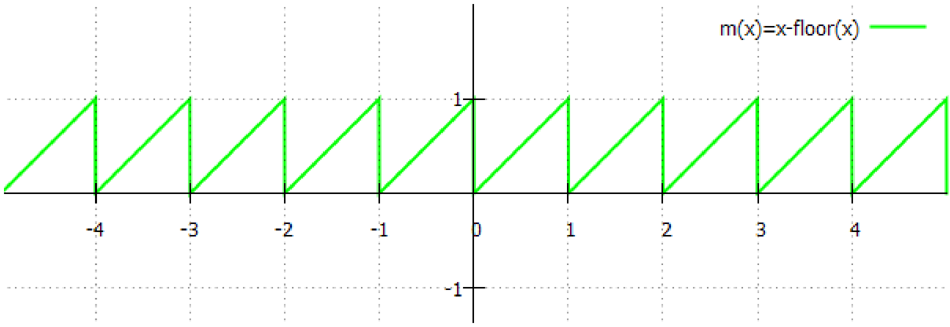
Przykład 10

Narysujemy jeszcze kilka wykresów przydatnych w wizualizacjach zjawisk falowych.

Fala (sygnał) piłokształtny o nieliniowym nachyleniu:

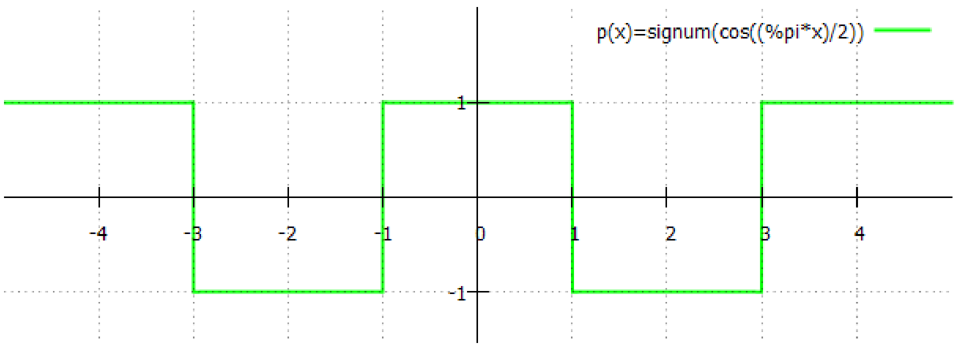
```
(%i1)   m(x):=x-floor(x)$

(%i2)   plot2d( m, [x,-5,5], [y,-2,2], [axes, true], [axes, solid], [box, false],
                [xtics,-4,1,4], [ytics,-1,2,1], [legend, sconcat("m(x)=", m(x))],
                [same_xy, true], grid2d, [style, [lines, 2, 3]])$
```

Fala o przebiegu trójkątnym (patrz przykład 2).



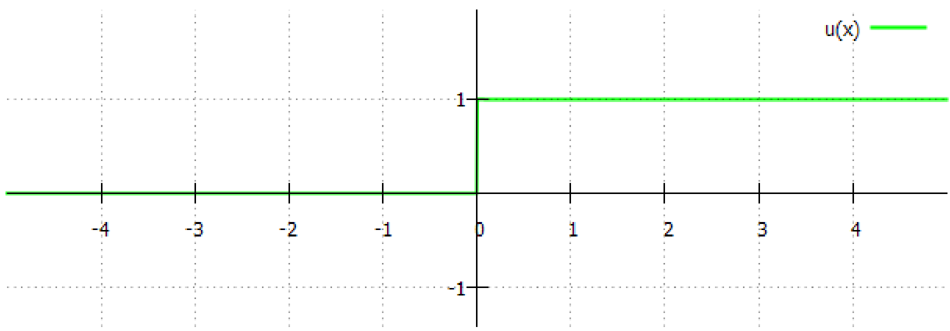
Fala (sygnał) o przebiegu prostokątnym:

```
(%i3)   p(x):=signum(cos(%pi/2*x))$

(%i4)   plot2d( p, [x,-5,5], [y,-2,2], [axes, true], [axes, solid], [box, false],
                [xtics,-4,1,4], [ytics,-1,2,1], [legend, sconcat("p(x)=", p(x))],
                [same_xy, true], grid2d, [style, [lines, 2, 2]])$
```



Skok jednostkowy:

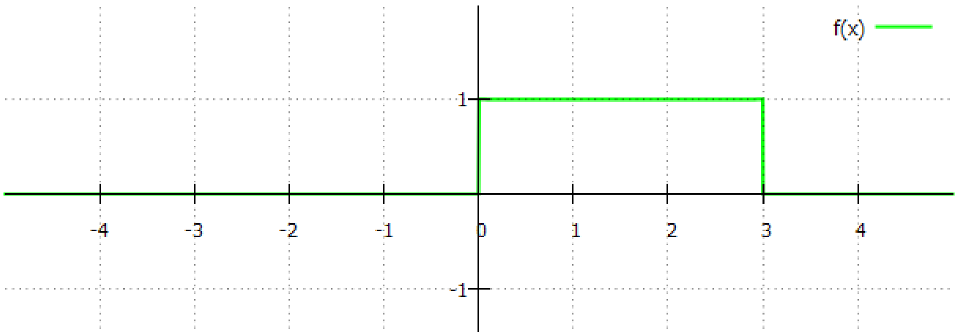
```
(%i5)   u(x):=unit_step(x)$

(%i6)   plot2d( u, [x,-5,5], [y,-2,2], [axes, true], [axes, solid], [box, false],
                [xtics,-4,1,4], [ytics,-1,2,1], [legend, "u(x)"],
                [same_xy, true], grid2d, [style, [lines, 2, 1]])$
```



Funkcje charakterystyczne zbioru:

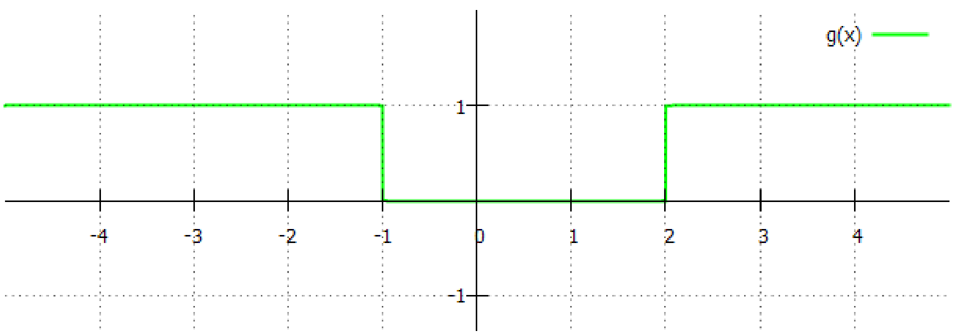
```
(%i7)   f(x):=charfun(x<3 and x>0)$
```

```
(%i8)   plot2d( f, [x,-5,5], [y,-2,2], [axes, true], [axes, solid], [box, false],
                [xtics,-4,1,4], [ytics,-1,2,1], [legend, "f(x)"],
                [same_xy, true], grid2d, [style, [lines, 2, 4]])$
```




```
(%i9)   g(x):=charfun(x>2or x<-1)$
```

```
(%i10)  plot2d( g, [x,-5,5], [y,-2,2], [axes, true], [axes, solid], [box, false],
                [xtics,-4,1,4], [ytics,-1,2,1], [legend, "g(x)"],
                [same_xy, true], grid2d, [style, [lines, 2, 3]])$
```

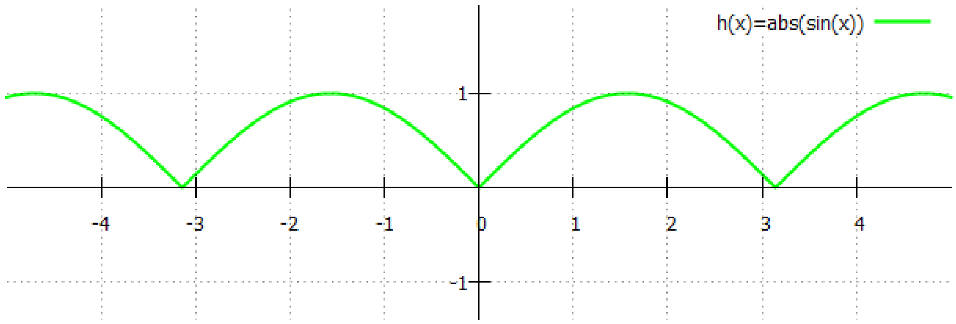


Polecenia graficzne w Maximie nie uwzględniają punktów nieciągłości (stąd pojawiły się pionowe odcinki, które nie należą do wykresu funkcji).

Wyprostowana fala sinusoidalna:

```
(%i11)  h(x):=abs(sin(x))$
```

```
(%i12)  plot2d( h, [x,-5,5], [y,-2,2], [axes, true], [axes, solid], [box, false],
                [xtics,-4,1,4], [ytics,-1,2,1], [legend, sconcat("h(x)=", h(x))],
                [same_xy, true], grid2d, [style, [lines, 2, 6]])$
```



ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Podać współczynniki rozwinięcia w szereg Fouriera funkcji $f(x) = x^2$ dla $x \in (-\pi, \pi)$.

Zadanie 2

Podać współczynniki rozwinięcia w szereg Fouriera według cosinusów funkcji $f(x) = x^2$ dla $x \in (0, \pi)$.

Zadanie 3

Podać współczynniki rozwinięcia w szereg Fouriera według sinusów funkcji $f(x) = x^2$ dla $x \in (0, \pi)$.

Zadanie 4

Rozwinąć w szereg Fouriera funkcję $f(x) = x^2$ dla $x \in [-\pi, \pi]$.

Zadanie 5

Rozwinąć w szereg Fouriera funkcję $f(x) = x^3$ dla $x \in (-\pi, \pi)$.

Zadanie 6

Rozwinąć w szereg Fouriera funkcję $f(x) = x^3$ dla $x \in (-2, 2)$.

Zadanie 7

Znaleźć rozwinięcie w szereg Fouriera funkcji $f(x) = \pi^2 - x^2$ na przedziale $[-\pi, \pi]$, a następnie wyznaczyć przybliżoną wartość $\frac{1}{3}\pi^2$.

Odpowiedzi

1. $a_0 = \frac{\pi^2}{3},\ a_n = \frac{4(-1)^n}{n^2}, b_n = 0$
2. $a_0 = \frac{\pi^2}{3},\ a_n = \frac{4(-1)^n}{n^2}$
3. $b_n = -\frac{2\left(\pi^2 n^2 (-1)^n - 2(-1)^n + 2\right)}{\pi n^3}$
4. $\frac{\pi^2}{3} + 4\sum_{n=1}^{\infty} \frac{(-1)^n}{n^2} \cos(nx)$
5. $-2\sum_{n=1}^{\infty} \frac{(n^2\pi^2 - 6)(-1)^n}{n^3} \sin(nx)$
6. $-\frac{54\sum_{n=1}^{\infty} \frac{(n^2\pi^2-6)(-1)^n \sin\left(\frac{n\pi x}{3}\right)}{n^3}}{\pi^3}$
7. $\pi^2 - x^2 = \frac{2\pi^2}{3} - 4\sum_{n=1}^{\infty} \frac{(-1)^n \cos(nx)}{n^2}, \qquad \frac{\pi^2}{3} \approx 3.289874857234324$

14. Rachunek całkowy funkcji jednej zmiennej

Tabela 14.1

POLECENIA	OPIS
integrate(f(x),x**)**	funkcja pierwotna funkcji f zmiennej x
integrate(f(x),x,a,b**)**	całka oznaczona na przedziale (a, b)
changevar(I,y=f(x),y,x**)**	zamiana zmiennej x na zmienną y przy podstawieniu $y = f(x)$ w całce I
romberg(f(x),x,a,b**)**	całka oznaczona na przedziale (a, b) obliczona numerycznie metodą Romberga[1)]
erf(x**)**	funkcja błędu[2)]
gamma_incomplete(a,z**)**	funkcja specjalna gamma[3)]
logarc:true, **logabs:**true	patrz tabela 4.6

[1)] Dotyczy całek właściwych, również wielokrotnych.

[2)] $\mathrm{erf}(x) = \frac{2}{\pi}\int_0^x e^{-t^2}dt$

[3)] $\mathrm{gamma\_incomplete}(a, z) = \int_z^\infty t^{a-1}e^{-t}dt$

ZADANIA PRZYKŁADOWE

Przykład 1

Znajdziemy kilka funkcji pierwotnych i całek oznaczonych dla funkcji całkowalnych przy zastosowaniu podstawowych metod całkowania lub metod numerycznych.

- funkcja pierwotna

```
(%i1) 'integrate(x^2,x)=integrate(x^2,x);
```

(%o1) $\int x^2 dx = \frac{x^3}{3}$

- całka oznaczona w granicach od 0 do 1

```
(%i2) 'integrate(x^2,x,0,1)=integrate(x^2,x,0,1);
```

(%o2) $\int_0^1 x^2 dx = \frac{1}{3}$

- całka z parametrem

```
(%i3) 'integrate(exp(a*x), x)=integrate(exp(a*x),x);
```

(%o3) $\int \%e^{a\,x} dx = \frac{\%e^{a\,x}}{a}$

- całka niewłaściwa (z założeniami dotyczącymi parametru)

```
(%i4) assume(a>0)$
(%i5) I:'integrate(exp(a*x), x,1,inf);
      ev(I,integrate);
      forget(a>0)$
```

(%o5) $\int_1^{\infty} \%e^{a\,x}\,dx$ **Całka rozbieżna.**

```
defint: integral is divergent.
 -- an error. To debug this try: debugmode(true);
```

```
(%i8) assume(a<0)$
(%i9) I=ev(I,integrate);
      forget(a<0)$
```

(%o9) $\int_1^{\infty} \%e^{a\,x}\,dx = -\frac{\%e^{a}}{a}$

Całka zbieżna.

Do deklaracji zakresu parametru a użyliśmy polecenia **assume**, natomiast do odwołania tej deklaracji polecenia **forget** (patrz też tabela 1.2).
Polecenie **ev** zastosowane do zadeklarowanej poniżej (symbolicznie) całki I (patrz tabela 2.2) wymagało warunku (opcji): **integrate** albo **nouns**.

- całka nieoznaczona, którą można obliczyć przez części

Patrz też przykład 3 w rozdziale 21.

```
(%i11) I:'integrate(x*log(x), x)$
(%i12) I=ev(I,nouns);
```

(%o12) $\int x\log(x)\,dx = \frac{x^2\log(x)}{2} - \frac{x^2}{4}$

```
(%i13) I=ev(I,nouns,factor);
```

(%o13) $\int x\log(x)\,dx = \frac{x^2\,(2\log(x)-1)}{4}$

- całka nieoznaczona, którą można obliczyć przez podstawienie

```
(%i14) I:'integrate(cos(log(x))/x, x);
```

(%o14) $\int \frac{\cos(\log(x))}{x}\,dx$

```
(%i15) changevar(I, y=log(x), y, x);
```

(%o15) $\int \cos(y)\,dy$

Podstawienie $y = \ln x$.

```
(%i16) ev(%,integrate);
```

(%o16) $\sin(y)$

Wyznaczenie funkcji pierwotnej.

```
(%i17) %,y=log(x);
```

(%o17) $\sin(\log(x))$

Funkcja pierwotna po powrocie do zmiennej x.

- całka oznaczona z podstawieniem

```
(%i18) I:'integrate(%e^x/(%e^(2*x)+1) ,x,0,1)$
(%i19) I=changevar(I, y=%e^x, y, x);
```

Podstawienie $y = e^x$.

(%o19) $$\int_0^1 \frac{\%e^x}{\%e^{2x}+1}dx = \int_1^{\%e} \frac{1}{y^2+1}dy$$

```
(%i20) %=ev(I,integrate);
```

(%o20) $$\int_0^1 \frac{\%e^x}{\%e^{2x}+1}dx = \int_1^{\%e} \frac{1}{y^2+1}dy = \mathrm{atan}(\%e) - \frac{\pi}{4}$$

```
(%i21) %=ev(I,integrate,bfloat,fpprec=3);
```

(%o21) $$\int_0^1 \frac{\%e^x}{\%e^{2x}+1}dx = \int_1^{\%e} \frac{1}{y^2+1}dy = \mathrm{atan}(\%e) - \frac{\pi}{4} = 4.33\mathrm{b}{-1}$$

- całki funkcji wymiernych (przekształcenia)

```
(%i22) logabs:true$
```

Ustawienie prezentacji logarytmów z modułem.

```
(%i23) I:'integrate(1/(x^2-4), x)$
(%i24) I=ev(I,integrate);
```

(%o24) $$\int \frac{1}{x^2-4}dx = \frac{\log(|x-2|)}{4} - \frac{\log(|x+2|)}{4}$$

```
(%i25) I=ev(I,integrate,factor);
```

Sprowadzenie do wspólnego mianownika.

(%o25) $$\int \frac{1}{x^2-4}dx = -\frac{\log(|x+2|)-\log(|x-2|)}{4}$$

```
(%i26) logcontract(%);
```

Przekształcenie logarytmów (zwinięcie).

(%o26) $$\int \frac{1}{x^2-4}dx = \frac{\log\left(\frac{|x-2|}{|x+2|}\right)}{4}$$

- całka z zapytaniem

```
(%i27) 'integrate(1/(k+x^2),x)=integrate(1/(k+x^2),x);
Is k positive or negative? p;
```

(%o27) $$\int \frac{1}{x^2+k}dx = \frac{\mathrm{atan}\left(\frac{x}{\sqrt{k}}\right)}{\sqrt{k}}$$

Po doprecyzowaniu informacji o zakresie zmienności stałej k można otrzymać wzory na poszczególne typy całek.

```
(%i28) 'integrate(1/(k+x^2),x)=integrate(1/(k+x^2),x),radcan,logcontract;
Is k positive or negative? n;
```

(%o28) $$\int \frac{1}{x^2+k}dx = \frac{\log\left(\frac{|x-\sqrt{-k}|}{|x+\sqrt{-k}|}\right)}{2\sqrt{-k}}$$

- całka funkcji wymiernej z rozkładem na ułamki proste

```
(%i29) w:(2*x^3)/(x^3+x^2+x+1)$
       I:'integrate(w, x);
       'integrate(partfrac(w,x),x);
       'integrate(partfrac(w,x),x),expand;
       ev(%,nouns);
```

(%o30) $$2\int \frac{x^3}{x^3+x^2+x+1}dx$$

(%o31) $$\int \frac{-x-1}{x^2+1}-\frac{1}{x+1}+2\,dx$$

(%o32) $$\int -\frac{x}{x^2+1}-\frac{1}{x^2+1}-\frac{1}{x+1}+2\,dx$$

(%o33) $$-\log(|x+1|)-\frac{\log(x^2+1)}{2}-\operatorname{atan}(x)+2x$$

- całki funkcji niewymiernych

```
(%i34) 'integrate(1/sqrt(3-x^2), x)=integrate(1/sqrt(3-x^2), x);
```

(%o34) $$\int \frac{1}{\sqrt{3-x^2}}dx = \operatorname{asin}\left(\frac{x}{\sqrt{3}}\right)$$

```
(%i35) 'integrate(1/sqrt(x^2-1), x)=integrate(1/sqrt(x^2-2), x);
```

(%o35) $$\int \frac{1}{\sqrt{x^2-1}}dx = \log\left(2\sqrt{x^2-2}+2x\right)$$

W przypadku dwóch poniżej przedstawionych całek polecenie **logarc** zamienia funkcję arcus sinus hiperboliczny na $\ln(\sqrt{x^2+1}+x)$.

```
(%i36) 'integrate(1/sqrt(x^2+1), x)=integrate(1/sqrt(x^2+1), x);
```

(%o36) $$\int \frac{1}{\sqrt{x^2+1}}dx = \operatorname{asinh}(x)$$

```
(%i37) logarc(%);
```

(%o37) $$\int \frac{1}{\sqrt{x^2+1}}dx = \log\left(\sqrt{x^2+1}+x\right)$$

```
(%i38) 'integrate(sqrt(x^2+1), x)=ev(integrate(sqrt(x^2+1), x));
```

(%o38) $\int \sqrt{x^2+1}\, dx = \frac{\operatorname{asinh}(x)}{2} + \frac{x\sqrt{x^2+1}}{2}$

```
(%i39) logarc(%);
```

(%o39) $\int \sqrt{x^2+1}\, dx = \frac{\log(\sqrt{x^2+1}+x)}{2} + \frac{x\sqrt{x^2+1}}{2}$

```
(%i40) 'integrate((-x+2)/sqrt(3+2*x-x^2), x)=integrate((-x+2)/sqrt(3+2*x-x^2), x)
```

(%o40) $\int \frac{2-x}{\sqrt{-x^2+2x+3}} dx = \sqrt{-x^2+2x+3} - \operatorname{asin}\left(\frac{2-2x}{4}\right)$

- całki funkcji trygonometrycznych

```
(%i41) I:'integrate(cos(x)^4,x)$
```

```
(%i42) I=ev(I,trigreduce);
```

(%o42) $\int \cos(x)^4 dx = \frac{\int \cos(4x) + 4\cos(2x) + 3\, dx}{8}$

```
(%i43) ev(I,integrate);%,expand;
```

(%o43) $\frac{\frac{\frac{\sin(4x)}{2}+2x}{8} + \frac{\sin(2x)}{2} + \frac{x}{2}}{2}$

(%o44) $\frac{\sin(4x)}{32} + \frac{\sin(2x)}{4} + \frac{3x}{8}$

Przy całkowaniu funkcji trygonometrycznych, aby doprowadzić wynik do najprostszej postaci, często potrzebujemy kilku przekształceń.

Uwaga 1

Istnieją takie funkcje, których nie da się scałkować podstawowymi metodami. W rozwiązaniach Maximy pojawiają się wtedy funkcje specjalne. Często możliwe jest jednak policzenie całki oznaczonej metodami numerycznymi.

- całki i funkcje specjalne

```
(%i45) 'integrate(1/log(x), x);
       integrate(1/log(x), x);
```

Całka nieoznaczona.

(%o45) $\int \frac{1}{\log(x)} dx$

(%o46) $-\text{gamma\_incomplete}(0, -\log(x))$

Definicja funkcji specjalnej w przypisie pod tabelą 14.1

```
(%i47) 'integrate(1/log(x), x,2,3);
       integrate(1/log(x), x,2,3),numer;
```

(%o47) $\int_{2}^{3} \frac{1}{\log(x)} dx$

(%o48) 1.118424814549699

Całka oznaczona, zbieżna. Wartość numeryczna całki.

```
(%i49) 'integrate(1/log(x), x,1/2,1);
       integrate(1/log(x), x,1/2,1);
```

(%o49) $\int_{\frac{1}{2}}^{1} \frac{1}{\log(x)} dx$

```
defint: integral is divergent.
 -- an error. To debug this try: debugmode(true);
```

Całka niewłaściwa, rozbieżna.

```
(%i51) 'integrate(%e^(x^2), x);
       integrate(%e^(x^2), x);
```

(%o51) $\int \%e^{x^2} dx$

(%o52) $-\frac{\sqrt{\pi}\ \%i\ \mathrm{erf}(\%i\ x)}{2}$

Całka nieoznaczona i funkcja błędu (patrz przypis po tabeli 14.1)

```
(%i53) 'integrate(%e^(x^2), x,0,1);
       romberg(%e^(x^2), x,0,1);
```

(%o53) $\int_{0}^{1} \%e^{x^2} dx$

(%o54) 1.462651757343228

Całka oznaczona. Zastosowanie polecenia **romberg** daje od razu wynik przybliżony.

```
(%i55) 'integrate(exp(x)/x, x);
       integrate(exp(x)/x, x);
```

(%o55) $\int \frac{\%e^{x}}{x} dx$

(%o56) $-\mathrm{gamma\_incomplete}(0, -x)$

Całka nieoznaczona i wynik z funkcją specjalną.

```
(%i57) 'integrate(exp(x)/x, x,1,2);
       romberg(exp(x)/x, x,1,2);
```

(%o57) $\int_{1}^{2} \frac{\%e^{x}}{x} dx$

(%o58) 3.059117265024728

Całka oznaczona z zastosowaniem metody numerycznej.

```
(%i59) 'integrate(x/sin(x), x);
       romberg(x/sin(x), x,1,2);
```

(%o59) $\int \frac{x}{\sin(x)} dx$

(%o60) 1.564647801696614

Postać wyniku dla polecenia **integrate** jest zbyt skomplikowana, by ją w tym miejscu wyświetlić.

Przykład 2

Zbadamy zbieżność kilku całek niewłaściwych.

Przedstawione poniżej całki I1, I2, I3 oraz I4, to całki niewłaściwe I rodzaju, natomiast całki I5, I6 są całkami niewłaściwymi II rodzaju.

```
(%i1) I1:'integrate(1/x, x, 1, inf);
      ev(I1,integrate);
```

(%o1) $$\int_1^{\infty} \frac{1}{x}\,dx$$

```
defint: integral is divergent.
 -- an error. To debug this try: debugmode(true);
```

```
(%i3) I2:'integrate(1/(x^2), x, 1, inf)$
      I2=ev(I2,integrate);
```

(%o4) $$\int_1^{\infty} \frac{1}{x^2}\,dx = 1$$

```
(%i5) I3:'integrate(1/(4+x^2), x, minf, inf)$
      I3=ev(I3,integrate);
```

(%o6) $$\int_{-\infty}^{\infty} \frac{1}{x^2+4}\,dx = \frac{\pi}{2}$$

```
(%i7) I4:'integrate(exp(-x^2), x, minf, inf)$
      I4=ev(I4,integrate);
```

(%o8) $$\int_{-\infty}^{\infty} \%e^{-x^2}\,dx = \sqrt{\pi}$$

```
(%i9) I5:'integrate(1/x, x, 0, 1);
      ev(I5,integrate);
```

(%o9) $$\int_0^{1} \frac{1}{x}\,dx$$

```
defint: integral is divergent.
 -- an error. To debug this try: debugmode(true);
```

```
(%i11) I6:'integrate(1/(x-1)^(1/3), x, 1, 9)$
       I6=ev(I6,integrate);
```

(%o12) $$\int_1^{9} \frac{1}{(x-1)^{1/3}}\,dx = 6$$

Przykład 3

Zbadamy zbieżność całki niewłaściwej $\int_0^1 x \ln x \, dx$, obserwując kolejne etapy obliczeń. Podamy też ilustrację graficzną rozważanego zagadnienia.

```
(%i1) f:x*log(x)$
      'integrate(f, x)=integrate(f, x);
```

(%o2) $\int x\log(x)\,dx = \frac{x^2\log(x)}{2} - \frac{x^2}{4}$

```
(%i3) assume(alpha>0,alpha<1);
```

(%o3) $[\alpha > 0, \alpha < 1]$

```
(%i4) I:'integrate(f, x, alpha, 1);
```

(%o4) $\int_{\alpha}^{1} x\log(x)\,dx$

```
(%i5) 'limit(I, alpha, 0, plus);
```

(%o5) $\lim\limits_{\alpha \to 0+} \int_{\alpha}^{1} x\log(x)\,dx$

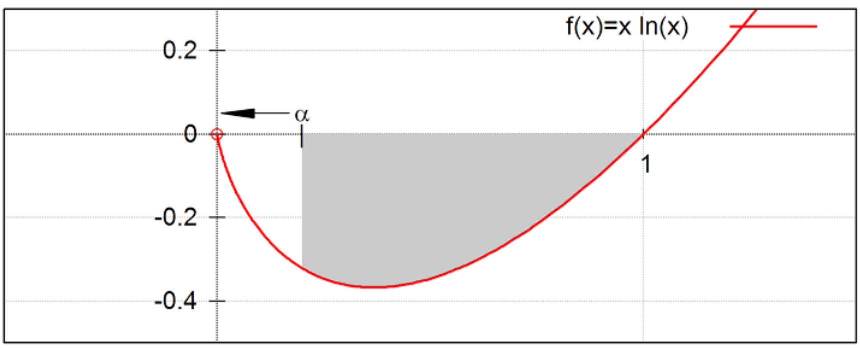
```
(%i6) ev(%,integrate);
```

(%o6) $\lim\limits_{\alpha \to 0+} -\frac{2\alpha^2\log(\alpha)-\alpha^2}{4} - \frac{1}{4}$

```
(%i7) ev(%,limit);
```

(%o7) $-\frac{1}{4}$

```
(%i8) load(draw)$ draw2d( dimensions = [600,400],
      xrange = [-0.5,1.5], yrange = [-0.5,0.3], grid = true,
      xaxis = true, xtics_axis = true, xtics = {1},
      yaxis = true, ytics_axis = true, ytics = 0.2, color = black,
      head_length = 0.3, head_angle = 4, vector([.17,0.05],[-0.16,0]),
      fill_color = gray80, filled_func = 0, explicit(f,x,0.2,1), filled_func = false,
      color = red, point_type = 6, point_size = 0.9, points([0],[0]),
      line_width = 2, key = "f(x)=x ln(x)   ", explicit(f,x,0,2),
      color = black, font = "Arial", font_size = 16,
      label(["{/Symbol a}",.2,0.05],["|",.2,0])   )$
```



Przykład 4

Zbadamy zbieżność całki $\int_1^{\infty} \frac{1}{x^{\alpha}}\, dx$ w zależności od parametru α, gdy α jest liczbą rzeczywistą dodatnią.

```
(%i1) kill(all)$
      assume(alpha>1);
(%o1) [α>1]
```

```
(%i2) 'integrate(1/(x^alpha), x,1,inf)=integrate(1/(x^alpha), x,1,inf);
```

(%o2) $\int_1^{\infty} \frac{1}{x^{\alpha}} dx = \frac{1}{\alpha - 1}$

```
(%i3) forget(alpha>1)$ assume(alpha>0,alpha<1);
(%o4) [α>0,α<1]
```

```
(%i5) 'integrate(1/(x^alpha), x,1,inf); integrate(1/(x^alpha), x,1,inf);
```

(%o5) $\int_1^{\infty} \frac{1}{x^{\alpha}} dx$

```
defint: integral is divergent.
 -- an error. To debug this try: debugmode(true);
```

```
(%i7) ev(%, alpha=1);ev(%, integrate);
```

(%o7) $\int_1^{\infty} \frac{1}{x} dx$

```
defint: integral is divergent.
 -- an error. To debug this try: debugmode(true);
```

Zatem całka niewłaściwa

$$\int_1^{\infty} \frac{1}{x^{\alpha}}\, dx\,,$$

podobnie jak szereg

$$\sum_{n=1}^{\infty} \frac{1}{n^{\alpha}}$$

jest zbieżna dla $\alpha > 1$, natomiast rozbieżna dla $0 < \alpha \leq 1$.

Przykład 5

Wyznaczymy pole figury ograniczonej krzywymi:

a) $y = \ln x,\ \ y = 0,\ x = e;$ b) $y = -x^2,\ \ y = x,\ \ x = 1;$

c) $y = 3x - x^2,\ \ y = x - 1.$

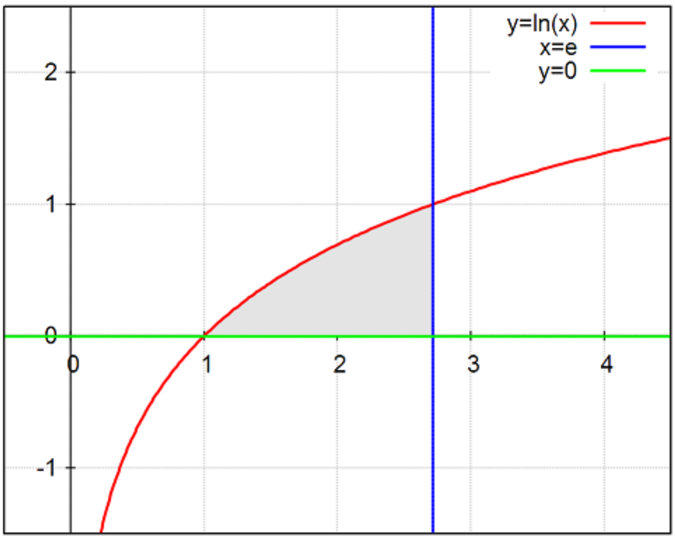
Najpierw narysujemy krzywe i zaznaczymy szukane figury, a następnie obliczymy odpowiednie całki.

W celu wykonania rysunków wczytamy pakiet **draw**, a następnie, stosując polecenie **set_draw_defaults**, ustawimy wartości domyślne dla instrukcji **draw2d**. Więcej opcji (wraz z opisem) zamieszczonych jest w tabelach rozdziału 6.2.

```
(%i1) load(draw)$
      set_draw_defaults(
          line_width = 2, grid = true,  proportional_axes = xy,
          xaxis = true, xaxis_type = solid, xtics_axis = true, xtics = 1,
          yaxis = true, yaxis_type = solid, ytics_axis = true, ytics = 1,
          fill_color = gray90, font = "Arial", font_size = 13 )$
```

a)

```
(%i3) draw2d(
      xrange = [-0.5,4.5], yrange = [-1.5,2.5],
      filled_func = 0, explicit(log(x),x,1,%e),
      color = red, key = "y=ln(x)", filled_func = false, explicit(log(x),x,0,4.5),
      color=blue, key = "x=e", implicit(x=%e,x,2,3,y,-1.5,2.5),
      color=green, key = "y=0", implicit(y=0,x,-0.5,4.5,y,-1.5,2.5) )$
rat: replaced -2.718281828459045 by -28245729/10391023 = -2.718281828459046
```

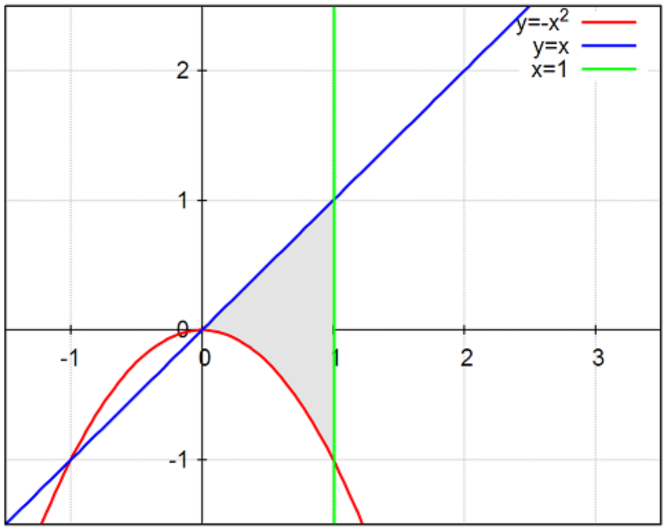



```
(%i4) 'integrate(log(x), x, 1, %e)=integrate(log(x), x, 1, %e);
```

(%o4) $\int_{1}^{\%e} \log(x)\,\mathrm{d}x = 1$

b)

```
(%i5) draw2d(
      xrange = [-1.5,3.5], yrange = [-1.5,2.5],
      filled_func = x, explicit(-x^2,x,0,1),
      color = red, key = "y=-x^2", filled_func = false, explicit(-x^2,x,-1.5,2.5),
      color = blue, key = "y=x", explicit(x,x,-1.5,3.5),
      color = green, key = "x=1", implicit(x=1,x,-1.5,3.5,y,-1.5,2.5) )$
```




```
(%i6) 'integrate(x+x^2, x, 0, 1)=integrate(x+x^2, x, 0, 1);
```

(%o6) $\int_0^1 x^2+x\,dx=\frac{5}{6}$

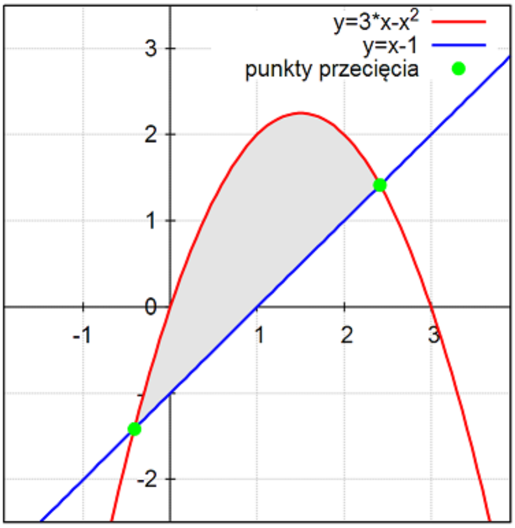
c)

Najpierw zdefiniujemy funkcje f i g, by dalej móc się do nich odwoływać. Następnie, korzystając z polecenia **solve**, wyznaczymy pierwiastki równania $f(x) = g(x)$. Polecenia **map** i **rhs** pozwolą utworzyć listę pierwiastków, a komenda **sort** z opcją „<" uporządkuje je rosnąco. W końcu, otrzymane pierwiastki przypiszemy do zmiennych $x1$ oraz $x2$.
Budujemy więc pewien szablon, w którym wystarczy jedynie zmienić definicje funkcji f i g, by otrzymać rozwiązanie nowego zadania.

```
(%i7) f(x):= -x^2+3*x$
      g(x):=  x-1$
      [x1, x2]: sort (map('rhs, solve(f(x)=g(x))), "<");
```

(%o9) $[1-\sqrt{2},\sqrt{2}+1]$

```
(%i10) draw2d(
       xrange = [x1-1.5,x2+1.5], yrange = [-2.5,3.5],
       filled_func = g(x), explicit(f(x),x,x1,x2),
       filled_func = false,
       color = red, key = sconcat("y=", f(x)), explicit(f(x),x,x1-2,x2+2),
       color = blue, key = sconcat("y=", g(x)), explicit(g(x),x,x1-2,x2+2),
       color = green, key = "punkty przecięcia", point_type = 7,
       points([x1,x2], [f(x1),f(x2)])  )$
```



(%i11) I:'integrate('f(x)-'g(x), x, 'x1, 'x2);

(%o11) $$\int_{x1}^{x2} f(x)-g(x)\,dx$$

(%i12) ev(I,f,g);

(%o12) $$\int_{1-\sqrt{2}}^{\sqrt{2}+1} -x^2+2x+1\,dx$$

(%i13) ev(I,f,g,integrate);

(%o13) $$\frac{2^{5/2}+5}{3}+\frac{2^{5/2}-5}{3}$$

(%i14) ev(%,radcan);

(%o14) $$\frac{2^{7/2}}{3}$$

Przykład 6

Obliczymy długość łuku krzywej $y = 2\ln(4 - x^2)$, $0 \le x \le 1$.

W przykładzie tym pokażemy, że na rozwiązania generowane przez Maximę trzeba też czasem spojrzeć krytycznym okiem.

Przypomnijmy, że długość łuku krzywej $y = f(x)$, $a \le x \le b$, obliczamy na podstawie wzoru:

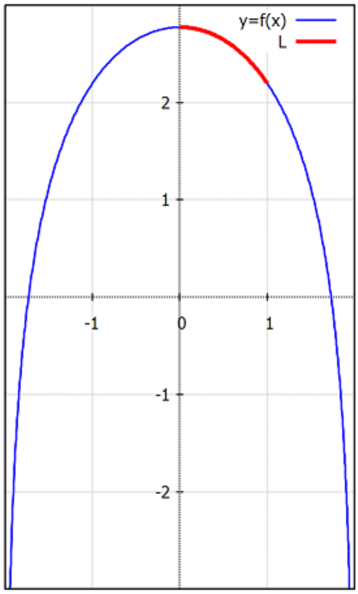
(%i1) "|L|"='integrate(sqrt(1+diff(f(x),x)^2),x,a,b);

(%o1) $$|L| = \int_a^b \sqrt{\left(\frac{d}{dx} f(x)\right)^2+1}\,dx$$

Przedstawimy najpierw fragment wykresu rozważanej krzywej z zaznaczonym na niej łukiem L. Podobnie jak w poprzednich przykładach, użyjemy pakietu **draw**.

```
(%i2) f(x):=2*log(4-x^2)$
```

```
(%i3) load(draw)$
      draw2d(grid = true,
      xrange = [-2,2],
      yrange = [-3,3],
      proportional_axes = xy,
      xaxis = true,
      yaxis = true,
      xtics = {-1,0,1},
      ytics = {-2,-1,1,2},
      xtics_axis = true,
      ytics_axis = true,
      line_width = 2,
      key = "y=f(x)",
      explicit(f,x,-2,2),
      color = red,
      line_width = 3,
      key = "L",
      explicit(f,x,0,1))$
```



Dla funkcji $f(x) = 2\ln(4 - x^2)$ oraz przedziału $[0,1]$ wzór (%o1) przyjmuje postać:

```
(%i5) L:'integrate(sqrt(1+diff(f(x),x)^2), x ,0, 1);
```

(%o5) $\int_0^1 \sqrt{\frac{16\,x^2}{(4-x^2)^2}+1}\,dx$

Aby uprościć wyrażenie pod całką, zastosujemy polecenie **ev** z opcją **radcan** (patrz też tabela 2.2 oraz tabela 4.5).

```
(%i6) ev(L,radcan);
```

(%o6) $\int_0^1 \frac{x^2+4}{x^2-4}\,dx$

Scałkujemy następnie uproszczoną funkcję i wyznaczymy wartość przybliżoną wyniku.

```
(%i7) ev(%,integrate);%,numer;
```

(%o7) $1-2\log(3)$

(%o8) -1.197224577336219

Okazuje się, że otrzymaliśmy wynik ujemny, co nie powinno mieć miejsca, gdy wyznaczamy długość łuku krzywej.

Sprawdźmy, na którym etapie obliczeń mógł się pojawić błąd.

Jeśli całkowanie wykonamy bezpośrednio (bez przekształceń), to otrzymamy:

```
(%i9) ev(L,integrate);%,numer;
(%o9) 2 log(3)-1
(%o10) 1.197224577336219
```

Tym razem wydaje się, że wynik powinien być poprawny, co może sugerować błąd generowany przez polecenie **radcan**. Sprawdźmy więc „krok po kroku” przekształcenia pod całką:

```
(%i11) diff(f(x),x);
       1+diff(f(x),x)^2;
       1+diff(f(x),x)^2,factor;
       sqrt(%);
```

(%o11) $-\frac{4\,x}{4-x^2}$

(%o12) $\frac{16\,x^2}{(4-x^2)^2}+1$

(%o13) $\frac{(x^2+4)^2}{(x-2)^2\,(x+2)^2}$

(%o14) $\frac{x^2+4}{|x-2|\,|x+2|}$

Pierwiastkując (%o13), otrzymujemy wyrażenie (%o14) z wartością bezwzględną. Natomiast w (%o6), gdzie zastosowane było polecenie **radcan**, moduły zostały pominięte bez uwzględnienia zakresu zmiennej x.

Wprowadzimy w takim razie deklarację dla zmiennej x:

```
(%i15) assume(x>=0,x<=1)$
```

Teraz otrzymujemy:

```
(%i16) 1+diff(f(x),x)^2,factor;
       sqrt(%);
       partfrac(%,x);
       'integrate(%,x);
```

(%o16) $\frac{(x^2+4)^2}{(x-2)^2\,(x+2)^2}$

(%o17) $\frac{x^2+4}{(2-x)\,(x+2)}$

Tym razem moduły zostały zastąpione poprawnie.

(%o18) $\frac{2}{x+2}-\frac{2}{x-2}-1$

Rozkład (%o17) na ułamki proste.

(%o19) $\int \frac{2}{x+2}-\frac{2}{x-2}-1\,dx$

Całka nieoznaczona zapisana symbolicznie.

Wyznaczymy (zdefiniujemy) teraz funkcję pierwotną dla całki (%o19):

```
(%i20) F(x):=''(ev(%,integrate));
       F(1)-F(0);
(%o20) F(x):=2 log(x+2)-2 log(x-2)-x
(%o21) 2 log(3)-2 log(2)-2 log(-1)+2 log(-2)-1
```

Pojawił się tu kolejny problem, gdyż Maxima „nie pamięta” o modułach przy całkowaniu funkcji wymiernych. Dodamy więc opcję **logabs** w poleceniu **ev**, która ma takie działanie jak deklaracja **logabs:true** (patrz przykład 1).

```
(%i22) F(x):=''(ev(%o19,integrate,logabs));
       F(1)-F(0);
(%o22) F(x):=2 log(x+2)-x-2 log(2-x)
(%o23) 2 log(3)-1
```

Powyższy wynik oraz wynik (%o9) są poprawne.

Przykład 7

Zdefiniujemy funkcję

$$F(x) = \int_1^x t^3\, dt\,, \quad x \in [1{,}5]$$

i wyznaczymy jej pochodną.

Funkcja F jest funkcją górnej granicy całkowania.

Funkcja podcałkowa jest ciągła na rozważanym przedziale, więc funkcja F jest na tym przedziale różniczkowalna i zachodzi równość $F'(x) = x^3$.

```
(%i1) F(x):='integrate(t^3,t,1,x);
```
$$\text{(\%o1) } F(x) := \int_1^x t^3\, dt$$

W definicji funkcji F „zatrzymujemy” wykonanie operacji całkowania.

```
(%i2) 'diff(F(x),x)=diff(F(x),x);
```
$$\text{(\%o2) } \frac{d}{d\,x}\int_1^x t^3\, dt = x^3$$

Zapis symboliczny i wynik obliczania pochodnej. Równość zachodzi oczywiście na rozważanym przedziale [1,5].

```
(%i3) 'F(2)=F(2);
```
$$\text{(\%o3) } F(2) = \int_1^2 t^3\, dt$$

Wartość funkcji wyrażona całką.

```
(%i4) F(x),integrate;
```
$$\text{(\%o4) } \frac{x^4}{4} - \frac{1}{4}$$

Funkcja F po wykonaniu operacji całkowania.

```
(%i5) 'F(2)=F(2),integrate;
```
$$\text{(\%o5) } F(2) = \frac{15}{4}$$

Wartość funkcji po scałkowaniu.

Przykład 8

Dystrybuanta standardowego rozkładu normalnego jako szczególny przypadek funkcji górnej granicy całkowania.

Przypomnijmy, że gęstość prawdopodobieństwa zmiennej losowej o standardowym rozkładzie normalnym dla $t \in \mathbb{R}$ dana jest wzorem:

```
(%i1) f(t):=1/sqrt(2*%pi)*%e^(-t^2/2);
```

(%o1) $\mathrm{f}(t):=\frac{1}{\sqrt{2\pi}}\%e^{-\frac{t^2}{2}}$

Funkcja f jako gęstość zmiennej losowej typu ciągłego ma własności:

```
(%i2) 'f(t)>=0;
      f(t)>=0,pred;
```

(%o2) $\mathrm{f}(t)>=0$

(%o3) true

- jest funkcją nieujemną dla $t \in \mathbb{R}$

```
(%i4) 'integrate('f(t),t,minf,inf)=1;
      integrate(f(t),t,minf,inf)=1,pred;
```

(%o4) $\int_{-\infty}^{\infty}\mathrm{f}(t)\,dt=1$

(%o5) true

- spełnia tzw. „warunek unormowania”

Dla zmiennej losowej ciągłej o gęstości f dystrybuanta F wyraża się wzorem: $F(x) = \int_{-\infty}^{x} f(t)\,dt\,, x \in \mathbb{R}$. W tym przypadku (dla rozkładu normalnego) nie jest to funkcja elementarna.

```
(%i6) F(x):=1/sqrt(2*%pi)*integrate(%e^(-t^2/2),t,minf,x);
```

(%o6) $\mathrm{F}(x):=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\%e^{-\frac{t^2}{2}}\,dt$

```
(%i7) ''(F(x)),ratsimp;
```

(%o7) $\frac{\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)+1}{2}$

Po uproszczeniu otrzymaliśmy funkcję specjalną **erf** (funkcję błędu).

Za pomocą funkcji gęstości lub dystrybuanty możemy wyznaczyć prawdopodobieństwa. Na przykład dla standardowego rozkładu normalnego mamy

$$P(0 < X < 1) = \int_0^1 f(x)\,dx = F(1) - F(0) \cong 0.341, \text{ gdyż}$$

```
(%i8) "F(1)-F(0)"=ev(F(1)-F(0),bfloat,fpprec=5);
      "F(1)-F(0)"=romberg(f(x),0,1);
```

(%o8) *F(1)-F(0)* =3.4134b-1

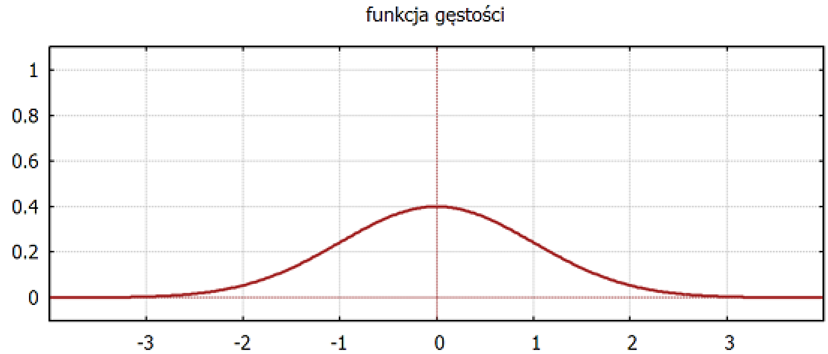
(%o9) *F(1)-F(0)* =0.34134473402459

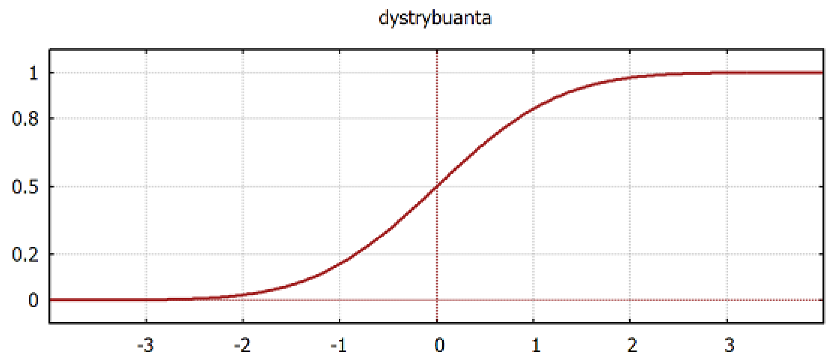
(%o8) oraz (%o9) podają wyniki dla dwóch różnych metod obliczeń.

Porównajmy wykresy funkcji gęstości i dystrybuanty rozważanej zmiennej losowej. Wykresy te zostały zamieszczone w jednym oknie i mają wspólne ustawienia **set_draw_defaults**. Więcej informacji na temat poleceń pakietu **draw** można znaleźć w rozdziale 6.2.

```
(%i10) load(draw)$
       set_draw_defaults(grid = true,
       xaxis = true, xaxis_color = brown, yaxis = true, yaxis_color = brown,
       xtics = [-3,1,3], yrange = [-0.1,1.1], line_width = 2, color = brown)$
```

```
(%i12) gp:gr2d(title="funkcja gęstości",explicit(f(x),x,-4,4),ytics=[0,0.2,1])$
       dyst:gr2d(title="dystrybuanta",explicit(F(x),x,-4,4),ytics={0,0.2,0.5,0.8,1})$
       draw(columns=1,dimensions=[600,600],gp,dyst)$
```





Uwaga 2

Wszystkie istotne funkcje i charakterystyki liczbowe związane z rozkładami prawdopodobieństwa (zarówno dla zmiennych skokowych jak i ciągłych) są dostępne po wczytaniu pakietu **distrib**. Są to m.in. funkcja gęstości (funkcja prawdopodobieństwa) **- pdf_***, dystrybuanta **- cdf_***, kwantyl **- quantile_***, wartość przeciętna **– mean_*** oraz inne charakterystyki liczbowe. W powyższych nazwach symbol ***** jest zastępowany przez nazwę rozkładu. Rozkład normalny standardowy ma nazwę **normal** oraz parametry **0** i **1.**

Poniżej kilka przykładów użycia poleceń tego pakietu:

```
(%i15) load(distrib)$ fpprintprec:6$

(%i17) "f(3)"=pdf_normal(3,0,1),float;
       "P(X<1)=F(1)"=cdf_normal(1,0,1),float;
       u[0.05]=quantile_normal(5/100,0,1),float;
(%o17) f(3) =0.004432                    - wartość funkcji gęstości
(%o18) P(X<1)=F(1) =0.8413               - wartość dystrybuanty
(%o19) u_0.05 = -1.64485                 - kwantyl rzędu 0.05
```

Na koniec ilustracja rachunkowa „reguły trzech sigm”:

```
(%i20) "P(-3<X<3)"=romberg(pdf_normal(t,0,1),t,-3,3);
(%o20) P(-3<X<3) =0.9973

(%i21) "P(-3<X<3)"=cdf_normal(3,0,1)-cdf_normal(-3,0,1),float;
(%o21) P(-3<X<3) =0.9973
```

Przykład 9

a) Dana jest funkcja kosztu krańcowego $K'(x) = x^2 + 2x - 9$. Wyznaczymy funkcję kosztu całkowitego $K(x)$ wiedząc, że koszty stałe wynoszą 15 jednostek pieniężnych.

```
(%i1) Kp(x):=x^2+2*x-9$

(%i2) K(x,C):=''(integrate(Kp(x), x)+C);
```

(%o2) $K(x,C) := C + \frac{x^3}{3} + x^2 - 9x$

```
(%i3) K(0,C);
(%o3) C
```

Mamy $K(0) = C$. Ponieważ koszty stałe są równe 15, więc $K(0) = 15$.
Stąd C=15.

```
(%i4) K(x,15);
```

(%o4) $\frac{x^3}{3} + x^2 - 9x + 15$

Tym samym $K(x) = \frac{x^3}{3} + x^2 - 9x + 15$.

b) Do magazynu dostarczany jest towar. Dostawa rozpoczyna się o godzinie 8, i w chwili t (licząc od rozpoczęcia dostawy) jej natężenie wynosi

$$f(t) = 1 + \frac{t}{4 + t^2}$$

jednostek towaru na sekundę. Obliczymy, ile towaru napłynie do magazynu między godziną 9 a 11.

Licząc godziny od rozpoczęcia dostawy, badamy wielkość dostawy od chwili $T_1 = 1$ do chwili $T_2 = 3$. Natężenie dostawy na godzinę jest 3600 razy większe niż na sekundę. Ilość towaru, jaka napłynie w tym czasie do magazynu, to odpowiednia całka oznaczona. Do jej obliczenia zastosujemy polecenie **romberg** (patrz tabela 14.1).

```
(%i5) f(t):=(1+t/(4+t^2))*3600;
```

(%o5) $f(t) := \left(1 + \frac{t}{4+t^2}\right) 3600$

```
(%i6) 'integrate(f(t), t, 1, 3)=romberg(f(t), t, 1, 3);
```

(%o6) $3600 \int_1^3 \frac{t}{t^2+4} + 1\, dt = 8919.9$

Do magazynu napłynie 8919 jednostek towaru.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Obliczyć całki nieoznaczone:

a) $\int \ln^2 x\, dx$, b) $\int e^x \sin x\, dx$, c) $\int (x+1)(x-2)^4\, dx$, d) $\int \frac{1}{\sqrt{4-x^2}} dx$,

e) $\int \frac{1}{\sqrt{4x-x^2}} dx$, f) $\int \frac{x^4}{x^2-5x+6} dx$, g) $\int \frac{x}{x^2+x+1} dx$, h) $\int \mathrm{tg}^4 x\, dx$.

Zadanie 2

Obliczyć całki oznaczone:

a) $\int_1^e x^2 \ln^2 x\, dx$, b) $\int_{-1}^0 \frac{x-3}{\sqrt{1-2x-x^2}} dx$, c) $\int_0^{\frac{\pi}{2}} \sin^2 x\, dx$, d) $\int_0^1 x\, e^{-x}\, dx$.

Zadanie 3

Zbadać zbieżność całki:

a) $\int_{-\infty}^0 x^3 e^{-x^2} dx$, b) $\int_0^2 \frac{1}{(x-2)^3} dx$, c) $\int_0^1 x\, e^{1/x} dx$, d) $\int_0^\infty \frac{1}{x^2-2x+6} dx$.

Zadanie 4

Obliczyć pole figury ograniczonej krzywymi:

a) $y = 2x - x^2,\ x + y = 0$;

b) $y = x \ln x,\ y = 0,\ x = 1,\ x = e$;

c) $y = xe^x,\ y = e - e(x-1)^2$;

d) $y = -x^2,\ y = 2x,\ y = x$.

Zadanie 5

a) Obliczyć objętość bryły powstałej przez obrót wokół osi Ox figury

$$D = \{(x, y) \in \mathbb{R}^2:\ 0 \le x \le \pi,\ \ 0 \le y \le \ \sin x\}.$$

b) Oblicz pole powierzchni powstałej z obrotu wokół osi Ox krzywej

$$y = \sqrt{9 - x^2},\quad -1 \le x \le 1.$$

c) Obliczyć długość łuku krzywej $y = 2\sqrt{x^3},\ \ 0 \le x \le 11.$

Zadanie 6

Wyznaczyć pochodną funkcji górnej granicy całkowania:

a) $\int_1^x \sqrt{2 + t^4}\, dx$, b) $\int_x^1 e^{-t^2/2}\, dx$.

Zadanie 7

a) Dana jest funkcja kosztu krańcowego $K'(x) = -10e^{-x}$. Wyznaczyć funkcję kosztu całkowitego $K(x)$, wiedząc, że koszty stałe wynoszą 50 jednostek pieniężnych.

b) Do magazynu dostarczany jest towar. Dostawa rozpoczyna się o godzinie 13 i w chwili t (licząc od rozpoczęcia dostawy) jej natężenie wynosi $f(t) = \frac{1+2t}{1+t^2}$ jednostek towaru na sekundę. Obliczyć, ile towaru napłynie do magazynu między godziną 13, a 14.

Odpowiedzi

1. a) $x(\ln^2 x - 2\ln x + 2)$, b) $\frac{e^x(\sin x - \cos x)}{2}$, c) $\frac{5x^6 - 42x^5 + 120x^4 - 80x^3 - 240x^2 + 480x}{30}$, d) $\arcsin\frac{x}{2}$, e) $\ln\left(2\sqrt{x^2+4x} + 2x + 4\right)$, f) $\frac{2x^3+15x^2+114x}{6} + \ln\frac{(x-3)^{81}}{(x-2)^{16}}$, g) $\ln\frac{(x^2+x+1)}{2} - \frac{1}{\sqrt{3}}\operatorname{arctg}\frac{2x+1}{\sqrt{3}}$, h) $\frac{\operatorname{tg}^3 x - 3\operatorname{tg} x}{3} + x$.
2. a) $\frac{1}{27}(5e^3 - 2)$, b) $-\pi + \sqrt{2} - 1$, c) $\frac{\pi}{4}$, d) $1 - \frac{2}{e}$.
3. a) zbieżna: $-\frac{1}{2}$, b) rozbieżna, c) rozbieżna, d) zbieżna: $\frac{\sqrt{5}}{5}\pi - \operatorname{arctg}\sqrt{5}$.
4. a) $\frac{9}{2}$, b) $\frac{e^2+1}{4}$, c) $\frac{2e-3}{3}$, d) $\frac{7}{6}$.
5. a) $\frac{\pi}{2}$, b) $12\,\pi$, c) 74.
6. a) $\sqrt{2+x^4}$, b) $-e^{-x^2/2}$.
7. a) $K(x) = 10e^{-x} + 40$, b) 5323 jednostki towaru.

15. Rachunek różniczkowy funkcji wielu zmiennych

Pochodną cząstkową funkcji f względem zmiennej x w punkcie (x_0, y_0) oznaczamy symbolem

$$f'_x(x_0, y_0) \text{ lub } \frac{\partial f}{\partial x}(x_0, y_0) \text{ lub } \frac{\partial}{\partial x} f(x, y)\Big|_{x=x_0, y=y_0}.$$

Pochodną cząstkową funkcji f względem zmiennej y w punkcie (x_0, y_0) oznaczamy symbolem

$$f'_y(x_0, y_0) \text{ lub } \frac{\partial f}{\partial y}(x_0, y_0) \text{ lub } \frac{\partial}{\partial y} f(x, y)\Big|_{x=x_0, y=y_0}.$$

Dla pochodnych cząstkowych drugiego rzędu funkcji f stosujemy oznaczenia:

$$(f'_x)'_x = f''_{xx},\ (f'_x)'_y = f''_{xy},\ \left(f'_y\right)'_x = f''_{yx},\ \left(f'_y\right)'_y = f''_{yy}$$

lub inaczej odpowiednio

$$\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial x^2},\ \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial y \partial x},\ \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) = \frac{\partial^2 f}{\partial x \partial y},\ \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) = \frac{\partial^2 f}{\partial y^2}.$$

Tabela 15.1

POLECENIA	OPIS
diff(f(x1,x2,…,xk),xi**)**	pochodna cząstkowa rzędu pierwszego funkcji f względem zmiennej x_i, $i = 1,2, \dots, k$
diff(f(x1,x2,…,xk),xi,n**)**	pochodna cząstkowa rzędu n-tego funkcji f względem zmiennej x_i, gdzie $i = 1,2, \dots, k$
diff(f,xi,n,xj,m,…**)**	pochodna cząstkowa mieszana
derivabbrev: true	zmiana notacji pochodnych (domyślnie: false)
at(diff(…),[x=x0,y=y0]**)**	wartość pochodnej funkcji f w punkcie (x_0, y_0)
hessian(f(x,y),[x,y]**)**	hesjan funkcji f dwóch zmiennych [1)]
jacobian(f(x,y),[x,y]**)**	w tym przypadku[2)] otrzymamy macierz, której jedyny wiersz jest gradientem funkcji f [3)]

[1)] Hesjanem (macierzą Hessego) nazywamy macierz drugich pochodnych cząstkowych funkcji (patrz też tabela 8.5).

[2)] Więcej informacji na temat funkcji **jacobian** znajduje się w rozdziale 8.4.

[3)] Gradient jest tu rozumiany jako wektor (lista) pochodnych cząstkowych pierwszego rzędu. Gradient funkcji w punkcie wskazuje kierunek najszybszego wzrostu jej wartości.

ZADANIA PRZYKŁADOWE

Przykład 1

Obliczymy wartości pierwszych pochodnych cząstkowych funkcji

$$f(x, y) = x^3 y^2 + 2 \operatorname{arctg}(3x + 4y) \text{ w punkcie } (-1,0).$$

Dziedziną funkcji f jest zbiór $D=\mathbb{R}^2$.

```
(%i1) f(x,y):= x^3 *y^2+2*atan(3*x+4*y)$
```

```
(%i2) 'diff('f(x,y),x);
      'diff(f(x,y),x);
      diff(f(x,y),x);
      %,x=-1,y=0;
```

(%o2) $\frac{d}{d\,x} f(x,y)$

(%o3) $\frac{d}{d\,x}\left(2\,\mathrm{atan}(4\,y+3\,x)+x^3\,y^2\right)$

(%o4) $\frac{6}{(4\,y+3\,x)^2+1}+3\,x^2\,y^2$

(%o5) $\frac{3}{5}$

```
(%i6) 'diff('f(x,y),y);
      'diff(f(x,y),y);
      diff(f(x,y),y);
      %,x=-1,y=0;
```

(%o6) $\frac{d}{d\,y} f(x,y)$

(%o7) $\frac{d}{d\,y}\left(2\,\mathrm{atan}(4\,y+3\,x)+x^3\,y^2\right)$

(%o8) $\frac{8}{(4\,y+3\,x)^2+1}+2\,x^3\,y$

(%o9) $\frac{4}{5}$

Można też użyć polecenia **jacobian** oraz **at**, pamiętając jednak, że wtedy wynik jest macierzą.

```
(%i10) 'at(jacobian(['f(x,y)],[x,y]),[x=-1,y=0]);
       %,nouns;
```

(%o10) $\left.\begin{bmatrix}\frac{d}{d\,x} f(x,y) & \frac{d}{d\,y} f(x,y)\end{bmatrix}\right|_{[x=-1,\,y=0]}$

(%o11) $\begin{bmatrix}\frac{3}{5} & \frac{4}{5}\end{bmatrix}$

Przykład 2

Obliczymy pochodne cząstkowe drugiego rzędu funkcji

$$f(x,y)=\ln(e^{2x}+3e^{-3y}).$$

Dziedziną funkcji f jest zbiór $D=\mathbb{R}^2$.

```
(%i1) f(x,y):= log(%e^(2*x)+%e^(-3*y) )$
```

Tym razem pojawi się inny symboliczny zapis pochodnych dzięki **derivabbrev**:

```
(%i2) derivabbrev:true$
```

```
(%i3) 'diff(f,x,2)=diff(f(x,y), x,2),factor;
```

(%o3) $f_{x\,x}=\frac{4\,\%e^{3\,y+2\,x}}{(\%e^{3\,y+2\,x}+1)^2}$

```
(%i5) 'diff(f,x,1,y,1)=diff(f(x,y),x,1,y,1);
```

(%o5) $f_{x\,y}=\frac{6\,\%e^{2\,x-3\,y}}{(\%e^{-3\,y}+\%e^{2\,x})^2}$

```
(%i4) 'diff(f,y,2)=diff(f(x,y), y,2),factor;
```

(%o4) $f_{y\,y}=\frac{9\,\%e^{3\,y+2\,x}}{(\%e^{3\,y+2\,x}+1)^2}$

```
(%i6) 'diff(f,y,1,x,1)=diff(f(x,y),y,1,x,1);
```

(%o6) $f_{x\,y}=\frac{6\,\%e^{2\,x-3\,y}}{(\%e^{-3\,y}+\%e^{2\,x})^2}$

Można też, do wyznaczenia wszystkich pochodnych cząstkowych drugiego rzędu, zastosować funkcję **hessian**.

```
(%i7) hessian('f(x,y),[x,y])=hessian(f(x,y),[x,y]),factor;
```

(%o7) $$\begin{bmatrix} f(x,y)_{x\,x} & f(x,y)_{x\,y} \\ f(x,y)_{x\,y} & f(x,y)_{y\,y} \end{bmatrix} = \begin{bmatrix} \dfrac{4\,\%e^{3\,y+2\,x}}{\left(\%e^{3\,y+2\,x}+1\right)^2} & \dfrac{6\,\%e^{3\,y+2\,x}}{\left(\%e^{3\,y+2\,x}+1\right)^2} \\ \dfrac{6\,\%e^{3\,y+2\,x}}{\left(\%e^{3\,y+2\,x}+1\right)^2} & \dfrac{9\,\%e^{3\,y+2\,x}}{\left(\%e^{3\,y+2\,x}+1\right)^2} \end{bmatrix}$$

Uwaga 1

Rozważane wyżej pochodne mieszane f''_{xy} oraz f''_{yx} są ciągłe na zbiorze $\mathbb{R}^2$, zatem na podstawie twierdzenia Schwarza wiemy, że są one równe.
W Maximie fakt ten oraz jego uogólnienie na pochodne mieszane wyższych rzędów jest wyraźnie zaznaczony, gdyż wszystkie pochodne mieszane mają jeden i ten sam symbol (patrz (%o5), (%o6), (%o7) oraz poniżej).

```
(%i1) derivabbrev:false$
      diff(f(x,y),x,1,y,1,x,1);
```

(%o2) $$\frac{d^3}{d\,x^2\,d\,y} f(x,y)$$

```
(%i3) derivabbrev:true$
      diff(f(x,y),x,1,y,1,x,1);
```

(%o4) $f(x,y)_{x\,x\,y}$

Przykład 3

Wyznaczymy ekstrema lokalne funkcji $f(x,y) = x^3 + 3xy^2 + 12xy$.

Dziedziną funkcji f jest zbiór $D = \mathbb{R}^2$.

```
(%i1) f(x,y):= x^3+3*x*y^2+12*x*y$
```

```
(%i2) fx:diff(f(x,y), x);
```

(%o2) $3\,y^2+12\,y+3\,x^2$

```
(%i3) fy: diff(f(x,y), y);
```

(%o3) $6\,x\,y+12\,x$

Wyznaczamy punkty stacjonarne oraz oznaczamy je: $P1$, $P2$, $P3$, $P4$.

```
(%i4) stacjonarne: solve([fx,fy], [x,y]);
```

(%o4) $[[x=0,y=-4],[x=0,y=0],[x=2,y=-2],[x=-2,y=-2]]$

```
(%i5) P1:stacjonarne[1]$P2:stacjonarne[2]$
      P3:stacjonarne[3]$P4:stacjonarne[4]$
```

```
(%i9) H: hessian(f(x,y), [x,y]);
```

Wyznaczamy hesjan funkcji f.

(%o9) $$\begin{bmatrix} 6\,x & 6\,y+12 \\ 6\,y+12 & 6\,x \end{bmatrix}$$

```
(%i10) W:determinant(H);
       W,P1;W,P2;W,P3;W,P4;
```
(%o10) $36\,x^2 - (6\,y + 12)^2$

(%o11) -144

(%o12) -144

(%o13) 144

(%o14) 144

Obliczamy wyznacznik W, hesjanu H, a następnie badamy znak tego wyznacznika w każdym punkcie stacjonarnym. Stąd wnioskujemy, że funkcja f posiada ekstrema lokalne jedynie w punktach $P3$ i $P4$.

```
(%i15) derivabbrev:true$
```

```
(%i16) fxx:H[1,1]$
```

```
(%i17) 'diff(f,x,2)= fxx;
```
(%o17) $f_{x\,x} = 6\,x$

Wybieramy z macierzy H pochodną f''_{xx} (element znajdujący się w pierwszym wierszu i pierwszej kolumnie) i przypisujemy do zmiennej fxx.

Następnie określamy znak tej pochodnej w punktach $P3$ oraz $P4$, aby ustalić, czy funkcja f osiąga w nich minimum czy maksimum lokalne.

```
(%i18) 'at(H[1,1],P3)=at(fxx,P3);
       'at(H[1,1],P4)=at(fxx,P4);
```
(%o18) $6\,x\big|_{[x=2,\,y=-2]} = 12$

(%o19) $6\,x\big|_{[x=-2,\,y=-2]} = -12$

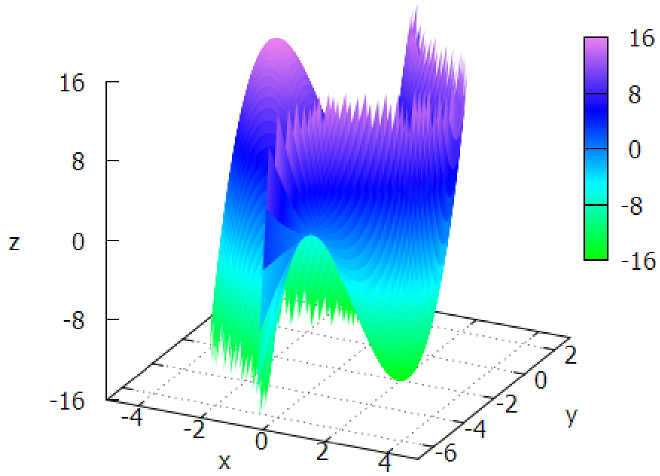
Otrzymujemy, że $f''_{xx}(2,-2) = 12 > 0$, więc funkcja osiąga minimum lokalne w tym punkcie. $f''_{xx}(-2,-2) = -12 < 0$, więc funkcja osiąga maksimum lokalne w tym punkcie.

Na końcu obliczamy wartości funkcji i szkicujemy jej wykres.

```
(%i20) f[min](x,y)=f(x,y),P3;
       f[max](x,y)=f(x,y),P4;
```
(%o20) $f_{min}(2,-2) = -16$

(%o21) $f_{max}(-2,-2) = 16$

```
(%i22) plot3d( f(x,y), [x, -5, 5], [y, -7, 3], [z,- 16,16],
                       [grid, 100, 100],
                       [legend, false],
                       [mesh_lines_color, false],
                       [xtics, 2], [ytics, 2], [ztics, 8],
                       same_xy,
                       [color_bar_tics, 8],
                       color_bar)$
```

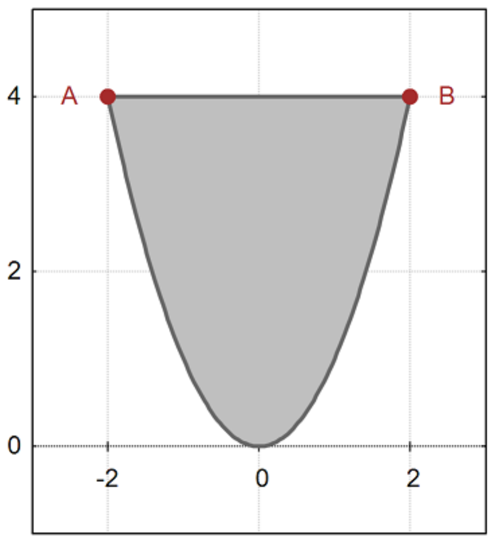


Przykład 4

Wyznaczymy wartość najmniejszą i największą funkcji
$f(x,y) = 2x^3 + 4x^2 + y^2 - 2xy$ na zbiorze $D = \{(x,y) \in \mathbb{R}^2 : x^2 \leq y \leq 4\}$.

```
(%i1) load(draw)$
```

```
(%i2) draw2d(grid=true,
             xtics_axis=true,
             xtics=2,ytics=2,
             xaxis=true,
             xaxis_type=solid,
             xrange=[-3,3],
             yrange=[-1,5],
             fill_color=grey,
             filled_func=4,
      explicit(x^2,x,-2,2),
             filled_func=false,
             color=gray40,
             line_width  = 3,
      explicit(4,x,-2,2),
      explicit(x^2,x,-2,2),
             color=brown,
             point_size=1.25,
             point_type=7,
             font= "Arial",
             font_size=14,
             label(["A",-2.5,4]),
             label(["B",2.5,4]),
      points([-2,2],[4,4]))$
```




```
(%i3) f(x,y):= 2*x^3+4*x^2+y^2-2*x*y$
```

Szukamy punktów stacjonarnych funkcji f we wnętrzu obszaru D.

```
(%i4) fx:diff(f(x,y), x);
```
(%o4) $-2\,y+6\,x^2+8\,x$

```
(%i5) fy: diff(f(x,y), y);
```
(%o5) $2\,y-2\,x$

```
(%i6) stacjonarne: solve([fx,fy], [x,y]);
```
(%o6) $[[x=-1,y=-1],[x=0,y=0]]$

Żaden z punktów stacjonarnych nie należy do wnętrza zbioru D.

Szukamy teraz punktów stacjonarnych funkcji f na brzegu opisanym równaniem $y = x^2$ dla $x \in (-2,2)$. Na rozważanym brzegu funkcja f przyjmuje postać:

```
(%i7) 'f(x,x^2)=f(x,x^2);
```
(%o7) $f(x,x^2)=x^4+4\,x^2$

Jest to funkcja jednej zmiennej x.

Nazwiemy ją g i wyznaczymy jej punkty stacjonarne.

```
(%i8) g(x):=''(f(x,x^2));
```
(%o8) $g(x):=x^4+4\,x^2$

```
(%i9) 'diff(g(x),x)=diff(g(x),x);
      realroots(diff(g(x),x));
```
(%o9) $\frac{d}{d\,x}(x^4+4\,x^2)=4\,x^3+8\,x$

(%o10) $[x=0]$

Ponieważ $x = 0$ należy do przedziału $(-2,2)$ oraz $y = x^2$, to otrzymujemy punkt stacjonarny $O = (0,0)$.

Badamy następnie istnienie punktów stacjonarnych funkcji f na brzegu opisanym równaniem $y = 4$ dla $x \in (-2,2)$. Funkcja f przyjmuje wtedy postać:

```
(%i11) 'f(x,4)=f(x,4);
```
(%o11) $f(x,4)=2\,x^3+4\,x^2-8\,x+16$

Jest to też funkcja jednej zmiennej x. Nazwiemy ją h.

```
(%i12) h(x):=''(f(x,4));
```
(%o12) $h(x):=2\,x^3+4\,x^2-8\,x+16$

Szukamy punktów stacjonarnych funkcji h, dla $x \in (-2,2)$.

```
(%i13) 'diff(h(x),x)=diff(h(x),x);
       solve(diff(h(x),x));
```
(%o13) $\frac{d}{d\,x}(2\,x^3+4\,x^2-8\,x+16)=6\,x^2+8\,x-8$

(%o14) $[x=-2,x=\frac{2}{3}]$

Tylko $x = \frac{2}{3}$ należy do przedziału (-2,2). Uwzględniając równanie brzegu $y = 4$, otrzymujemy więc kolejny punkt stacjonarny $P = \left(\frac{2}{3}, 4\right)$.
Aby wyznaczyć wartość najmniejszą i największą funkcji f na zbiorze D, porównujemy wartości funkcji w punktach O i P oraz w punktach brzegowych $A = (-2,4)$ i $B = (2,4)$.

```
(%i15) 'f(P)=f(2/3,4);'f(O)=f(0,0);'f(A)=f(-2,4);'f(B)=f(2,4);
```

(%o15) $f(P) = \frac{352}{27}$

(%o16) $f(O) = 0$

(%o17) $f(A) = 32$

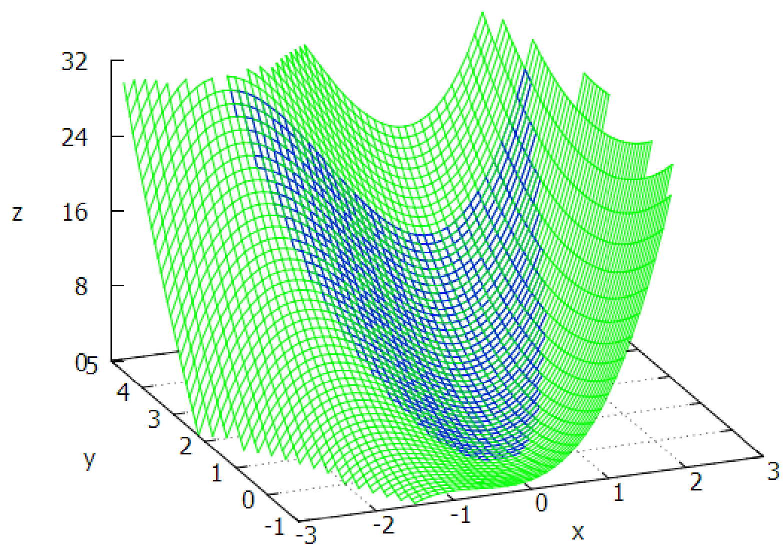
(%o18) $f(B) = 32$

Funkcja f przyjmuje zatem wartość najmniejszą, równą zero, w punkcie O, natomiast wartość największą, równą trzydzieści dwa, w punktach A oraz B.
Rysujemy teraz wykres funkcji f określonej na kwadracie $[-3,3] \times [-1,5]$ (kolorem zielonym) oraz zaznaczamy obcięcie tej funkcji do zbioru D (kolorem niebieskim). Aby uzyskać bardziej czytelny wykres, zbiór wartości funkcji f ograniczamy do przedziału $[0,32]$.

Definiujemy funkcję na zbiorze D.

```
(%i19) fD(x,y):=if x^2<=y and y<=4 then f(x,y)$
```

```
(%i20) plot3d([f(x,y),fD(x,y),[x,-3,3],[y,-1,5]],[z,0,32],[grid,80,80],[ztics,8],
                                   [palette, false], [color,green,blue],[legend,false])$
```



Przykład 5

Jak dobrać wymiary pudełka w kształcie prostopadłościanu, aby miało 1 m^3 objętości i jak najmniejszą powierzchnię?

Z warunków zadania wynika, że zachodzi równość $x\,y\,z = 1$, gdzie x, y, z są długościami boków prostopadłościanu. Z powyższej równości wyznaczamy zmienną z i wstawiamy ją do wzoru na pole powierzchni prostopadłościanu

$$p = 2(x\,y + y\,z + z\,x).$$

Dziedziną funkcji $p = p(x, y)$ jest zbiór $D = (0, \infty) \times (0, \infty)$.

```
(%i1) p(x,y):=''(ev(2*(x*y+y*z+z*x),z=1/(x*y)));
```
(%o1) $p(x, y) := 2\left(x\,y + \frac{1}{y} + \frac{1}{x}\right)$

W trudniejszych rachunkowo przykładach można też inaczej „wyeliminować” zmienną (w tym przypadku z).

```
(%i2) eliminate([x*y*z=1,  p=2*(x*y+y*z+z*x)],[z]);
```
(%o2) $[-2\,x^2\,y^2 - (2 - p\,x)\,y - 2\,x]$

```
(%i3) solve(%,p),expand;
```
(%o3) $[p = 2\,x\,y + \frac{2}{y} + \frac{2}{x}]$

W celu wyznaczenia wymiarów pudełka, które ma najmniejszą powierzchnię, obliczamy pochodne cząstkowe funkcji p i wyznaczamy jej punkty stacjonarne.

```
(%i4) px: diff(p(x,y), x),factor;
```
(%o4) $\frac{2\,(x^2\,y - 1)}{x^2}$

```
(%i5) py: diff(p(x,y), y),factor;
```
(%o5) $\frac{2\,(x\,y^2 - 1)}{y^2}$

Do rozwiązania układu równań stosujemy polecenie **algsys** z dodatkowym ustawieniem **realonly**:true, dzięki któremu w rozwiązaniu pojawią się jedynie punkty o współrzędnych rzeczywistych.

```
(%i6) stacjonarne: algsys([px=0,py=0], [x,y]),realonly:true;
```
(%o6) $[[x = 1, y = 1]]$

```
(%i7) P1:stacjonarne[1];
```
(%o7) $[x = 1, y = 1]$

Obliczamy Hesjan i jego wartość w punkcie stacjonarnym $P1$.

```
(%i8) H :hessian(p(x,y), [x,y]);
```
(%o8) $\begin{bmatrix} \frac{4}{x^3} & 2 \\ 2 & \frac{4}{y^3} \end{bmatrix}$

```
(%i9) W:determinant(H);W,P1;
```
(%o9) $\frac{16}{x^3\,y^3} - 4$

(%o10) 12

Ponieważ $W(P1) = 12 > 0$, więc funkcja f posiada ekstremum w tym punkcie. Badamy teraz znak $p''_{xx}(P1)$.

```
(%i11) pxx:H[1,1];pxx,P1;
(%o11) 4/x^3
(%o12) 4
```

$p''_{xx}(P1) = 4 > 0$, stąd funkcja f osiąga w tym punkcie minimum lokalne.

```
(%i13) 'p[min](x,y)=p(x,y),P1;
(%o13) p_min(1,1)=6
```

Pudełko o wymiarach $1\,m$, $1\,m$, $1\,m$ będzie miało najmniejszą powierzchnię, równą $6\,m^2$.

Przykład 6

Funkcja popytu na pewien produkt ma postać $f(p,d) = \frac{de^{2d-4}}{p^2}$, gdzie p oznacza cenę produktu, natomiast d oznacza dochód konsumenta. Obecna cena produktu wynosi $p_0 = 3$ jednostki pieniężne, natomiast dochód konsumenta jest równy $d_0 = 5$ jednostek pieniężnych. Korzystając z funkcji elastyczności ustalimy, jak zmieni się popyt na rozważany produkt, jeżeli
a) cenę podniesiemy o 1%, b) dochód konsumenta wzrośnie o 1%.

a) Korzystamy z interpretacji ekonomicznej elastyczności popytu na dane dobro względem ceny, która jest zdefiniowana wzorem

$$E_p(p,d) = f'_p(p,d)\frac{p}{f(p,d)}.$$

Elastyczność ta określa w przybliżeniu, o ile procent zmieni się popyt na dane dobro, gdy cena p wzrośnie o 1%.
Aby wyznaczyć elastyczność popytu względem ceny, definiujemy funkcję popytu f i obliczamy jej pochodną cząstkową względem p.

```
(%i1) f(p,d):=d*%e^(2*d-4)/p^2;
(%o1) f(p,d):= d %e^(2 d - 4) / p^2
```

```
(%i2) fp:diff(f(p,d),p);
(%o2) - 2 d %e^(2 d - 4) / p^3
```

Definiujemy funkcję elastyczności popytu względem ceny i liczymy jej wartość dla ceny $p_o = 3$ i dochodu $d_0 = 5$.

```
(%i3) Ep:(p/f(p,d))*fp;
(%o3) -2
```

```
(%i4) Ep,p=3,d=5;
(%o4) -2
```

Wynik oznacza, że wzrost ceny o 1% spowoduje spadek popytu o 2%.

b) Korzystamy z interpretacji ekonomicznej elastyczności popytu na dane dobro względem dochodu, która jest zdefiniowana wzorem

$$E_d(p,d) = f'_d(p,d)\frac{d}{f(p,d)}.$$

Elastyczność ta określa w przybliżeniu, o ile procent zmieni się popyt na dane dobro, gdy dochód d wzrośnie o 1%.
W celu wyznaczenia elastyczności popytu względem dochodu liczymy pochodną cząstkową zdefiniowanej wcześniej funkcji popytu względem zmiennej d.

```
(%i5) fd:diff(f(p,d),d),factor;
```

(%o5) $\dfrac{(2\,d+1)\,\%e^{2\,d-4}}{p^2}$

Definiujemy funkcję elastyczności popytu względem dochodu i liczymy jej wartość dla ceny $p_o = 3$ i dochodu $d_0 = 5$.

```
(%i6) Ed:(d/f(p,d))*fd;
```

(%o6) $2\,d+1$

```
(%i7) Ed,p=3,d=5;
```

(%o7) 11

Zatem wzrost dochodu o 1% spowoduje wzrost popytu o 11%.

Przykład 7

Dana jest funkcja produkcji $f(k,z) = (2k^{-0.2} + 3z^{-0.2})^{-\frac{1}{0.2}}$, gdzie k oznacza nakład kapitału, natomiast z - nakład pracy. Obecnie nakład kapitału wynosi $k_0 = 2$ jednostki pieniężne, natomiast nakład pracy jest równy $d_0 = 500$ roboczogodzin. Obliczyć:
a) o ile procent musi w przybliżeniu wzrosnąć nakład pracy, gdy nakład kapitału spadnie o 1%,
b) o ile procent musi w przybliżeniu wzrosnąć nakład kapitału, gdy nakład pracy spadnie o 1%,
aby produkcja się nie zmieniła.

a) W przypadku, gdy znana jest funkcja produkcji $y = f(k,z)$, gdzie y oznacza wielkość produkcji (w jednostkach fizycznych), k - nakład kapitału, natomiast z - nakład pracy (liczba zatrudnionych lub roboczogodzin), korzystamy z interpretacji ekonomicznej elastyczności substytucji kapitału przez pracę, która jest zdefiniowana wzorem

$$E_{kz}^{f}(k,z) = \frac{f_k'\,(k,z)}{f_z'\,(k,z)} \cdot \frac{k}{z}\ .$$

Elastyczność ta informuje, o ile procent musi w przybliżeniu wzrosnąć nakład pracy, gdy nakład kapitału spadł o 1%, aby produkcja się nie zmieniła.

W celu wyznaczenia elastyczności definiujemy funkcję produkcji i liczymy jej pochodne cząstkowe.

```
(%i1) f(k,z):=(2*k^(-0.2)+3*z^(-0.2))^(-1/0.2);
```

(%o1) $f(k,z):=(2\,k^{-0.2}+3\,z^{-0.2})^{\frac{-1}{0.2}}$

```
(%i2) fk:diff(f(k,z),k)$   fz:diff(f(k,z),z)$
```

Definiujemy funkcję elastyczności substytucji kapitału przez pracę i liczymy jej wartość dla nakładu kapitału $k_0 = 2$ i nakładu pracy $z_0 = 500$.
Wykonujemy też przekształcenie (patrz tabela 4.5) do postaci zawierającej ułamki zwykłe, aby otrzymać wynik dokładny dla postaci funkcji E^f_{kz}.
Następnie obliczamy wartości dokładną i przybliżoną, zgodnie z podanymi wyżej warunkami zadania.

```
(%i4) Ekz:(fk/fz)*(k/z),factor,rootscontract;
rat: replaced 0.19999999999999 by 1/5 = 0.2
rat: replaced - 0.19999999999999 by -1/5 = -            0.2
rat: replaced 0.66666666666666 by 2/3 = 0.66666666666666
```

(%o4) $\dfrac{2\left(\frac{z}{k}\right)^{1/5}}{3}$

```
(%i5) Ekz,k=2,z=500;%,numer;
```

(%o5) $\dfrac{2\,250^{1/5}}{3}$

(%o6) 2.01139211218172

Wynik oznacza, że wartość produkcji się nie zmieni, gdy w przypadku spadku nakładu kapitału o 1% nakład pracy w przybliżeniu wzrośnie o 2 %.

b) Korzystamy z interpretacji ekonomicznej elastyczności substytucji pracy przez kapitał, która jest zdefiniowana wzorem

$$E^f_{zk}(k,z) = \frac{f'_z\,(k,z)}{f'_k\,(k,z)} \cdot \frac{z}{k}\,.$$

Elastyczność ta informuje, o ile procent musi w przybliżeniu wzrosnąć nakład kapitału, gdy nakład pracy spadł o 1%, aby produkcja się nie zmieniła.

Definiując funkcję elastyczności substytucji pracy przez kapitał, zauważamy, że $E^f_{zk}(k,z) \cdot E^f_{kz}(k,z) = 1$. Następnie obliczamy jej wartość dla nakładu kapitału $k_0 = 2$ i nakładu pracy $z_0 = 500$.

```
(%i7) Ezk:1/Ekz,rootscontract;
```

(%o7) $\dfrac{3\left(\frac{k}{z}\right)^{1/5}}{2}$

```
(%i8) Ezk,k=2,z=500;%,numer;
```

(%o8) $\dfrac{3}{2\,250^{1/5}}$

(%o9) 0.49716810260099

Wynik interpretujemy następująco: wartość produkcji się nie zmieni, gdy w przypadku spadku nakładu pracy o 1% nakład kapitału w przybliżeniu wzrośnie o 0.5%.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Obliczyć wartości pierwszych pochodnych cząstkowych funkcji f w punkcie P_0, jeżeli:

a) $f(x,y) = 4x^2y^3\mathrm{arctg}(2x-y)$, $P_0 = (1,1)$,
b) $f(x,y) = e^{3x-2y}(2x+5y+1)$, $P_0 = (0,1)$.

Zadanie 2

Obliczyć pochodne cząstkowe drugiego rzędu funkcji:

a) $f(x,y) = \ln(e^{-x} + e^{3y})$, b) $f(x,y) = e^{x+4y}(3x+2y)$.

Zadanie 3

Wyznaczyć ekstrema lokalne funkcji:

a) $f(x,y) = -4x^2 + 2xy - y^2 - 6x + 2y$, b) $f(x,y) = e^{y-x}(2x^2 - y^2)$,

c) $f(x,y) = 6x^2y - 4x^3 + y^2 - 4y$, d) $f(x,y) = \ln(y+2x) - 2x - y^2$.

Zadanie 4

Wyznaczyć wartość najmniejszą i największą funkcji f na zbiorze D, jeżeli:

a) $f(x,y) = x^2 - 2xy + 2y^3 + 4y^2$, $D = \{(x,y) \in \mathbb{R}^2 : x^2 \le y \le 1\}$,
b) $f(x,y) = xe^{x-3y}$, $D = \{(x,y) \in \mathbb{R}^2 : 1 \le x \le e \wedge 0 \le y \le \ln x\}$,
c) $f(x,y) = y^2 - x^2$, $D = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 \le 9 \wedge x \ge 0\}$,
d) $f(x,y) = 2x^2 - y^2 - 18x$, $D = \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 \le 16\}$.

Zadanie 5

Znaleźć takie trzy liczby, których suma jest równa 1, natomiast iloczyn jest największy. Jaka jest wartość tego iloczynu?

Zadanie 6

Przedstawić liczbę 64 w postaci iloczynu trzech czynników tak, by ich suma była najmniejsza. Jakie to liczby?

Zadanie 7

Jakie powinny być wymiary otwartego zbiornika w kształcie prostopadłościanu o objętości 4 m^3, żeby jego pole powierzchni było najmniejsze?

Zadanie 8

Jakie powinny być wymiary prostopadłościanu o powierzchni 27 m^2, żeby jego objętość była największa?

Zadanie 9

Funkcja popytu na pewne dobro ma postać $f(p,d) = \frac{d}{p(d+9p)} + \frac{9p}{d(d+9p)}$, gdzie p oznacza cenę dobra, natomiast d oznacza dochód konsumenta. Obecna cena dobra wynosi $p_0 = 0.1$ jednostki pieniężnej, natomiast dochód konsumenta jest równy $d_0 = 1$ jednostkę pieniężną. Korzystając z funkcji elastyczności, ustalić, jak zmieni się popyt na rozważane dobro, jeżeli:

a) cenę podniesiemy o 1%,

b) dochód konsumenta wzrośnie o 1%.

Zadanie 10

Dana jest funkcja produkcji $f(k,z) = 2k^{-0.2}z^{-0.5}$, gdzie k oznacza nakład kapitału, natomiast z – nakład pracy. Obecnie nakład kapitału wynosi $k_0 = 2$ jednostki pieniężne, natomiast nakład pracy jest równy $d_0 = 500$ roboczogodzin. Wyznaczyć:

a) o ile procent musi wzrosnąć nakład pracy, gdy nakład kapitału spadnie o 1%,

b) o ile procent musi wzrosnąć nakład kapitału, gdy nakład pracy spadnie o 1%,

aby produkcja się nie zmieniła.

Odpowiedzi

1. a) $f'_x(1,1) = 2\pi + 4$, $f'_y(1,1) = 3\pi - 2$, b) $f'_x(0,1) = 20e^{-2}$, $f'_y(0,1) = -7e^{-2}$.
2. a) $f''_{xx}(x,y) = \frac{e^{3y+x}}{(e^{3y+x}+1)^2}$, $f''_{yy}(x,y) = \frac{9e^{3y+x}}{(e^{3y+x}+1)^2}$, $f''_{xy}(x,y) = f''_{yx}(x,y) = \frac{3e^{3y-x}}{(e^{3y}+e^{-x})^2}$,
   b) $f''_{xx}(x,y) = (2y+3x+6)e^{4y+x}$, $f''_{yy}(x,y) = 16(2y+3x+1)e^{4y+x}$,
   $f''_{xy}(x,y) = f''_{yx}(x,y) = 2(4y+6x+7)e^{4y+x}$.
3. a) $f_{max}\left(-\frac{2}{3},\frac{1}{3}\right) = \frac{7}{3}$, b) $f_{min}(-2,-4) = -8e^{-2}$, c) $f_{min}(0,2) = -4$,
   d) $f_{max}\left(\frac{1}{4},\frac{1}{2}\right) = -\frac{3}{4}$,
4. a) $f_{min}(0,0) = 0$, $f_{max}(-1,1) = 9$, b) $f_{min}(2,\ln 2) = \frac{e^2}{4}$, $f_{max}(e,0) = e^{e+1}$,
   c) $f_{min}(-3,0) = f_{min}(3,0) = -9$, $f_{max}(0,3) = 9$,
   d) $f_{min}(3,-\sqrt{7}) = f_{min}(3,\sqrt{7}) = -43$, $f_{max}(-4,0) = 104$.
5. $\frac{1}{3}$, $\frac{1}{3}$, $\frac{1}{3}$.
6. 4, 4, 4.
7. 2 m, 2 m, 1 m.
8. 3 m, 3 m, 3 m.
9. a) Wzrost ceny o 1% spowoduje spadek popytu w przybliżeniu o 1.3 %.
   b) Wzrost dochodu o 1% spowoduje wzrost popytu w przybliżeniu o 0.3 %.
10. a) 0.4 %, b) 2.5 %.

16. Funkcja uwikłana jednej zmiennej

Niech F będzie funkcją dwóch zmiennych. Każdą funkcję $y = y(x)$ ciągłą na pewnym przedziale I oraz spełniającą równość $F(x, y(x)) = 0$ dla wszystkich $x \in I$ nazywamy funkcją uwikłaną określoną równaniem $F(x, y) = 0$.
Równanie $F(x, y) = 0$ nazywamy wtedy postacią uwikłaną funkcji $y = y(x)$.
Aby narysować wykres funkcji jednej zmiennej danej w postaci uwikłanej oraz obliczać jej pochodne, będziemy korzystać z poleceń przedstawionych w tabeli 16.1 oraz w tabeli 16.2.

Tabela 16.1

PODSTAWOWE POLECENIA GRAFICZNE [1)]

implicit_plot(F(x,y)=0, [x, x_{min}, x_{max}], [y, y_{min}, y_{max}]**)**
- rysuje krzywą opisaną równaniem $F(x, y) = 0$, ograniczając ją do prostokąta $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ [2)]

implicit_plot([r_1, r_2, …], [x, x_{min}, x_{max}], [y, y_{min}, y_{max}]**)**
- rysuje krzywe opisane równaniami: r_1, r_2, ... ograniczając je do prostokąta $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ [2)]

contour_plot(f(x,y), [x, x_{min}, x_{max}], [y, y_{min}, y_{max}], *opcje***)**
- rysuje poziomice funkcji f (patrz tabela 6.1)
opcje: [**gnuplot_preamble**, *"set cntrparam levels ….;…* [3)] *"*]

k	- k poziomów
incremental w_{min},*k*,w_{max}	- poziomy od w_{min} do w_{max} co *k*
discrete w_1, w_2, …, w_n	- lista wybranych poziomów
set size ratio 1	- ustawienie proporcji jednostek na osiach

draw2d(*opcje*, **implicit(**F(x,y)=0,x,x_{min},x_{max},y,y_{min},y_{max}**))**
- rysuje krzywą opisaną równaniem $F(x, y) = 0$, ograniczając ją do prostokąta $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$
opcje: ip_grid oraz ip_grid_in (patrz tabela 6.3)

draw3d(*opcje*, **explicit(**F(x,y),x,x_{min},x_{max},y,y_{min},y_{max}**))**
- rysuje powierzchnię $z = F(x, y)$ z zaznaczonymi poziomicami,
opcje: contour_levels oraz contour (patrz tabela 6.4 oraz przykład 7 z rozdziału 6)

1) Wszystkie polecenia graficzne wymienione w tabeli 16.1 występują również w wersji z dodanym przedrostkiem **wx**, pozwalającym na bezpośrednie wklejenie grafiki do skryptu Maximy.
2) Funkcja dostępna po wczytaniu pakietu **implicit_plot**. Podobnie działa polecenie **draw2d(implicit(**F(x,y)=0,x,x_{min},x_{max},y,y_{min},y_{max}**))** (patrz tabela 6.5 i tabela 6.6), które występuje również w wersji dla funkcji uwikłanej dwóch zmiennych (patrz przykład 6 w rozdziale 6).
3) Inne opcje: set cntrparam bspline|linear - metody przybliżania krzywych, set cntrparam order n - rząd aproksymacji dla bspline (od 2 do 10), set key … - ustawienia legendy.

Tabela 16.2

POCHODNE FUNKCJI UWIKŁANEJ
implicit_derivative(r, [x], [n], y**)** - oblicza pochodne (od pierwszego do n-tego rzędu) funkcji uwikłanej $y = y(x)$ określonej równaniem $r[0]: F(x, y) = 0$ i przechowuje je w formie tablicy, ale ich nie wyświetla [1)]
listarray(r**)** - ogólnie: listuje wektor r, w tym przypadku: podaje równanie $r[0]$ oraz kolejne pochodne i-tego rzędu $r[i]$ funkcji uwikłanej $y = y(x)$, dla $i \le n$, gdzie n deklarowane jest w poleceniu **implicit_derivative( )**

[1)] Funkcja dostępna po wczytaniu pakietu **impdiff**.

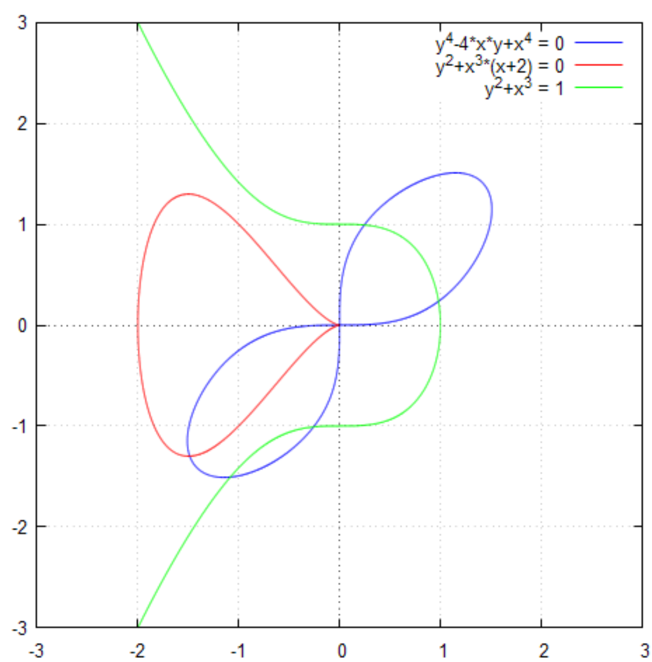
ZADANIA PRZYKŁADOWE

Przykład 1

Korzystając z pakietu **implicit_plot** naszkicujemy wykresy krzywych danych równaniami:

a) $x^4 + y^4 - 4xy = 0$, b) $y^2 + x^3(2 + x) = 0$, c) $x^3 + y^2 = 1$.

```
(%i2)   load(implicit_plot)$ wxplot_size:[500,500]$

(%i3)   wximplicit_plot([x^4+y^4-4*x*y=0, y^2+x^3*(2+x)=0, x^3+y^2=1],
                 [x,-3,3], [y,-3,3], [same_xy,true], grid2d)$
```



Przykład 2

Podamy wzory i naszkicujemy wykresy wszystkich funkcji uwikłanych postaci $y = y(x)$, ciągłych i określonych na możliwie maksymalnym przedziale, spełniających równanie $x^2 + y^2 = 2y$.

Przypiszemy rozważane równanie do zmiennej r i wyznaczymy z niego, za pomocą polecenia **solve,** zmienną y.

```
(%i1)  r: x^2+y^2=2*y;
(r)    y^2+x^2=2 y
```

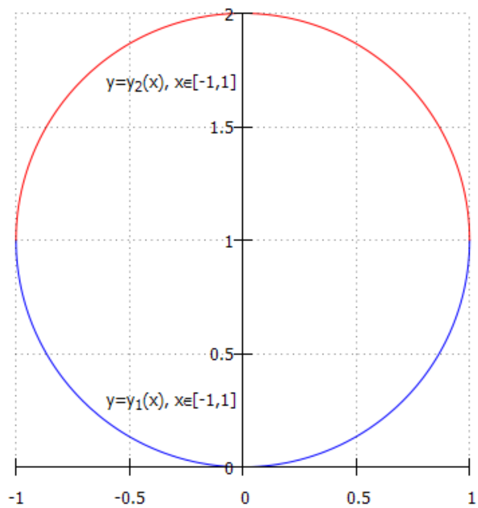
```
(%i2)  k:solve(r,y);
(k)    [y=1-sqrt(1-x^2), y=sqrt(1-x^2)+1]
```

Otrzymaliśmy dwie funkcje, których naturalną dziedziną jest przedział [-1,1]:

$$y_1(x) = 1 - \sqrt{1-x^2} \text{ oraz } y_2(x) = 1 + \sqrt{1-x^2}.$$

Narysujemy je korzystając z polecenia **plot2d (**z odpowiednimi ustawieniami) oraz z polecenia **rhs (**aby przywołać prawe strony równości z listy k).

```
(%i3)  plot2d([rhs(k[1]), rhs(k[2])], [x,-1,1], [y,0,2], [box,false],
          grid2d, [legend,false], [axes,solid], [same_xy,true],
          [label,["y=y_1(x), x∈[-1,1]",-0.6,0.3],
             ["y=y_2(x), x∈[-1,1]",-0.6,1.7]] )$
```



Uwaga 1.

Rozważane w powyższym przykładzie równanie jest równaniem okręgu o postaci kanonicznej $x^2 + (y-1)^2 = 1$.

Uwaga 2

Jeśli jednak nie jest możliwe bezpośrednie wyznaczenie z równania $F(x,y)=0$ funkcji uwikłanych postaci $y=y(x)$ (tak jak w przykładzie 2), to do badania, czy równanie określa jednoznacznie ciągłą funkcję uwikłaną na pewnym otoczeniu wskazanego punktu, stosujemy twierdzenie o istnieniu i różniczkowalności funkcji uwikłanej. Korzystając z niego możemy wyznaczyć pochodną funkcji y, a co za tym idzie, badać jej ekstrema i przedziały monotoniczności.

Przykład 3

Wyznaczymy pochodną funkcji uwikłanej $y=y(x)$ określonej równaniem $(x^2+y^2)^2=2(x^2-y^2)$ (lemniskata Bernoulliego).

Najpierw wprowadzimy zależność funkcyjną zmiennej y od x, a następnie zróżniczkujemy stronami równanie r (pochodne lewej i prawej strony równania r, to pochodne cząstkowe względem zmiennej x funkcji złożonych).

```
(%i3)  r: (x^2+y^2)^2=2*(x^2-y^2);
       depends(y,x);
       diff(r,x);
```

(r) $(y^2+x^2)^2=2\,(x^2-y^2)$

(%o2) $[y(x)]$

(%o3) $2\,(y^2+x^2)\left(2\,y\left(\frac{d}{d\,x}\,y\right)+2\,x\right)=2\left(2\,x-2\,y\left(\frac{d}{d\,x}\,y\right)\right)$

Otrzymane równanie rozwiążemy względem pochodnej funkcji uwikłanej, czyli względem $\frac{dy}{dx}$. Ponieważ polecnie **solve** podaje wynik w postaci wektora rozwiązań, dlatego użyjemy **%[1]**. Zadbamy też o postać iloczynową otrzymanego wyrażenia.

```
(%i5)  solve(%,'diff(y,x))$
       %[1],factor;
```

(%o5) $\frac{d}{d\,x}\,y=-\frac{x\,(y^2+x^2-1)}{y\,(y^2+x^2+1)}$

Powyższe zadanie możemy też rozwiązać inną metodą - poprzez wprowadzenie funkcji F tak, jak w twierdzeniu o istnieniu funkcji uwikłanej.
Najpierw musimy odwołać zadeklarowaną wcześniej zależność y od x.

```
(%i9)  remove([y,x],dependency)$
       F(x,y):=''(lhs(r)-rhs(r));
```

(%o9) $F(x,y):=(y^2+x^2)^2-2\,(x^2-y^2)$

```
(%i11) Fx:diff(F(x,y),x),factor;
       Fy:diff(F(x,y),y),factor;
```

(Fx) $4\,x\,(y^2+x^2-1)$

(Fy) $4\,y\,(y^2+x^2+1)$

(%i12) "y'(x)"=-Fx/Fy;

(%o12) $y'(x) = -\frac{x\,(y^2+x^2-1)}{y\,(y^2+x^2+1)}$

Otrzymaliśmy ten sam wynik, co w (%o5).

Wyrażenie $-\frac{x\,(x^2+y^2-1)}{y\,(x^2+y^2+1)}$ jest określone dla $y \neq 0$. Zauważmy, że jeśli do równania r podstawimy teraz w miejsce zmiennej y wartość 0, to otrzymamy punkty, na otoczeniu których nie istnieje jednoznacznie określona funkcja uwikłana.

```
(%i7) r,y=0;
      solve(%,x);
```

(%o6) $x^4 = 2\,x^2$

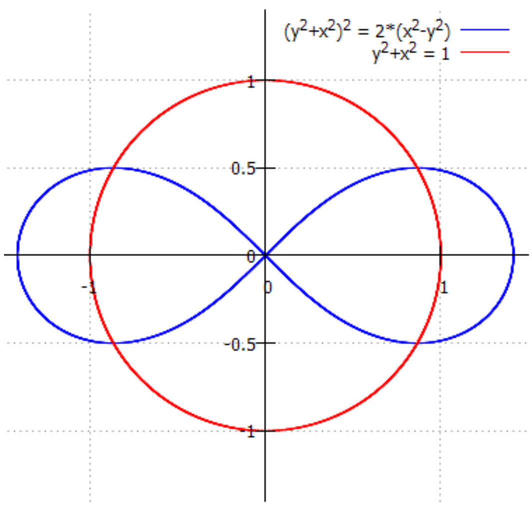
(%o7) $[x = -\sqrt{2}, x = \sqrt{2}, x = 0]$

Natomiast (0,0), $(\sqrt{2}, 0)$, $(-\sqrt{2}, 0)$ to punkty krzywej, w których $F_y' = 0$.

Zauważmy jeszcze, że pochodna przyjmuje wartość 0, gdy $F_x' = 0$, a więc dla $x = 0$ lub $x^2 + y^2 = 1$. Oznacza, to że ekstrema lokalne funkcji uwikłanej mogą się pojawić w punktach o odciętej 0 lub w punktach przecięcia krzywej z okręgiem.

```
(%i13) load(implicit_plot)$
```

```
(%i14) implicit_plot([r, y^2+x^2=1], [x,-1.5,1.5], [y,-1.4,1.4], [xtics,1], [ytics,0.5],
                 grid2d, [box,false], [axes,solid], [same_xy,true])$
```



Przykład 4

Obliczymy, przy pomocy pakietu **impdiff**, pochodną pierwszego oraz drugiego rzędu funkcji uwikłanej $y = h(x)$ określonej równaniem $x^3 + y^2 = 3xy$. Następnie sprawdzimy, że punkt (2,2) leży na krzywej opisanej tym równaniem i wyznaczymy równanie stycznej do krzywej w tym punkcie.

Zaczniemy od wczytania pakietu **impdiff** oraz danego równania jako elementu tablicy r.

```
(%i1)	load(impdiff)$

(%i2)	r[0]:x^3+y^2=3*x*y$
```

Następnie zastosujemy polecenia **implicit_derivative** oraz **listarray**. Pierwsze oblicza i przechowuje wyniki w tablicy r, natomiast drugie pozwala wyświetlić wyniki (zgodnie z zadeklarowanym n). Możemy też do pochodnych odpowiednio pierwszego i drugiego rzędu odwoływać się przez $r[1]$ i $r[2]$.

```
(%i4)	implicit_derivative(r,[x],[2],y)$ listarray(r);
```

(%o4) $$[y^2+x^3=3\,x\,y, \frac{3\,y-3\,x^2}{2\,y-3\,x}, -\frac{(24\,x-18)\,y^2+\left(54\,x-72\,x^2\right)\,y+18\,x^4}{8\,y^3-36\,x\,y^2+54\,x^2\,y-27\,x^3}]$$

```
(%i7)	r[0];r[1];r[2];
```

(%o5) $$y^2+x^3=3\,x\,y$$

(%o6) $$\frac{3\,y-3\,x^2}{2\,y-3\,x}$$

(%o7) $$-\frac{(24\,x-18)\,y^2+\left(54\,x-72\,x^2\right)\,y+18\,x^4}{8\,y^3-36\,x\,y^2+54\,x^2\,y-27\,x^3}$$

Zbadamy teraz, czy współrzędne punktu (2,2) spełniają równanie $r[0]$.

```
(%i8)	[x0,y0] : [2,2]$

(%i9)	r[0], x=x0, y=y0, pred;
(%o9)	true
```

Równanie stycznej do funkcji uwikłanej $y = h(x)$ w punkcie x_0 ma postać:

$$y = h_1'(x_0)(x - x_0) + h(x_0).$$

W naszym zadaniu $h(x_0)$ to y_0, natomiast $h'(x_0)$ to $r[1]$, jeżeli odpowiednio w miejsce x oraz y wstawimy x_0 oraz y_0.

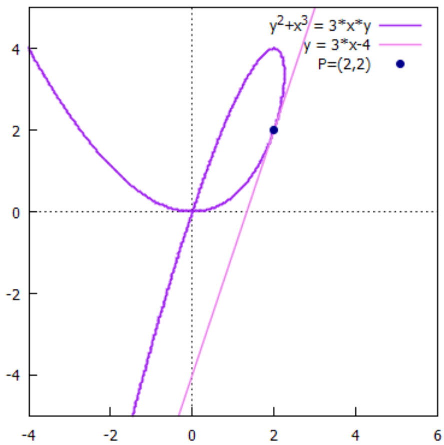
```
(%i10)	m: r[1], x=x0, y=y0;
(m)	3
```

Równanie szukanej stycznej ma więc postać:

```
(%i11)	styczna: y = m*(x-x0) + y0,expand;
(styczna)	y=3 x -4
```

Na koniec narysujemy krzywą określoną równaniem $x^3 + y^2 = 3xy$ i styczną do tej krzywej w punkcie (2,2), ograniczając się jedynie do pewnego zakresu zmiennych x oraz y. Oczywiście, styczna do wykresu funkcji h jest równocześnie styczną do krzywej.

```
load(draw)$
draw2d( terminal=wxt, dimensions=[400,400],
        xaxis=true, yaxis=true,
        yrange=[-5,5], line_width=1.5,
        proportional_axes=xy,
        key=string(r[0]),
        color=purple,
    implicit(r[0],x,-4,6,y,-5,5),
        key=string(styczna),
        color=violet,
    explicit(rhs(styczna),x,-4,6),
        key=concat("P=(",x0,",",y0,")"),
        color=dark_blue,
        point_type=filled_circle,
        points([[x0,y0]]) )$
```



Przykład 5

Wyznaczymy ekstrema lokalne funkcji uwikłanej postaci $y = y(x)$ określonej równaniem $(x^2 + y^2 - x)^2 = x^2 + y^2$ (kardioida).

Równanie to można zapisać w postaci $F(x, y) = 0$, gdzie

$$F(x, y) = (x^2 + y^2 - x)^2 - x^2 - y^2.$$

Dla uproszczenia obliczeń w Maximie, zamiast definiować funkcje przez $nazwa(x, y) :=$ będziemy stosować przypisania do zmiennej $nazwa$: .

```
(%i1)   F:(x^2+y^2-x)^2-(x^2+y^2);
(F)     (y^2+x^2-x)^2-y^2-x^2
```

Aby wyznaczyć ekstrema lokalne funkcji uwikłanej zmiennej x, znajdziemy te punkty, które równocześnie spełniają warunki:

$$F(x, y) = 0, \quad F'_x(x, y) = 0 \ \text{ oraz } \ F'_y(x, y) \neq 0.$$

Obliczymy najpierw pochodne F'_x i F'_y

```
(%i3)   Fx:diff(F,x),factor;Fy:diff(F,y),factor;
(Fx)    2 (2 x y^2 - y^2 + 2 x^3 - 3 x^2)
(Fy)    2 y (2 y^2 + 2 x^2 - 2 x - 1)
```

Następnie, po zadeklarowaniu, że szukamy rozwiązań rzeczywistych zapisywanych według reguł **algebraic**, rozwiążemy układ równań za pomocą polecenia **algsys** (można też użyć **solve**).

```
(%i5)   realonly:true$ algebraic:true$
```

```
(%i6)   P:algsys([F=0,Fx=0],[x,y]);
```

(P) $[[x=\frac{3}{4},y=-\frac{3^{3/2}}{4}],[x=\frac{3}{4},y=\frac{3^{3/2}}{4}],[x=0,y=0]]$

Liczbę rozwiązań przypiszemy do zmiennej n.

```
(%i7)   n:length(P)$
```

Wyznaczymy F''_{xx} oraz iloraz $I(x,y) = -\frac{F''_{xx}(x,y)}{F'_y(x,y)}$ i sformułujemy wnioski w oparciu o intrukcję pętli **for** z instrukcjami warunkowymi **if** i poleceniem **print**.

```
(%i8)   Fxx:diff(F,x,2),factor;
```

(Fxx) $4\,(y^2+3\,x^2-3\,x)$

```
(%i9)   I:-Fxx/Fy;
```

(I) $-\frac{2\,(y^2+3\,x^2-3\,x)}{y\,(2\,y^2+2\,x^2-2\,x-1)}$

```
(%i10)  for i:1 thru n do if subst(P[i],Fy)#0  then
           ( print("Dla punktu ",'P[i],"=(",rhs(P[i][1]),", ",rat(rhs(P[i][2])),"):") ,
             print(" y''(",rhs(P[i][1]),")=",subst(P[i],I),compare(subst(P[i],I),0),"0,")
          if subst(P[i],I)<0 then
             print("maximum lokalne  ",y[max](rhs(P[i][1]))=rat(rhs(P[i][2])),".")
        else (if subst(P[i],I)>0 then
             print("minimum lokalne  ",y[min](rhs(P[i][1]))=rat(rhs(P[i][2])),"."))
        else print("W punkcie ",'P[i],"=(",rhs(P[i][1]),",",
             rat(rhs(P[i][2])),") nie ma ekstremum.")$
```

Dla punktu P_1 $=(\ \frac{3}{4}\ ,\ \ -\frac{3\sqrt{3}}{4}\ )$:

$y''(\ \frac{3}{4}\ )=\ \frac{\sqrt{3}}{2}\ >\ 0,$

minimum lokalne $y_{min}\left(\frac{3}{4}\right)=-\frac{3\sqrt{3}}{4}$.

Dla punktu P_2 $=(\ \frac{3}{4}\ ,\ \ \frac{3\sqrt{3}}{4}\ )$:

$y''(\ \frac{3}{4}\ )=\ -\frac{\sqrt{3}}{2}\ <\ 0,$

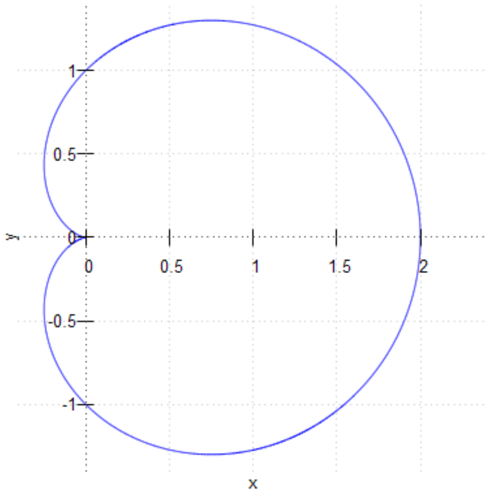
maximum lokalne $y_{max}\left(\frac{3}{4}\right)=\frac{3\sqrt{3}}{4}$.

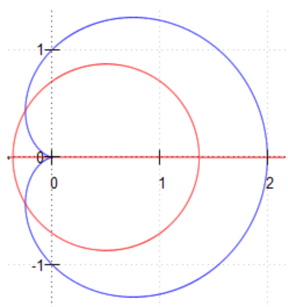
W punkcie P_3 $=(\ 0\ ,\ 0\ )$ *nie ma ekstremum.*

Dla kolejnych punktów z listy P (rozwiązań układu $F(x,y)=0$, $F'_x(x,y)=0$) sprawdziliśmy warunek $F'_y(x_i,y_i)\neq 0$, gdzie $i=1,\dots,n$. Następnie sprawdziliśmy, czy punkty spełniające warunek $F'_y(x_i,y_i)\neq 0$, spełniają też warunek wystarczający istnienia ekstremum tzn. $-\frac{F''_{xx}(x_i,y_i)}{F'_y(x_i,y_i)}\neq 0$. Iloraz ten (oznaczyliśmy go przez I) jest oczywiście równy drugiej pochodnej funkcji uwikłanej w punkcie x_i, a jego znak rozstrzyga, czy mamy minimum, czy maksimum lokalne.

Narysujemy teraz wykres rozważanej kardioidy.

```
(%i11) load(implicit_plot)$ wxplot_size:[400,400]$

(%i13) ip_grid:[150,150]$ ip_grid_in:[30,30]$

(%i14) wximplicit_plot([F=0],[x,-0.4,2.4], [y,-1.4,1.4], [same_xy,true],
                [xtics,0,0.5,2], [ytics,-1,0.5,1], grid2d, [box,false] )$
```





Zauważmy jeszcze, że punkty przecięcia rozważanej krzywej ze zbiorem rozwiązań równania $F'_y=0$, czyli $y(2y^2+2x^2-2x-1)=0 \Leftrightarrow y=0 \vee \left(x-\frac{1}{2}\right)^2+y^2=\frac{1}{4}$ (zbiór zaznaczony na rysunku kolorem czerwonym) to punkty, o których nie możemy powiedzieć, że na ich otoczeniu podane równanie jednoznacznie określa pewną ciągłą funkcję uwikłaną postaci $y=y(x)$. Widać, że mamy 4 takie punkty. Jeden z nich, punkt (0,0), pojawił się wśród punktów stacjonarnych. W pozostałych - styczna do krzywej jest prostopadła do osi Ox.
Punkty, w których zerują się obie pochodne cząstkowe to punkty osobliwe (punkty zwrotu, izolowane albo węzłowe). Punkt (0,0) jest punktem zwrotu.

Dla funkcji $F\colon \mathbb{R}^2 \to \mathbb{R}$ poziomice (warstwice) to zbiory postaci

$$\{(x,y) \in \mathbb{R}^2\colon\ F(x,y) = c\},\ \text{gdzie}\ \ c \in \mathbb{R}.$$

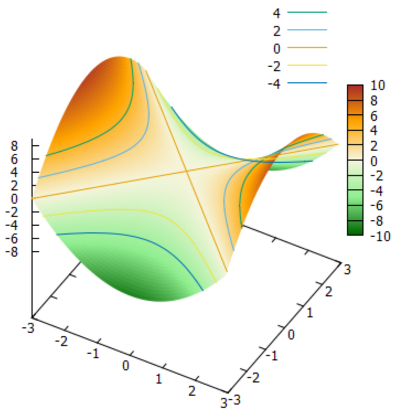
W przykładzie 6 oraz w przykładzie 7 przedstawimy wykresy poziomic oraz powierzchni $z = F(x,y)$, z których te poziomice powstały. Poziomice są podzbiorami płaszczyzny Oxy. Jednak dla lepszej wizualizacji przedstawimy też zbiory

$$\{(x,y,z) \in \mathbb{R}^3\colon\ F(x,y) = c\ \wedge\ z = c\,\},\ \text{gdzie}\ \ c \in \mathbb{R}.$$

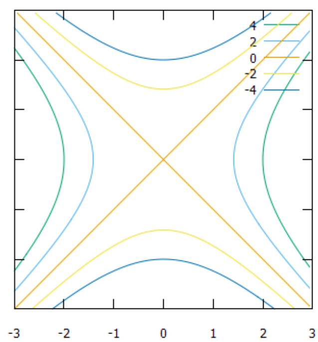
Przykład 6

Naszkicujemy powierzchnię $f(x,y) = x^2 - y^2$, $(x,y) \in [-3,3] \times [-3,3]$ oraz kilka jej poziomic.

```
(%i1)   f(x,y):=x^2-y^2$

(%i3)   load(draw)$
        set_draw_defaults(contour_levels=[-4,2,4],
                          terminal=wxt,
                          xu_grid=100,
                          yv_grid=100)$
```




```
(%i6)   g1:gr3d(enhanced3d=true,
                view=[50,30],
                palette=[dark-green,light-green,
                         beige,orange,brown],
                contour=surface,
                dimensions=[400,900],
                explicit(f(x,y),x,-3,3,y,-3,3))$
        g2:gr3d(contour=map,
                proportional_axes=xy,
                explicit(f(x,y),x,-3,3,y,-3,3))$
        draw(columns=1,g1,g2)$
```



Przykład 7

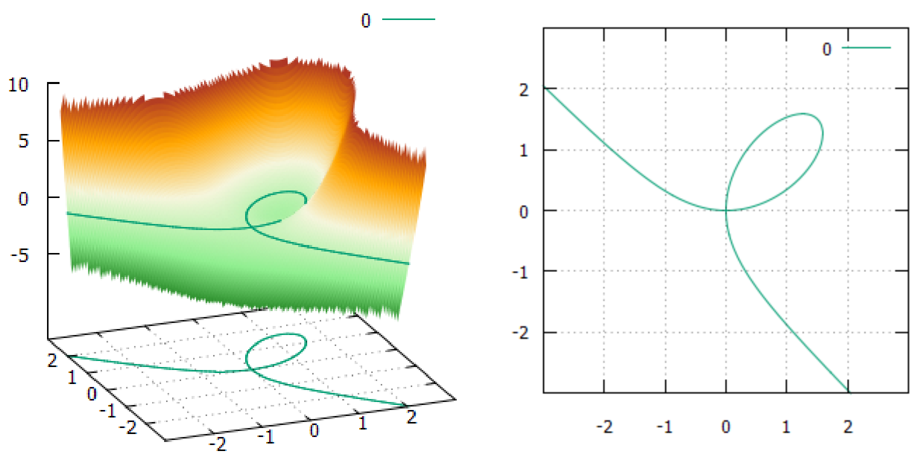
Naszkicujemy krzywą daną równaniem $F(x,y) = c$ jako poziomicę funkcji F.

a) $x^3 + y^3 - 3xy = 0$ (liść Kartezjusza), b) $x^4 + y^4 - x^2 - y^2 = -\frac{1}{4}$.

Podobnie jak w przykładzie 2, zastosujemy polecenia pakietu **draw**. Zaczniemy od wczytania pewnych ustawień domyślnych dla każdego z podpunktów.

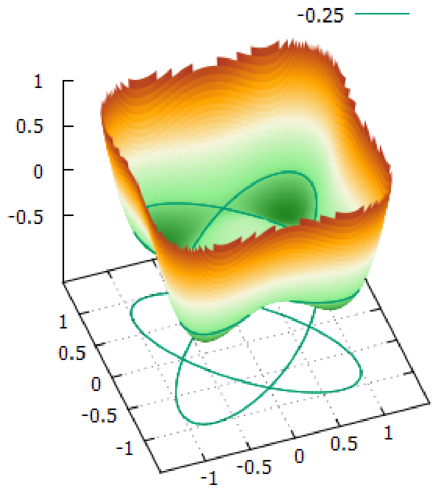
a)

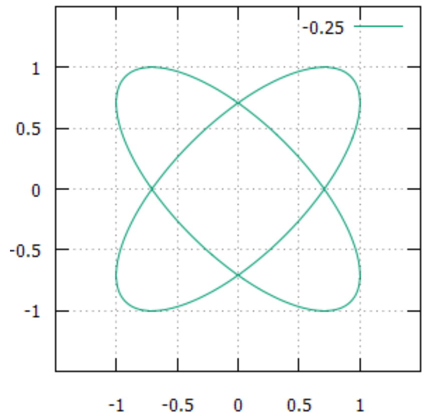
```
(%i1)  load(draw)$

(%i2)  set_draw_defaults(terminal=wxt, dimensions=[400,400], view=[70,340],
                xtics=[-2,1,2], ytics=[-2,1,2], ztics=5, proportional_axes=xy,
                enhanced3d=true, colorbox=false, surface_hide=true,
                palette=[dark-green, light-green, beige, orange, brown],
                contour_levels={0}, xu_grid=150, yv_grid=150)$

(%i3)  F(x,y):=x^3+y^3-3*x*y$

(%i4)  draw3d(contour=both, zrange=[-5,10], explicit(F(x,y),x,-3,3,y,-3,3))$

(%i5)  draw3d(contour=map, grid=true, explicit(F(x,y),x,-3,3,y,-3,3))$
```



b)

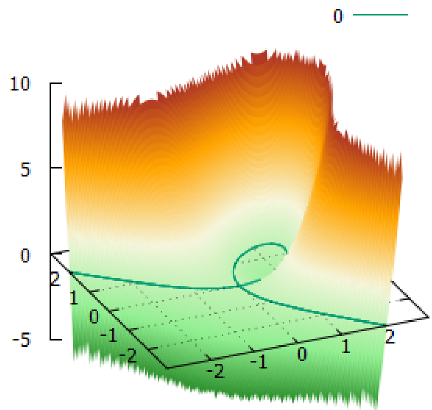
```
(%i1)  load(draw)$

(%i2)  set_draw_defaults(terminal=wxt, dimensions=[400,400], proportional_axes=xy,
                xtics=[-1,0.5,1], ytics=[-1,0.5,1], ztics=0.5, view=[45,340],
                enhanced3d=true, colorbox=false, surface_hide=true,
                palette=[dark-green, light-green, beige, orange, brown],
                contour_levels={-0.25}, xu_grid=150, yv_grid=150)$

(%i3)  F(x,y):=x^4+y^4-x^2-y^2$

(%i4)  draw3d(contour=both, zrange=[-0.5,1], explicit(F(x,y),x,-1.5,1.5,y,-1.5,1.5))$

(%i5)  draw3d(contour=map, grid=true, explicit(F(x,y),x,-1.5,1.5,y,-1.5,1.5))$
```





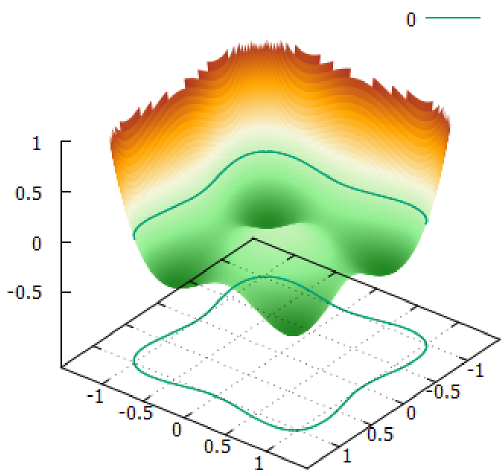
Uwaga 3

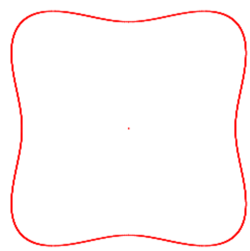
Jeśli w podpunkcie a) ustawimy dodatkowo opcję **xyplane**=0, to otrzymamy szukaną krzywą jako przecięcie powierzchni $z = x^3 + y^3 - 3xy$ z płaszczyzną Oxy (patrz rys. obok).



Zmieniając w podpunkcie b) wartość opcji **contour_levels** na {0} oraz **view** na [120,40] otrzymamy widok powierzchni „od spodu", a poziomica będzie mieć inny kształt. Zauważmy, że w tym przypadku do poziomicy należy punkt (0,0), który nie jest widoczny na rysunku.

Zbiorem punktów spełniających równanie $x^4 + y^4 - x^2 - y^2 = 0$ jest więc widoczna poniżej krzywa oraz punkt (0,0), który jest punktem izolowanym.



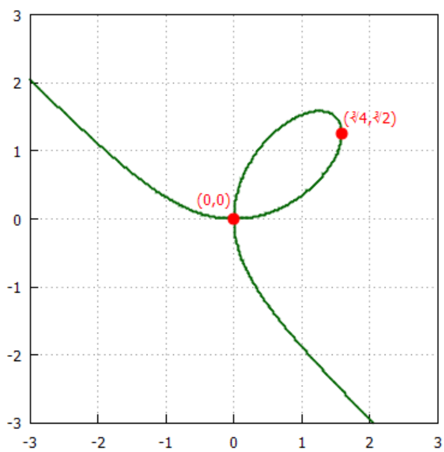


Uwaga 4

Pakiet **draw** pozwala rysować krzywe uwikłane zarówno przy pomocy funkcji **draw3d** i użytych w niej obiektów typu **explicit** (poziomice funkcji dwóch zmiennych) jak też przy pomocy funkcji **draw2d** i użytych w niej obiektów typu **implicit**.

W przykładzie 6 i w przykładzie 7 stosowaliśmy funkcję **draw3d**. Teraz do krzywej z przykładu 7 a) (podobnie jak w przykładzie 4) zastosujemy funkcję **draw2d,** aby zaznaczyć punkty i ich współrzędne.

```
(%i1)   load(draw)$
(%i2)   set_draw_defaults(terminal=wxt, grid=true,
                          dimensions=[400,400],
                          proportional_axes=xy)$
(%i3)   F(x,y):=x^3+y^3-3*x*y$
(%i4)   draw2d(line_width=1.5, color=dark-green,
               implicit(x^3+y^3-3*x*y=0,x,-3,3,y,-3,3),
               color=red, point_type=7, point_size=1.5,
               label(["(0,0)",-0.3,0.3],["(∛4,∛2)",2,1.5]),
               points([0,4^(1/3)],[0,2^(1/3)]))$
```



Zauważmy, że zarówno w punkcie (0,0) jak i w punkcie $\left(\sqrt[3]{4}, \sqrt[3]{2}\right)$ nie istnieją jednoznacznie określone funkcje uwikłane postaci $y = y(x)$ opisane równaniem $x^3 + y^3 - 3xy = 0$. Punkt (0,0) to punkt węzłowy krzywej.

Przykład 8

Dane są funkcje

$$F(x,y) = x^2 + y^2, \; G(x,y) = \sqrt{x^2 + y^2} \text{ oraz } H(x,y) = e^{-x^2-y^2}.$$

Rozważać je będziemy na zbiorze $[-2,2] \times [-2,2]$.

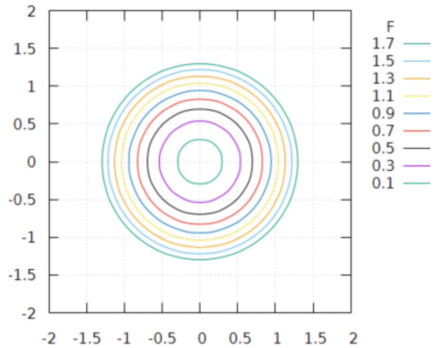
Wyznaczymy poziomice każdej z funkcji dla tych samych wartości stałych c_i zmieniających się od 0.1 do 1.9 z krokiem równym 0.2, a następnie porównamy ich układ.

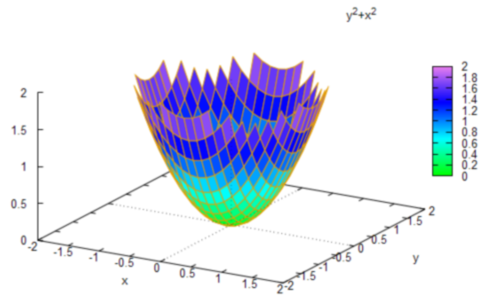
Tym razem skorzystamy z polecenia **contour_plot** i powiązanych z tym poleceniem opcji pakietu Gnuplot. Dla funkcji F, G oraz H ustalimy jednakowe zakresy zmiennych x, y oraz z.

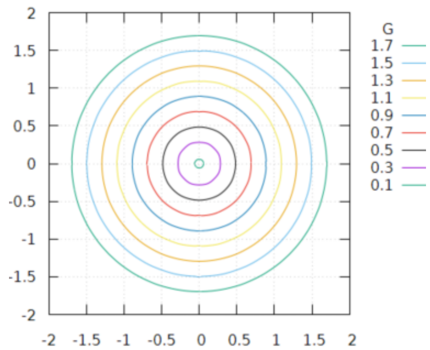
```
(%i11)  ustawienia:"set cntrparam levels incremental  0.1,0.2,1.9; set grid;
                    set cntrparam bspline; set cntrparam order 10;
                    set key rmargin top horizontal; set size ratio 1; "$

(%i3)   F(x,y):=x^2+y^2$ G(x,y):=sqrt(x^2+y^2)$ H(x,y):=2*exp(-x^2-y^2)$

(%i7)   wxcontour_plot(F, [x,-2,2], [y,-2,2], [gnuplot_preamble, ustawienia])$
        wxcontour_plot(G, [x,-2,2], [y,-2,2], [gnuplot_preamble, ustawienia])$
        wxcontour_plot(H, [x,-2,2], [y,-2,2], [gnuplot_preamble, ustawienia])$

(%i10)  wxplot3d(F, [x,-2,2], [y,-2,2], [z,0,2], [color_bar, true])$
        wxplot3d(G, [x,-2,2], [y,-2,2], [z,0,2], [color_bar, true])$
        wxplot3d(H, [x,-2,2], [y,-2,2], [z,0,2], [color_bar, true])$
```

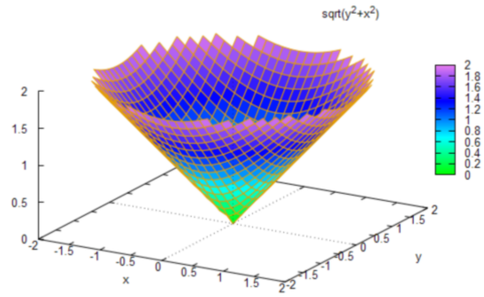
Wszystkie widoczne poziomice są okręgami scentrowanymi w punkcie (0,0), a ich zagęszczenie mówi, jak stroma jest powierzchnia na danej wysokości.

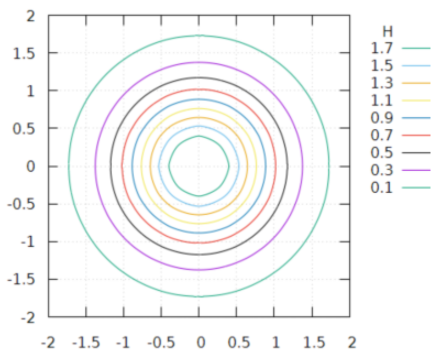
Z zamieszczonych poniżej rysunków można odczytać, że poziomice stożka układają się równomiernie, poziomice paraboloidy - im dalej od środka tym gęściej, a poziomice na ostatniej powierzchni zagęszczają się w pobliżu środka.

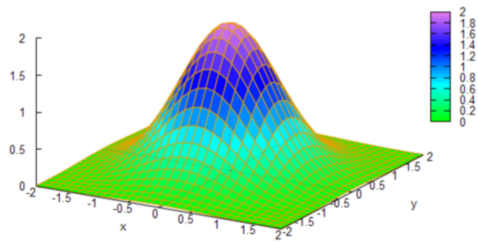












Uwaga 5

Kolejny przykład może być ilustracją tego, że Maxima (a właściwie Gnuplot) nie zawsze potrafi narysować dokładnie poziomice. Pokaże też, że czasem więcej informacji o wartościach badanej funkcji możemy uzyskać biorąc po uwagę kształt i gęstość poziomic, niż analizując wykres samej funkcji przy automatycznych ustawieniach.

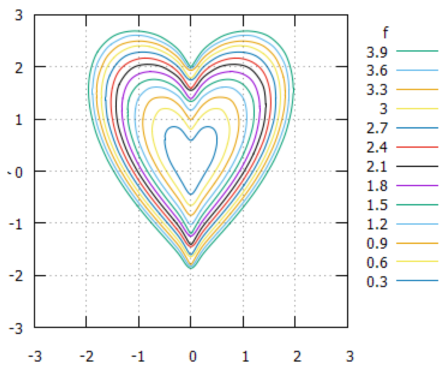
Przykład 9

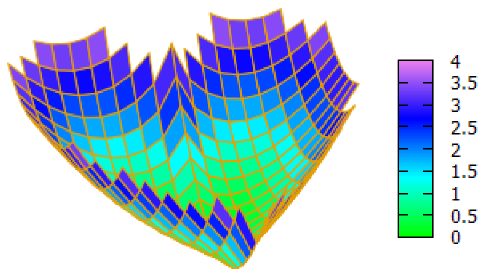
Naszkicujemy poziomice na wybranych poziomach oraz wykresy funkcji, z których powstały:

a) $f(x,y) = x^2 + \left(y - \sqrt[3]{x^2}\right)^2$, b) $g(x,y) = (x^2 + y^2)^3 - 4x^2y^2$.

a)

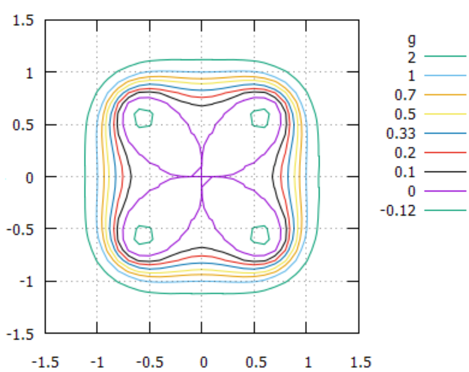
```
(%i3)  f(x,y):= x^2 + (y-x^(2/3))^2$
       contour_plot(f, [x,-3,3], [y, -3,3],[gnuplot_preamble,
                 " set size ratio 1; set grid; set cntrparam order 10;
                   set cntrparam bspline; set key rmargin top horizontal;
                   set cntrparam levels incremental  0,0.3,4; " ])$
       plot3d( f, [x,-3,3], [y,-3,3], [z,0,4], [color_bar, true], [box,false])$
```

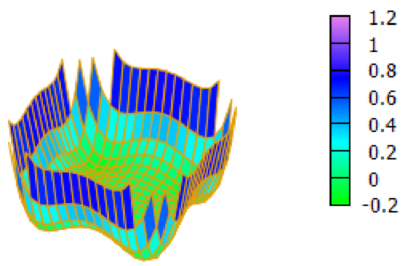




b)

```
(%i6)  g(x,y):=(x^2+y^2)^3-4*x^2*y^2$
       contour_plot(g, [x,-1.5,1.5], [y, -1.5,1.5],
                 [gnuplot_preamble, " set size ratio 1; set grid; set cntrparam line;
                 set cntrparam order 10; set key rmargin top horizontal;
                 set cntrparam levels discrete -0.12,0,0.1,0.2,0.33,0.5,0.7,1,2;"])$
       plot3d( g, [x,-1.5,1.5], [y,-1.5,1.5], [z,-0.2,1.2],
                 [same_xyz,true],[color_bar, true], [box,false])$
```





ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1.

Naszkicować krzywe określone poniższymi równaniami, a następnie wyznaczyć pochodne funkcji uwikłanych postaci $y = y(x)$ określonych tymi równaniami:
a) $x^2 + y^2 + 8x + 4y + 19 = 0$ b) $x^4 + 2y^2 - 4xy = 0$,
c) $1 - xy - e^{xy} = 0$, d) $\ln(xy) = 0$.

Zadanie 2.

Wyznaczyć ekstrema lokalne funkcji uwikłanych $y = y(x)$ określonych równaniami:
a) $x^2 + y^2 + 8x + 4y + 19 = 0$, b) $x^4 + 2y^2 - 4xy = 0$.

Odpowiedzi

1. a) $y'(x) = -\frac{x+4}{y+2}$, b) $y'(x) = \frac{y-x^3}{y-x}$, c) $y'(x) = -\frac{y}{x}$, d) $y'(x) = -\frac{y}{x}$.
2. a) $y_{max}(-4) = -1$, $y_{min}(-4) = -3$;
   b) $y_{min}\left(-\frac{\sqrt{3}}{\sqrt{2}}\right) = -\frac{3\sqrt{3}}{2\sqrt{2}}$, $y_{max}\left(\frac{\sqrt{3}}{\sqrt{2}}\right) = \frac{3\sqrt{3}}{2\sqrt{2}}$.

17. Rachunek całkowy funkcji wielu zmiennych

Tabela 17.1

POLECENIA	OPIS
integrate(f(x), x, a, b**)**	całka oznaczona funkcji f na przedziale (a, b)
depends([x,y],[u,v]**)**	wprowadza zależność funkcyjną zmiennych x, y od zmiennych u, v
jacobian([x,y],[u,v]**)**	jakobian przekształcenia $x = x(u, v)$, $y = y(u, v)$
dblint('f, 'r, 's , a, b**)**	całka podwójna funkcji f zmiennych x i y po obszarze ograniczonym funkcjami r i s zmiennej x (odpowiednio z dołu i z góry), gdzie $x \in (a, b)$ [1)]

[1)] Funkcja znajduje się w pakiecie **dblint**.

ZADANIA PRZYKŁADOWE

Przykład 1

Obliczymy całki iterowane $\int_0^1 \left(\int_{1-x}^{e^x} f(x,y)\, dy \right) dx$, gdy $f(x,y) = x + 2y$ oraz naszkicujemy obszar całkowania.

Najpierw zdefiniujemy odpowiednie funkcje:

```
(%i1) f(x,y):=x+2*y$  r(x):=1-x$  s(x):=exp(x)$  a:0$  b:1$
```

Przedstawimy rozważane całki w zapisie symbolicznym (z przypisaniem do zmiennej I):

```
(%i6) I:'integrate('integrate('f(x,y),y,r(x),s(x)),x,a,b);
```

(%o6) $\int_0^1 \int_{1-x}^{\%e^x} f(x,y)\, dy\, dx$

By otrzymać wynik, wystarczy zastosować komendę **nouns** do zmiennej I:

```
(%i7) I,nouns;
```

(%o7) $\frac{\%e^2}{2}$

A teraz wykonamy etapy pośrednie obliczeń:

```
(%i8) 'integrate(f(x,y),y,r(x),s(x));
```

(%o8) $\int_{1-x}^{\%e^x} 2y + x\, dy$

Najpierw całka wewnętrzna

```
(%i9) integrate(f(x,y),y);
```

(%o9) $y^2 + x\, y$

i jej funkcja pierwotna.

Podstawiamy teraz do funkcji pierwotnej (tu przywoływanej przez **%**) najpierw górną, a później dolną granicę całkowania i obliczamy odpowiednią różnicę.

```
(%i10) ev(%,y=s(x))-ev(%,y=r(x)),expand;
```

(%o10) $\%e^{2x} + x\,\%e^{x} + x - 1$

Zatem otrzymaliśmy, że:

```
(%i11) display(integrate(f(x,y),y,r(x),s(x)))$
```

$$\int_{1-x}^{\%e^{x}} 2y + x\,dy = \%e^{2x} + x\,\%e^{x} + x - 1$$

Podstawiając wynik całki wewnętrznej (tu przywoływany przez **%th(**2**)** tzn. drugi wynik liczony od końca), dostajemy całkę pojedynczą.

```
(%i12) 'integrate(%th(2),x,a,b);
```

(%o12) $\int_{0}^{1} \%e^{2x} + x\,\%e^{x} + x - 1\,dx$

Wyznaczamy funkcję pierwotną funkcji o etykiecie **(%o10)**, którą tym razem przywołamy przez **%th(**3**)**.

```
(%i13) integrate(%th(3),x);
```

(%o13) $\frac{\%e^{2x}}{2} + (x-1)\,\%e^{x} + \frac{x^2}{2} - x$

Podstawiamy granice całkowania i, jeśli zachodzi taka potrzeba, dodajemy polecenia upraszczające w zależności od typu funkcji występujących w całce (podobnie jak w **(%i10)**).

```
(%i14) ev(%,x=b)-ev(%,x=a);
```

(%o14) $\frac{\%e^{2}}{2}$

Zatem mamy

```
(%i15) display(integrate(integrate(f(x,y),y,r(x),s(x)),x,a,b))$
```

$$\int_{0}^{1} \%e^{2x} + x\,\%e^{x} + x - 1\,dx = \frac{\%e^{2}}{2}$$

Mogliśmy też skorzystać z polecenia **dblint**, aby uzyskać wynik numeryczny.

```
(%i16) load (dblint)$
```

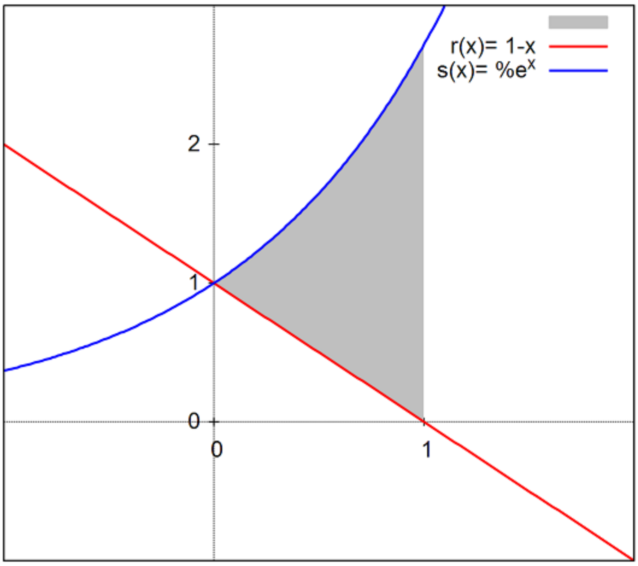
```
(%i17) dblint ( 'f, 'r, 's , a, b);
(%o17) 3.694530095348207
```

Uwaga 1

Należy pamiętać, że składnia polecenia **dblint** wymaga wcześniejszego zdefiniowania funkcji f, r oraz s, tak jak to zrobiliśmy na początku tego przykładu. Polecenie **dblint** stosuje algorytm numeryczny – metodę Simpsona.

Na koniec ilustracja graficzna obszaru całkowania:

```
(%i18) load(draw)$

(%i19) draw2d(dimensions = [600,600], font = "Arial", font_size  = 14,
       xrange = [a-1,b+1], yrange = [-1,3], grid = true,
       xaxis = true, xtics_axis = true, xtics = {0,1},
       yaxis = true, ytics_axis = true, ytics= {0,1,2},
       line_width = 2, fill_color = grey, key = " ", filled_func = s,
       explicit(r,x,a,b),  filled_func = false,
       key = concat("r(x)= ", string(r(x)) ), color = red, explicit(r,x,a-1,b+1),
       key = concat("s(x)= ", string(s(x)) ), color = blue, explicit(s,x,a-1,b+1)
       )$
```



Przykład 2

Obliczymy całkę podwójną $\iint_D 2xy\,dx\,dy$, gdzie D jest obszarem ograniczonym krzywymi: $y = x^2 - 2x$, $y = 1 - x$.

Podobnie jak w przykładzie 5 c) z rozdziału 14, najpierw wyznaczamy odcięte punktów przecięcia podanych krzywych, a następnie rysujemy krzywe i zaznaczamy obszar całkowania.

```
(%i1) [f1,f2]: [x^2-2*x,1-x]$
      [x1,x2]: sort(map('rhs,solve(f1=f2)),"<");
```

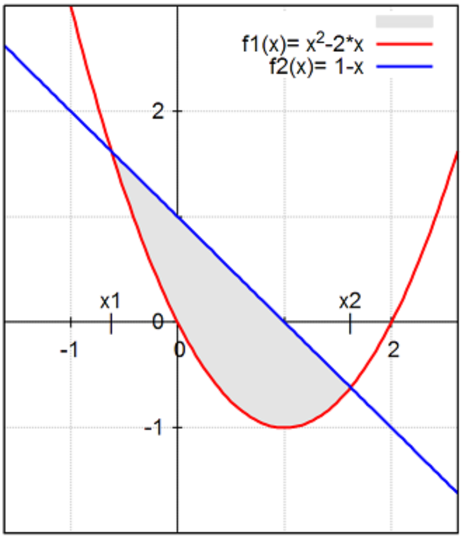
(%o2) $[-\frac{\sqrt{5}-1}{2}, \frac{\sqrt{5}+1}{2}]$

Na podstawie powyższych obliczeń i zamieszczonego poniżej wykresu zamieniamy całkę podwójną na iterowane.

$$\iint_D 2xy\,dx\,dy = \int_{x1}^{x2} \left(\int_{f1(x)}^{f2(x)} 2xy\,dy \right) dx$$

```
(%i3) load(draw)$
```

```
(%i4) draw2d(proportional_axes = xy, font = "Arial", font_size = 12,
      xrange = [x1-1,x2+1], yrange = [-2,3], grid = true,
      xaxis = true, xaxis_type = solid, xtics_axis = true, xtics = [-1,1,2],
      yaxis = true, yaxis_type = solid, ytics_axis = true, ytics = [-1,1,2],
      line_width = 2, fill_color = gray90, key = "  ", filled_func = f2,
      explicit(f1,x,x1,x2), filled_func = false,
      key = concat("f1(x)= ", string(f1)), color = red, explicit(f1,x,x1-1,x2+1),
      key = concat("f2(x)= ", string(f2)), color = blue, explicit(f2,x,x1-1,x2+1),
      color = black, label(["x1",x1,0.2],["|",x1,0],["x2",x2,0.2],["|",x2,0])
      )$
```



Obliczamy całki:

```
(%i5) 'integrate('integrate(2*x*y,y,f1,f2),x,x1,x2);
```

(%o5) $2\int_{-\frac{\sqrt{5}-1}{2}}^{\frac{\sqrt{5}+1}{2}} x \int_{x^2-2x}^{1-x} y\, dy\, dx$

```
(%i6) 'integrate(integrate(2*x*y,y,f1,f2),x,x1,x2),ratsimp;
```

(%o6) $-\int_{-\frac{\sqrt{5}-1}{2}}^{\frac{\sqrt{5}+1}{2}} x^5 - 4x^4 + 3x^3 + 2x^2 - x\, dx$

```
(%i7) integrate(integrate(2*x*y,y,f1,f2),x,x1,x2),expand;
```

(%o7) $-\frac{5^{3/2}}{12}$

Przykład 3

Obliczymy całkę $\int_0^1 \left(\int_{\sqrt{y}}^1 xy^2\,dx\right)dy$, a następnie narysujemy obszar całkowania oraz dokonamy zmiany kolejności całkowania.

```
(%i1) 'integrate('integrate(x*y^2,x,sqrt(y),1),y,0,1);
      'integrate(integrate(x*y^2,x,sqrt(y),1),y,0,1);
      %,expand;%,nouns;
```

(%o1) $\int_0^1 \int_{\sqrt{y}}^1 x\,dx\; y^2\,dy$

(%o2) $\int_0^1 \left(\frac{1}{2}-\frac{y}{2}\right) y^2\,dy$

(%o3) $\int_0^1 \frac{y^2}{2}-\frac{y^3}{2}\,dy$

(%o4) $\frac{1}{24}$

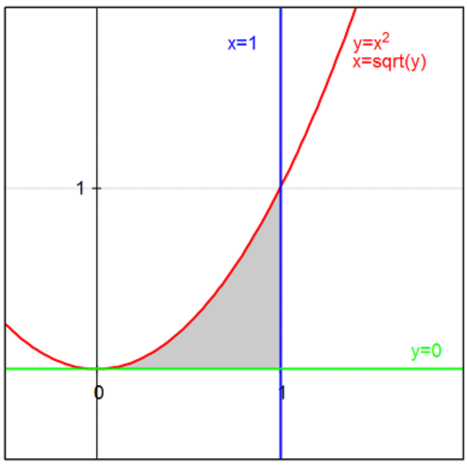
Poniżej zamieszczamy rysunek obszaru całkowania uzyskany w Maximie za pomocą pakietu **draw**. Obszar (zaznaczony szarym kolorem) jest obszarem normalnym zarówno względem osi Oy

$$D = \{(x,y) \in \mathbb{R}^2 : 0 \le y \le 1 \ \land\ \sqrt{y} \le x \le 1\}$$

jak i osi Ox

$$D = \{(x,y) \in \mathbb{R}^2 : 0 \le x \le 1 \ \land\ 0 \le y \le x^2\}.$$

```
draw2d(proportional_axes = xy,
font = "Arial", font_size = 12,
xtics_axis = true, xtics = {0,1},
ytics_axis = true, ytics = {1},
yaxis = true, yaxis_type = solid,
xrange = [-1/2,2], yrange = [-1/2,2],
grid = true, line_width = 2,
fill_color = gray80, filled_func = 0,
explicit(x^2,x,0,1), filled_func = false,
color = red, explicit(x^2,x,-1/2,2),
label(["y=x^2",1.5,1.8],["x=sqrt(y)",1.6,1.7]),
color = blue, implicit(x=1,x,-1/2,2,y,-1,2),
label(["x=1",.8,1.8]),
color = green, implicit(y=0,x,-1/2,2,y,-1,2),
label(["y=0",1.8,0.1])
)$
```



Sprawdzimy teraz poniższą równość

$$\int_0^1 \left(\int_{\sqrt{y}}^1 xy^2\,dx\right)dy = \int_0^1 \left(\int_0^{x^2} xy^2\,dy\right)dx.$$

W tym celu obliczymy całki po prawej stronie równości i porównamy z wynikiem (%o4).

```
(%i5) 'integrate('integrate(x*y^2,y,0,x^2),x,0,1);
      'integrate(integrate(x*y^2,y,0,x^2),x,0,1);
      %,nouns;
```

(%o5) $\int_0^1 x \int_0^{x^2} y^2 \, dy \, dx$

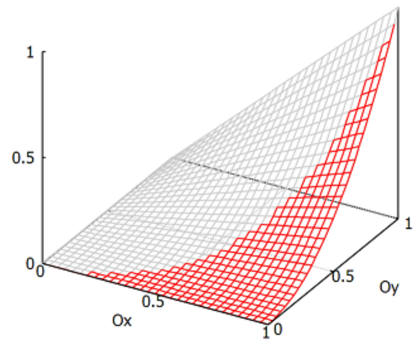
(%o6) $\dfrac{\int_0^1 x^7 \, dx}{3}$

(%o7) $\dfrac{1}{24}$

Ponieważ funkcja podcałkowa jest nieujemna na obszarze D, to możemy rozważaną całkę i jej wynik interpretować jako objętość obszaru przestrzennego pomiędzy powierzchnią zaznaczoną kolorem czerwonym (wykres zdefiniowanej poniżej funkcji f), a jej prostokątnym rzutem na płaszczyznę Oxy.

```
(%i1) f(x,y):=if (x>sqrt(y) and x<1 and x>0) then x*y^2$
```

```
(%i2) load(draw)$
      draw3d(zrange = [0,1],
      grid = true, xplane = 0,
      proportional_axes = xyz,
      xtics = 0.5, ytics = 0.5,
      ztics = 0.5, color = grey80,
      explicit(x*y^2,x,0,1,y,0,1),
      xlabel = "Ox", ylabel = "Oy",
      color = red,
      explicit(f(x,y),x,0,1,y,0,1)
       )$
```



Przykład 4

Obliczymy objętość bryły ograniczonej wykresami funkcji:

$$f_1(x,y) = x^2 + y^2, \quad f_2(x,y) = 6 - \sqrt{x^2 + y^2}.$$

```
(%i1) f1(x,y):=x^2+y^2$
      f2(x,y):=6-sqrt(x^2+y^2)$
```

Najpierw, podstawiając $s = \sqrt{x^2 + y^2}$, wyznaczymy równanie krzywej przecięcia wykresów funkcji f_1 oraz f_2.

```
(%i3) f1(x,y)=f2(x,y);
      s^2=6-s;solve(%);
```

(%o3) $y^2 + x^2 = 6 - \sqrt{y^2 + x^2}$

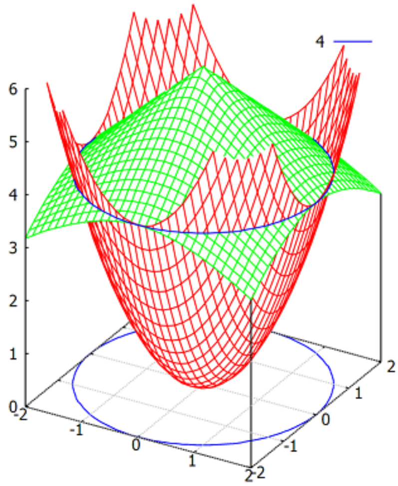
(%o4) $s^2 = 6 - s$

(%o5) $[s = -3, s = 2]$

Otrzymujemy równanie kwadratowe. Interesować nas będzie tylko dodatni pierwiastek.

Zauważmy, że dla $s = 2$ obie funkcje f_1 i f_2 przyjmują wartość 4. Oznacza to, że ich wykresy przecinają się na wysokości 4 nad płaszczyzną Oxy oraz że krzywa przecięcia jest okręgiem o równaniu $x^2 + y^2 = 4$ (gdyż $s^2 = 4$).

```
(%i6) load(draw)$
      draw3d(
      grid = true,
      xtics = 1, ytics = 1,
      zrange = [0,6],
      surface_hide = true,
      xplane = 0,
      contour_levels = 1,
      contour = both,
      proportional_axes = xyz,
      color = red,
      explicit(f1,x,-2,2,y,-2,2),
      color = green,
      explicit(f2,x,-2,2,y,-2,2)
      )$
```



$$|V| = \iint_D (f_2(x,y) - f_1(x,y))\, dx\, dy, \qquad D = \{(x,y) \in \mathbb{R}^2 \colon x^2 + y^2 \le 4\}.$$

Powyższą całkę obliczymy dokonując zamiany zmiennych na współrzędne biegunowe. Obszar całkowania we współrzędnych biegunowych jest opisany nierównościami:

$$0 \le \varphi \le 2\pi, \;\; 0 \le r \le 2\,.$$

Najpierw określimy przekształcenie zmiennych i wyznaczymy jakobian tego przekształcenia (patrz też rozdział 8.4):

```
(%i8) [x,y]:[r*cos(phi),r*sin(phi)]$
```

```
(%i9) jacobian([x,y],[r,phi])$ determinant(%)$ J:trigsimp(%);
(%o11) r
```

Następnie obliczymy całki:

```
(%i12) assume(r>=0);
(%o12) [r>=0]
```

Deklarujemy zakres zmiennej r, by poprawnie wyznaczyć całki.

```
(%i13) 'integrate('integrate((f2(x,y)-f1(x,y))*J,phi,0,2*%pi),r,0,2);
```

(%o13) $\displaystyle\int_0^2 r \int_0^{2\pi} -\sqrt{\sin(\phi)^2 r^2 + \cos(\phi)^2 r^2} - \sin(\phi)^2 r^2 - \cos(\phi)^2 r^2 + 6\, d\phi\, dr$

```
(%i14) trigsimp(%),factor;
```

(%o14) $\displaystyle -2\pi \int_0^2 r\,(r^2 + r - 6)\, dr$

Po zastosowaniu polecenia upraszczającego **trigsimp** wyrażenie pod całką nie zależy już od φ, a całka zostaje policzona automatycznie.

```
(%i15) %,integrate;
```

(%o15) $\dfrac{32\pi}{3}$

Przykład 5

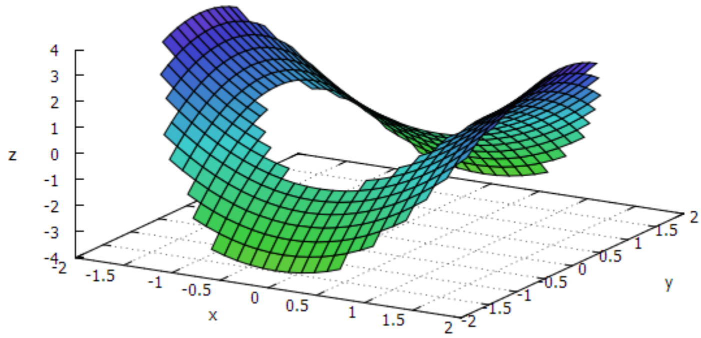
Obliczymy pole powierzchni płata

$$S = \{(x, y, z) \in \mathbb{R}^3\colon\ z = x^2 - y^2 \ \wedge\ 1 \leq x^2 + y^2 \leq 4\}.$$

Aby narysować płat S, najpierw (korzystając z konstrukcji warunkowej) definiujemy funkcję $f = f(x, y)$, która jest obcięciem funkcji $z = x^2 - y^2$ do zbioru

$$D = \{(x, y) \in \mathbb{R}^2 \colon 1 \leq x^2 + y^2 \leq 4\}.$$

```
(%i1) z(x,y):=x^2-y^2$

(%i2) f(x,y):=if x^2+y^2<4 and x^2+y^2>1 then z(x,y)$

(%i3) plot3d([f, [x,-2,2], [y,-2,2]],[gnuplot_pm3d,false])$
```



Przypomnijmy, że

$$|S| = \iint\limits_D \sqrt{1 + (z'_x)^2 + (z'_y)^2}\, dx\, dy$$

Funkcja podcałkowa przyjmuje więc postać:

```
(%i4) derivabbrev:true$

(%i5) sqrt(1+diff('z(x,y),x)^2+diff('z(x,y),y)^2);
      %,nouns;
```

(%o5) $\sqrt{\left(z(x,y)_y\right)^2 + \left(z(x,y)_x\right)^2 + 1}$

(%o6) $\sqrt{4\,y^2 + 4\,x^2 + 1}$

W rozważanym przykładzie zbiór D jest pierścieniem, więc aby obliczyć całkę podwójną, dokonamy zamiany współrzędnych kartezjańskich x i y na współrzędne biegunowe r i φ.

Jakobian tego przekształcenia jest równy r. Jakobian ten oraz jakobiany innych przekształceń wyznaczaliśmy w poprzednim przykładzie oraz w rozdziale 8.4 (przykład 3).

```
(%i7) %,x=r*cos(phi),y=r*sin(phi);
      trigsimp(%);
      'integrate('integrate(%*r, r, 1, 2),phi,0,2*%pi);
```

(%o7) $\sqrt{4\sin(\phi)^2 r^2 + 4\cos(\phi)^2 r^2 + 1}$

(%o8) $\sqrt{4r^2+1}$

(%o9) $2\pi \int_1^2 r\sqrt{4r^2+1}\,dr$

Na końcu podamy trzy różne postacie wyniku:

```
(%i10) %,nouns;%,factor;%,bfloat,fpprec:5;
```

(%o10) $2\left(\frac{17^{3/2}}{12}-\frac{5^{3/2}}{12}\right)\pi$

(%o11) $\frac{(\sqrt{17}-\sqrt{5})(\sqrt{5}\sqrt{17}+22)\pi}{6}$

(%o12) 3.0847b1

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Obliczyć całkę podwójną:

a) $\iint_D y dxdy\,,\ D = \{(x,y) \in \mathbb{R}^2 \colon -x \le y \le 2x - x^2\}$;
b) $\iint_D \ln x\, dxdy\,,\ D = \{(x,y) \in \mathbb{R}^2 \colon x + 1 \le 2y \le 2x,\ x \le 2\}$;
c) $\iint_D e^y dxdy\,,\ D = \{(x,y) \in \mathbb{R}^2 \colon y^2 \le x \le 2y\}$;
d) $\iint_D y^2 dxdy\,,\ D = \{(x,y) \in \mathbb{R}^2 \colon x^2 + y^2 \le 1,\ x \ge 0\}$.

Zadanie 2

Obliczyć objętość bryły ograniczonej powierzchniami:

a) $z = 6 - 2x - 3y,\ \ x = 0,\ \ y = 0,\ \ z = 0$;
b) $z = x^2,\ \ z = 0,\ \ x = 0,\ \ x = 1,\ \ y = 0,\ \ y = 1$;
c) $z = 8 - x^2 - y^2\,,\ \ z = -1$;
d) $z = \sqrt{x^2+y^2}\,,\ \ z = 4,\ \ x^2 + y^2 = 1,\ \ x^2 + y^2 = 4$.

Zadanie 3

Obliczyć pole powierzchni płata S, gdy:

a) $S = \{(x,y,z) \in \mathbb{R}^3 \colon\ z = \sqrt{x^2+y^2} \wedge x^2 + y^2 \le 9\}$;
b) $S = \{(x,y,z) \in \mathbb{R}^3 \colon\ z = x^2 + y^2 \wedge x^2 + y^2 \le 9\}$.

Odpowiedzi

1. a) -2.7, b) $\frac{1}{8}$, c) 4, d) $\frac{\pi}{8}$.
2. a) 6, b) $\frac{1}{3}$, c) $\frac{81}{2}\pi$, d) $\frac{22}{3}\pi$.
3. a) $9\sqrt{2}\,\pi$, b) $\frac{\pi}{6}\left(\sqrt{37^3} - 1\right)$.

18. Równania różniczkowe zwyczajne

Tabela 18.1

POLECENIA	OPIS
ode2(rrz,y,x**)**	rozwiązuje równania różniczkowe zwyczajne rzędu I oraz II względem zmiennej niezależnej x oraz zmiennej zależnej y
ic1(rozw,x=x0,y=y0**)**	rozwiązuje zagadnienie początkowe z warunkiem $y(x_0)=y_0$ dla równania rzędu I [1)]
ic2(rozw,x=x0,y=y0, v=v0**)**	rozwiązuje zagadnienie początkowe z warunkami $y'(x_0)=v_0$, $y(x_0)=y_0$ dla równania rzędu II [1)]
bc2(rozw,x=x1,y=y1, x=x2,y=y2**)**	rozwiązuje zagadnienie brzegowe z warunkami $y(x_1)=y_1$, $y(x_2)=y_2$ dla równania rzędu II [1)]
depends(y,x**)**	wprowadza zależność funkcyjną $y=y(x)$ pomiędzy zmiennymi x i y
%c	stała całkowania w rozwiązaniu równania I rzędu [2)]
%k1, %k2	stałe całkowania w rozwiązaniu równania II rzędu [2)]
method	przechowuje informację o ostatnio zastosowanej metodzie rozwiązywania równania różniczkowego przy pomocy polecenia **ode2** [3)]

[1)] Pierwszym argumentem funkcji (poleceń) **ic1**, **ic2** oraz **bc2** jest rozwiązanie równania różniczkowego uzyskane za pomocą polecenia **ode2**.
[2)] Stałe %c, %k1,%k2 pojawiają się w rozwiązaniach ogólnych.
[3)] Nazwa metody pokrywa się z nazwą typu równania różniczkowego (patrz tabela 18.2 oraz tabela 18.3).

Podstawowym poleceniem stosowanym do rozwiązywania równań różniczkowych zwyczajnych I oraz II rzędu jest polecenie **ode2**. Wprowadzając równanie różniczkowe, należy pamiętać, że pochodne muszą być zapisane symbolicznie. Możemy to zrobić na trzy różne sposoby:

- używając operatora ʻ przed pochodną, tzn. zamiast y' w równaniu piszemy **ʻdiff(y,x)**,
- wcześniej deklarując zależność między zmiennymi np. **depends(y,x)** i wtedy wystarczy zapisać **diff(y,x)**,
- określając zależność między zmiennymi w samym zapisie pochodnej np. **diff(y(x),x)**.

W rozważanych poniżej przykładach najczęściej będziemy stosować pierwszą z wymienionych notacji.

Do rysowania krzywych całkowych równania różniczkowego będziemy stosować polecenie **plot2d** oraz polecenia pakietów **drawdf** i **plotdf**.

Tabela 18.2. Stosowane metody dla równań I rzędu

linear	równanie liniowe
separable	równanie o zmiennych rozdzielonych
homogenous	równanie jednorodne względem x i y
exact	równanie zupełne
intfactor	podaje czynnik całkujący (po rozwiązaniu równania zupełnego)
bernoulli	równanie Bernoulliego

Tabela 18.3. Stosowane metody dla równań II rzędu

constcoef	równanie liniowe jednorodne o stałych współczynnikach
variationofparameters	równanie liniowe
freeofy	równanie „bez y” tzn. bez zmiennej y występującej w sposób jawny
freeofx	równanie „bez x” tzn. bez zmiennej x występującej w sposób jawny
euler	równanie Eulera

Tabela 18.4

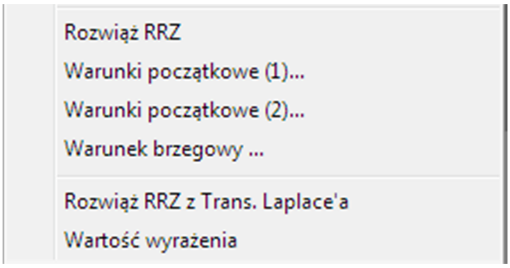
POLECENIA	OPIS
laplace(f(t),t,s**)**	obraz funkcji w przekształceniu Laplace’a
ilt(F(s),s,t**)**	oryginał funkcji w przekształceniu Laplace’a
desolve(u,zm**)**	rozwiązuje równania i układy dwóch i więcej równań różniczkowych liniowych oraz niektóre typy równań wyższych rzędów korzystając z transformacji Laplace’a
atvalue(y(x),x=x_0,y_0**)** **atvalue(**diff(y(x),x),x=x_0,v_0**)**	deklaracje warunków początkowych do polecenia **desolve**, należy je wczytać przed rozwiązywaniem równania
printprops([zm], atvalue**)**	wyświetla warunki z **atvalue**

W przypadku, gdy nie potrafimy wyznaczyć rozwiązania równania różniczkowego $y' = f(x,y)$ w postaci dokładnej, możemy narysować pole wektorowe związane z tym równaniem i na jego podstawie określić w przybliżeniu przebieg rozwiązań. Wektory bądź odcinki tworzące pole mają tę samą długość i są nachylone do osi Ox pod kątem, którego tangens wynosi $f(x,y)$. Taką reprezentację nazywamy polem kierunków równania różniczkowego. Do rysowania pola kierunków równania różniczkowego I rzędu (lub układu autonomicznego) można zastosować polecenie **plotdf**:

plotdf(f(x,y),[trajectory_at,x0,y0],[y,ymin,ymax],[x,xmin,xmax],*opcje***)**.

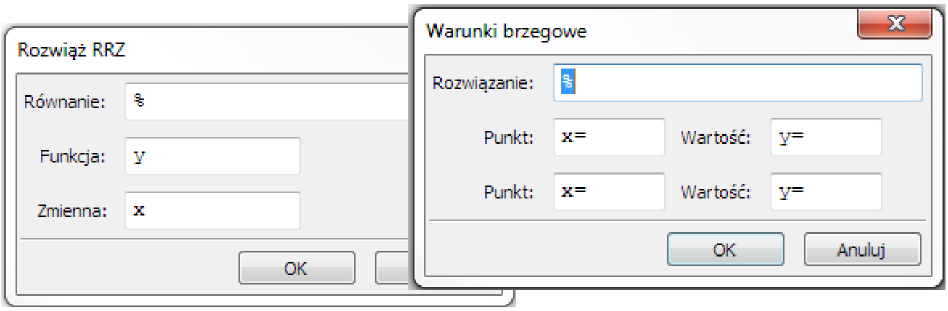
Podobnie działa polecenie **drawdf**, ale dostępne jest dopiero po wczytaniu pakietu o tej samej nazwie.

Polecenia dostępne w **Menu-Równania** oraz ustawienia domyślne formularzy.



ode2(%,y,x**)**
ic1(%,x=…,y=…**)**
ic2(%,x=… ,y=… ,'diff(y,x)=… **)**
bc2(%,x=… ,y=… ,x=… ,y=…**)**

desolve([%],[y(x)]**)**
atvalue(%,x=0,0**)**



Rysunek 18.1. **Wybrane formularze**

Uwaga 1
W poniższych przykładach, gdy nie zaznaczymy inaczej, rozumiemy, że pojawiające się w rozwiązaniach $\%c$, $\%k1$, $\%k2$, C_1, C_2, to dowolne stałe rzeczywiste.

ZADANIA PRZYKŁADOWE

Przykład 1
Wyznaczymy rozwiązanie ogólne równania $y' - 2xy = 2xe^{x^2}$.

```
(%i1) ode2('diff(y,x)-2*x*y=2*x*exp(x^2), y, x);
```

(%o1) $y = (x^2 + \%c)\,\%e^{x^2}$

Przykład 2
Wyznaczymy rozwiązania równania $y' + 2xy^2 = 0$. Wynik przedstawimy w postaci jawnej.

```
(%i1) ode2('diff(y,x)+2*x*y^2=0, y, x);
```

(%o1) $\frac{1}{2\,y} = \frac{x^2}{2} + \%c$

```
(%i2) solve(%,y);
```

$$(\%o2)\ [y=\frac{1}{x^2+2\ \%c}]$$

W rozwiązaniu (%o2) należy oczywiście założyć, że $x^2 + 2\%c \neq 0$.
Zauważmy też, że $y = 0$ jest rozwiązaniem szczególnym rozważanego równania (nieuwzględnionym w (%o2)).

Przykład 3

Wyznaczymy rozwiązanie szczególne równania $y' + y = x^2$ spełniające warunek początkowy $y(0) = 3$.

```
(%i1) rrz:'diff(y,x)+y=x^2;
```

$$(\%o1)\ \frac{d}{d\,x}\,y+y=x^2$$

Metoda I

Rozwiążemy powyższe zagadnienie początkowe, korzystając z poleceń **ode2** oraz **ic1**. Zastosujemy też polecenie **expand**.

```
(%i2) rozw:ode2(rrz, y, x);
      rozw:rozw,expand;
```

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

$$(\%o3)\ y=\%c\ \%e^{-x}+x^2-2\,x+2$$

```
(%i4) ic1(rozw, x=0, y=3);
      %,expand;
```

$$(\%o4)\ y=\%e^{-x}\left(\left(x^2-2\,x+2\right)\%e^{x}+1\right)$$

$$(\%o5)\ y=\%e^{-x}+x^2-2\,x+2$$

Metoda II

Tym razem skorzystamy z poleceń **desolve** oraz **atvalue**.

```
(%i6) rrz:'diff(y(x),x)+y(x)=x^2;
```

$$(\%o6)\ \frac{d}{d\,x}\,y(x)+y(x)=x^2$$

```
(%i7) atvalue(y(x), x=0, 3)$
      desolve(rrz,y(x));
```

$$(\%o8)\ y(x)=\%e^{-x}+x^2-2\,x+2$$

Uwaga 2

Składnia poleceń **desolve** i **atvalue** wymaga zapisania zmiennej zależnej w postaci $y(x)$. Ponadto warunek początkowy może być ustalany jedynie dla $x = 0$ oraz określenie tego warunku (za pomocą **atvalue**) powinno wystąpić przed rozwiązaniem równania. Polecenie **desolve** wykorzystuje transformatę Laplace'a (patrz też tabela 16.4) i można je stosować do pewnych typów układów równań różniczkowych liniowych oraz do równań różniczkowych liniowych wyższych rzędów z odpowiednio dobranymi warunkami.

Przykład 4

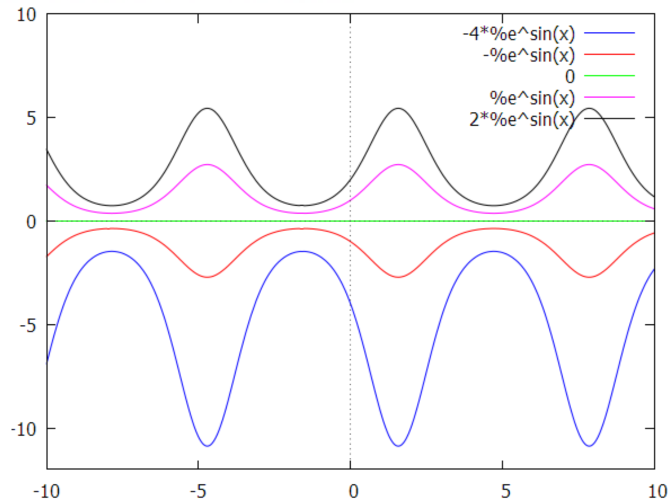
Wyznaczymy całkę ogólną równania $y' - y\cos x = 0$ i naszkicujemy kilka krzywych całkowych tego równania.

```
(%i1) rownanie: 'diff(y,x)-cos(x)*y=0;
      rozwiazanie: ode2( rownanie, y, x);
```

(%o1) $\frac{d}{d\,x}\,y - \cos(x)\,y = 0$

(%o2) $y = \%c\,\%e^{\sin(x)}$

```
(%i3) rhs(rozwiazanie),%c=[-4,-1,0,1,2]$
      plot2d(%, [x,-10,10],[y,-12,10])$
```



Przykład 5

Znajdziemy rozwiązanie ogólne równania $xy' = y + x\ln x$ i sprawdzimy, jaka metoda została zastosowana w Maximie do jego uzyskania, tzn. jaki typ równania rozpoznał program.

```
(%i1) ode2(x*'diff(y,x)=y+x*log(x), y, x);
      %,expand;
```

(%o1) $y = x\left(\frac{\log(x)^2}{2} + \%c\right)$

(%o2) $y = \frac{x\log(x)^2}{2} + \%c\,x$

W (%o1) i (%o2) są różne sposoby przedstawienia wyniku. Zakładamy, że $x > 0$, a stała $\%c \in \mathbb{R}$.

```
(%i3) method;
```

(%o3) *linear*

Jest to równanie różniczkowe liniowe I rzędu.

Uwaga 3

W przypadku gdy Maxima nie potrafi rozwiązać danego równania różniczkowego lub układu równań, w odpowiedzi pojawia się **false**.

Przykład 6

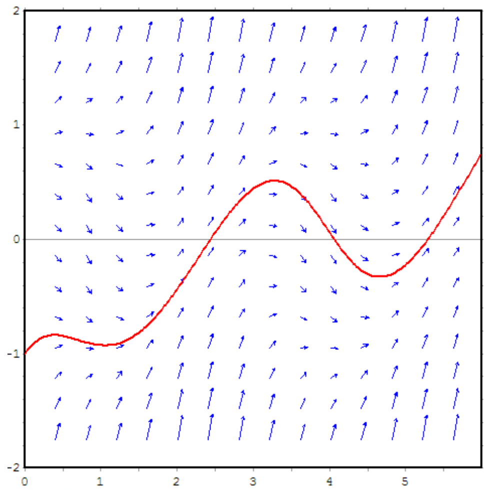
Sprawdzimy, że dla zagadnienia początkowego: $y' = y^2 - \sin 2x\,,\ \ y(0) = -1$ nie otrzymamy rozwiązania analitycznego.

```
(%i1) ode2('diff(y,x)=y^2-sin(2*x), y, x);
(%o1) false
```

```
(%i2) method;
(%o2) none
```

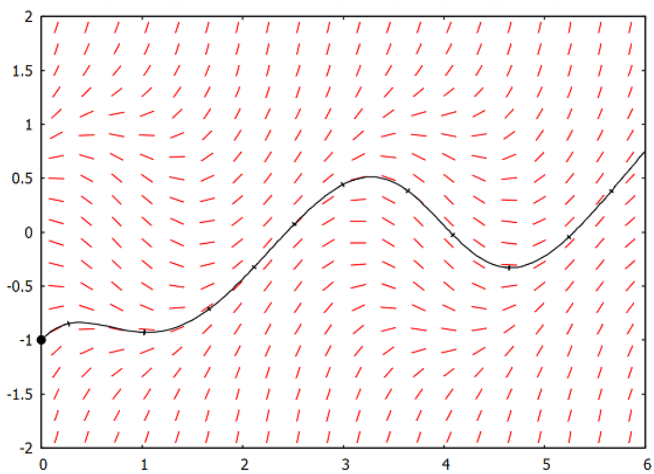
Możemy jednak, na bazie metod numerycznych, przedstawić rozwiązanie graficzne tego zagadnienia. Polecenia **plotdf** oraz **drawdf** dają możliwość wygenerowania pola kierunków tego równania i zaznaczenia krzywej całkowej spełniającej podany warunek początkowy.

```
(%i3) plotdf(y^2-sin(2*x),[trajectory_at,0,-1],[y,-2,2],[x,0,6])$
```



```
(%i4) load(drawdf)$
```

```
(%i5) drawdf(y^2-sin(2*x), [x,0,6], [y,-2,2], field_degree=1, field_grid=[20,20],
      soln_arrows=true, point_at(0,-1), solns_at([0,-1]))$
```



Okno graficzne dostępne po wykonaniu polecenia **plotdf** daje możliwość interakcji (m.in. dorysowywania kolejnych krzywych). Polecenie **drawdf** ma wiele dostępnych opcji (niektóre z nich zostały użyte powyżej).

Przykład 7

Rozwiążemy poniższe równania różniczkowe I rzędu i wskażemy wśród nich równania liniowe.

1) $y' = \frac{y}{\ln y}$,
2) $(2\ln x\ y + 1)y' + \frac{y^2}{x} = 0$,
3) $y' - 2xy = 2xe^{x^2}$,
4) $y' - y = 2 - x^2$,
5) $y' = \frac{4y}{x^2-1}$,
6) $y' + y = xy^2$,
7) $y' - y = e^x \cos x$,
8) $x^2 + x\,y' = y$.

Najpierw przypiszemy do zmiennej $r[i]$ odpowiednio i-te równanie. Następnie zastosujemy polecenie **genmatrix** do utworzenia macierzy (wymiaru 8×4), której pierwsza kolumna zawiera numery równań, druga – równania, trzecia – rozwiązania, natomiast czwarta – nazwę użytej metody. Przydatne też będą wyrażenie **lambda** i konstrukcja **if** (patrz rozdział 21).

```
(%i1) r[1]:'diff(y,x)=y/log(y)$
      r[2]:(1+2*y*log(x))*'diff(y,x)+y^2/x=0$
      r[3]:'diff(y,x)-2*x*y=2*x*exp(x^2)$
      r[4]:'diff(y,x)-y=2-x^2$
      r[5]:'diff(y,x)=4*y/(x^2-1)$
      r[6]:'diff(y,x)+y=x*y^2$
      r[7]:'diff(y,x)-y=exp(x)*cos(x)$
      r[8]:x^2+x*'diff(y,x)=y$
(%i9) genmatrix(lambda([i,j], if j=1 then concat(i,")") elseif j=2 then r[i]
      elseif j=3 then ev(ode2(r[i],y,x),factor) else  method), 8,4);
```

(%o9)
$$\begin{bmatrix}
1) & \frac{d}{dx}y = \frac{y}{\log(y)} & \frac{\log(y)^2}{2} = x + \%c & \mathit{separable} \\
2) & (2\log(x)\,y+1)\left(\frac{d}{dx}y\right) + \frac{y^2}{x} = 0 & y\,(\log(x)\,y+1) = \%c & \mathit{exact} \\
3) & \frac{d}{dx}y - 2\,x\,y = 2\,x\,\%e^{x^2} & y = (x^2+\%c)\,\%e^{x^2} & \mathit{linear} \\
4) & \frac{d}{dx}y - y = 2 - x^2 & y = \%c\,\%e^{x} + x^2 + 2\,x & \mathit{linear} \\
5) & \frac{d}{dx}y = \frac{4\,y}{x^2-1} & y = \frac{\%c\,(x-1)^2}{(x+1)^2} & \mathit{linear} \\
6) & \frac{d}{dx}y + y = x\,y^2 & y = \frac{1}{\%c\,\%e^{x} + x + 1} & \mathit{bernoulli} \\
7) & \frac{d}{dx}y - y = \%e^{x}\cos(x) & y = \%e^{x}(\sin(x) + \%c) & \mathit{linear} \\
8) & x\left(\frac{d}{dx}y\right) + x^2 = y & y = -x\,(x - \%c) & \mathit{linear}
\end{bmatrix}$$

Równaniami liniowymi są więc równania 3), 4), 5), 7) oraz 8).

Przykład 8

Znajdziemy rozwiązania ogólne równań różniczkowych liniowych jednorodnych II rzędu (o stałych współczynnikach):

a) $y'' - 4y' + 4y = 0$, c) $y'' + 3y' = 0$,

b) $y'' - 4y' + 3y = 0$, d) $y'' + 3y = 0$.

Oznaczymy równania zgodnie z ich ustawieniem alfabetycznym i rozwiążemy je, stosując polecenie **ode2**:

```
(%i1) rra:'diff(y,x,2)-4*'diff(y,x)+4*y=0; ode2(rra,y,x),expand;
```

(%o1) $\frac{d^2}{d\,x^2}\,y - 4\left(\frac{d}{d\,x}\,y\right) + 4\,y = 0$

(%o2) $y = \%k2\;x\;\%e^{2\,x} + \%k1\;\%e^{2\,x}$

```
(%i3) rrb:'diff(y,x,2)-4*'diff(y,x)+3*y=0; ode2(rrb,y,x);
```

(%o3) $\frac{d^2}{d\,x^2}\,y - 4\left(\frac{d}{d\,x}\,y\right) + 3\,y = 0$

(%o4) $y = \%k1\;\%e^{3\,x} + \%k2\;\%e^{x}$

```
(%i5) rrc:'diff(y,x,2)+3*'diff(y,x)=0; ode2(rrc,y,x);
```

(%o5) $\frac{d^2}{d\,x^2}\,y + 3\left(\frac{d}{d\,x}\,y\right) = 0$

(%o6) $y = \%k2\;\%e^{-3\,x} + \%k1$

```
(%i7) rrd:'diff(y,x,2)+3*y=0; ode2(rrd,y,x);
```

(%o7) $\frac{d^2}{d\,x^2}\,y + 3\,y = 0$

(%o8) $y = \%k1\,\sin(\sqrt{3}\,x) + \%k2\,\cos(\sqrt{3}\,x)$

Podpunkt b) rozwiążemy też metodą „krok po kroku”.

```
(%i9) w_rchar:['diff(y,x,2)=r^2,'diff(y,x)=r,y=1]$
      rrb,w_rchar;
      solve(%);
      map(rhs,%);
      map(exp,%*x);
      map("*",[C1,C2],%);
      yo=apply("+",%);
```

(%o10) $r^2 - 4\,r + 3 = 0$

(%o11) $[r = 3, r = 1]$

(%o12) $[3, 1]$

(%o13) $[\%e^{3\,x}, \%e^{x}]$

(%o14) $[\%e^{3\,x}\,C1, \%e^{x}\,C2]$

(%o15) $yo = \%e^{x}\,C2 + \%e^{3\,x}\,C1$

W równaniu *rrb* wykonujemy podstawienia zgodnie z warunkami *w_rchar*, budując w ten sposób równanie charakterystyczne.
Następnie wyznaczamy jego rozwiązania.
Polecenia **map** i **apply** pozwalają otrzymać postać rozwiązania ogólnego y_o.

Stałe C_1 i C_2 są dowolnymi liczbami rzeczywistymi.

Przykład 9

Wyznaczymy rozwiązania ogólne równań różniczkowych liniowych niejednorodnych II rzędu:

a) $y'' + 2y' + 3y = xe^{-x}$, b) $y'' + 4y = \frac{1}{\cos 2x}$, c) $y'' - y' = 2\cos x - \sin x$.

a)

```
(%i1) rra:'diff(y,x,2)+2*'diff(y,x)+3*y=x*exp(-x); ode2(rra,y,x),expand;
```

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

(%o2) $y = \%k1\ \%e^{-x} \sin(\sqrt{2}\ x) + \%k2\ \%e^{-x} \cos(\sqrt{2}\ x) + \frac{x\ \%e^{-x}}{2}$

b)

```
(%i3) rrb:'diff(y,x,2)+4*y=1/cos(2*x); ode2(rrb,y,x),expand;
```

(%o3) $\frac{d^2}{d x^2} y + 4 y = \frac{1}{\cos(2 x)}$

(%o4) $y = \frac{\cos(2 x) \log(\cos(2 x))}{4} + \frac{x \sin(2 x)}{2} + \%k1 \sin(2 x) + \%k2 \cos(2 x)$

c) Ten przykład rozwiążemy metodą „krok po kroku”. Zaczniemy od zdefiniowania rozważanego równania różniczkowego (liniowego niejednorodnego).

```
(%i1) rrc:'diff(y,x,2)-'diff(y,x)=2*cos(x)-3*sin(x)$ display(rrc)$
```

$\frac{d^2}{d x^2} y - \frac{d}{d x} y = 2 \cos(x) - 3 \sin(x)$

1) Zajmiemy się teraz równaniem liniowym jednorodnym $y'' - y' = 0$, które możemy otrzymać z powyższego, stosując polecenie **lhs**. Podobnie jak w przykładzie 8, zbudujemy równanie charakterystyczne, wyznaczymy jego pierwiastki oraz funkcje stanowiące układ fundamentalny: $y_1 = 1$, $y_2 = e^x$, a na koniec ich liniową kombinację (polecenia **map** oraz **apply**).

```
(%i3) w_rchar:['diff(y,x,2)=r^2,'diff(y,x)=r,y=1]$
      lhs(rrc)=0,w_rchar;
      solve(%);
      map(rhs,%);
      map(exp,%*x);
      map("*",[C1,C2],%);
      yo=apply("+",%);
```

(%o4) $r^2 - r = 0$

(%o5) $[r = 0, r = 1]$

(%o6) $[0, 1]$

(%o7) $[1, \%e^x]$

(%o8) $[C1, \%e^x\ C2]$

(%o9) $yo = \%e^x\ C2 + C1$

W (%o9) otrzymujemy rozwiązanie ogólne równania jednorodnego. Stałe C_1 i C_2 są dowolnymi liczbami rzeczywistymi.

2) Kolejny etap, to wyznaczenie rozwiązania szczególnego równania niejednorodnego. Zastosujemy metodę przewidywania. Zgodnie z postacią prawej strony równania rrc mamy:

```
(%i10) ys:A*sin(x)+B*cos(x)$ display(ys)$
       rrc,y=ys$
       w:ev(%,diff);
       rat(%,cos(x),sin(x));
       makelist(ratcoeff(w,i),i,[cos(x),sin(x)]);
       solve(%);
       ys:ys,%$
       display(ys)$
```

$ys = \cos(x)\,B + \sin(x)\,A$

(%o13) $\sin(x)\,B - \cos(x)\,B - \sin(x)\,A - \cos(x)\,A = 2\cos(x) - 3\sin(x)$

(%o14)/R/ $(B - A)\sin(x) + (-B - A)\cos(x) = -3\sin(x) + 2\cos(x)$

(%o15) $[-B - A = 2, B - A = -3]$

(%o16) $[[B = -\frac{5}{2}, A = \frac{1}{2}]]$

$ys = \frac{\sin(x)}{2} - \frac{5\cos(x)}{2}$

Po podstawieniu funkcji y_s oraz y_s' do wyjściowego równania rrc, jego lewa strona została uporządkowana względem sinusów i cosinusów dzięki poleceniu **rat**. Porównanie współczynników (polecenia: **ratcoeff**, **makelist**) i rozwiązanie układu równań pozwoliło wyznaczyć wartości stałych A i B oraz ostatecznie y_s.

3) Ostatni etap, to sumowanie rozwiązań y_o oraz y_s. Zamiast sumowania zastosujemy polecenie **ode2**, aby otrzymać rozwiązanie ogólne równania niejednorodnego.

```
(%i19) ode2(rrc,y,x),expand;
```

(%o19) $y = \frac{\sin(x)}{2} - \frac{5\cos(x)}{2} + \%k1\,\%e^{x} + \%k2$

Przykład 10

Rozwiążemy następujące zagadnienie początkowe

$$y'' + 2y = 2\sin x,\;\; y(0) = 0,\;\;\; y'(0) = -1$$

oraz naszkicujemy otrzymaną krzywą całkową z zaznaczonym punktem (0,0) i styczną w tym punkcie.

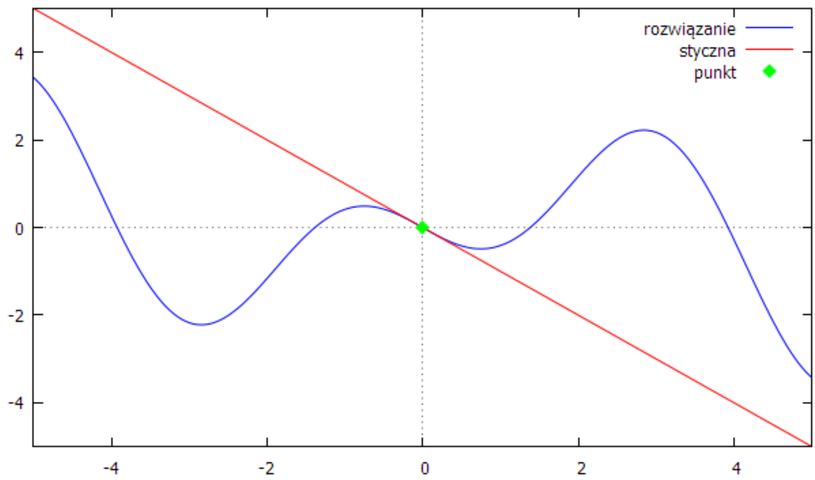
```
(%i1) ode2('diff(y,x,2)+2*y=2*sin(x), y, x);
```

(%o1) $y = \%k1\sin(\sqrt{2}\,x) + \%k2\cos(\sqrt{2}\,x) + 2\sin(x)$

```
(%i2) cs:ic2(%, x=0, y=0, 'diff(y,x)=-1);
```

(%o2) $y = 2\sin(x) - \frac{3\sin(\sqrt{2}\,x)}{\sqrt{2}}$

```
(%i3) plot2d([rhs(cs),-x,[discrete, [[0],[0]]]], [x,-5,5], [y,-5,5],
             [style, lines, lines, points], [point_type, diamond],
             [legend, "rozwiązanie", "styczna", "punkt"])$
```



Przykład 11

Rozwiążemy następujące zagadnienie brzegowe

$$y'' + 2y' + y = 2e^{-x},\ \ y(0) = 0,\ \ \ y(1) = \frac{1}{e}$$

i narysujemy krzywą całkową z zaznaczonymi z punktami: (0,0), $\left(1, \frac{1}{e}\right)$.

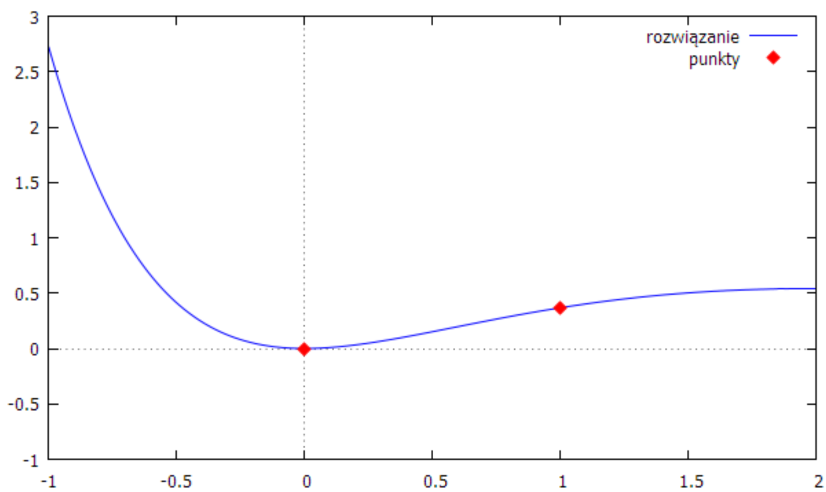
```
(%i1) ode2('diff(y,x,2)+2*'diff(y,x)+y=2*exp(-x), y, x);
```

(%o1) $y = x^2\ \%e^{-x} + (\%k2\ x + \%k1)\ \%e^{-x}$

```
(%i2) cs:bc2(%,x=0,y=0,x=1,y=1/%e);
```

(%o2) $y = x^2\ \%e^{-x}$

```
(%i3) plot2d([rhs(cs),[discrete, [[0,0],[1,1/%e]]]], [x,-1,2], [y,-1,3],
             [style, lines, points], [point_type, diamond],
             [legend, "rozwiazanie", "punkty"])$
```



Przykład 12

Rozwiążemy następujące zagadnienie początkowe dla równania III rzędu:

$$y''' - y' = x + 2,\ \ y(0) = 1,\ \ y'(0) = 2,\ \ y''(0) = -1.$$

Polecenie **ode2** można stosować tylko do równań I i II rzędu, więc użyjemy polecenia **desolve**, które „radzi sobie" z równaniami liniowymi wyższych rzędów. Pamiętać też musimy o zapisie zmiennej zależnej y w postaci funkcyjnej $\boldsymbol{y(x)}$.

```
(%i1) rr:diff(y(x),x,3)-diff(y(x),x)=x+2;
      atvalue(y(x), x=0, 1)$
      atvalue(diff(y(x),x), x=0, 2)$
      atvalue(diff(y(x),x,2), x=0,-1)$
      desolve(rr, y(x));
```

(%o1) $\frac{d^3}{d x^3} y(x) - \frac{d}{d x} y(x) = x + 2$

(%o5) $y(x) = 2\,\%e^{x} - 2\,\%e^{-x} - \frac{x^2}{2} - 2\,x + 1$

Przykład 13

Rozwiążemy układ równań różniczkowych liniowych niejednorodnych I rzędu z podanymi warunkami początkowymi:

$$\begin{cases} y' = 2y - z - t \\ z' = y - 2\sin t \end{cases},\qquad \begin{cases} y(0) = 1 \\ z(0) = 1 \end{cases}.$$

```
(%i1) rr1:'diff(y(t),t)=2*y(t)-z(t)-t$
      rr2:'diff(z(t),t)=y(t)-2*sin(t)$
      atvalue(y(t),t=0,1)$ atvalue(z(t),t=0,1)$
      roz:desolve([rr1,rr2], [y(t), z(t)])$
      print(rr1)$ display(rr2)$
      printprops([y,z],atvalue)$
      print(roz)$
```

$\frac{d}{d t} y(t) = -z(t) + 2\,y(t) - t$

$\frac{d}{d t} z(t) = y(t) - 2\sin(t)$

$y(0) = 1$

$z(0) = 1$

$[y(t) = \cos(t) + \%e^{t} - 1, z(t) = \sin(t) + 2\cos(t) + \%e^{t} - t - 2]$

Przykład 14

Przedstawimy poniżej kilka przekształceń i własności związanych z transformatą Laplace'a.

W literaturze, dla transformaty Laplace'a najczęściej są stosowane oznaczenia: $F(s) = \mathcal{L}\{f(t)\}$, bądź $F(s) = L[f(t)]$, gdzie $t \in \mathbb{R}, s \in \mathbb{C}$, a dla transformaty odwrotnej: $\mathcal{L}^{-1}\{F(s)\}$, bądź $L^{-1}[F(s)]$.
Natomiast w Maximie odpowiednio **laplace(**f(t),t,s**)** oraz **ilt(**F(s),s,t**)**.

Polecenia (funkcje) **laplace** oraz **ilt** możemy znaleźć w **Menu-RRC**. Uzupełniamy wtedy okna odpowiedniego formularza, wprowadzając funkcję i zmienne.

Transformata Laplace'a ma własność liniowości:

```
(%i1) laplace(a*f(x)+b*g(x), x, s);
```

(%o1) $b\,\mathrm{laplace}(\mathrm{g}(x),x,s)+a\,\mathrm{laplace}(\mathrm{f}(x),x,s)$

Poniżej obliczymy transformaty Laplace'a dla wybranych funkcji:

```
(%i2) laplace(x*exp(x), x, s);
```

(%o2) $\dfrac{1}{(s-1)^2}$

$$L[xe^x]=\frac{1}{(s-1)^2}$$

```
(%i3) laplace(exp(x)*cos(3*x), x, s);
```

(%o3) $\dfrac{s-1}{s^2-2\,s+10}$

$$L[e^x\cos 3x]=\frac{s-1}{s^2-2s+10}$$

```
(%i4) ilt((s+1)/(s^2+1), s, x);
```

(%o4) $\sin(x)+\cos(x)$

$$L^{-1}\left[\frac{s+1}{s^2+1}\right]=\sin x+\cos x$$

```
(%i5) ilt((2*s+3)/(s^2-4*s),s,x);
```

(%o5) $\dfrac{11\,\%e^{4\,x}}{4}-\dfrac{3}{4}$

$$L^{-1}\left[\frac{2s+3}{s^2-4s}\right]=\frac{11}{4}e^x-\frac{3}{4}$$

Tablica przekształceń Laplace'a wygenerowana w Maximie.

```
(%i6) l1:[1,x,x^2,exp(a*x),sin(b*x),cos(b*x)]$
      l0:makelist(i,i,1,length(l1))$
      l2:makelist(rat(makefact((laplace(f,x,s)))),f,l1)$
      text:matrix(["  Lp.  ","  Oryginał f(x)  ","  Obraz L[(f(x)]   "])$
```

```
(%i11) addrow(text,transpose(matrix(l0,l1,l2)));
```

(%o11)/R/
$$\begin{bmatrix} \text{Lp.} & \text{Oryginał f(x)} & \text{Obraz L[(f(x)]} \\ 1 & 1 & \frac{1}{s} \\ 2 & x & \frac{1}{s^2} \\ 3 & x^2 & \frac{2}{s^3} \\ 4 & \%e^{a\,x} & \frac{1}{s-a} \\ 5 & \sin(b\,x) & \frac{b}{s^2+b^2} \\ 6 & \cos(b\,x) & \frac{s}{s^2+b^2} \end{bmatrix}$$

Poniżej wyznaczamy $L[y'(x)]$ oraz $L[y''(x)]$. Wywołanie **rat**(%,s) spowoduje uporządkowanie wyrażenia % według potęg zmiennej s.

```
(%i11) laplace(diff(y(x),x,1),x,s);
```
(%o11) $s\,\mathrm{laplace}(y(x),x,s)-y(0)$

```
(%i12) laplace(diff(y(x),x,2),x,s);rat(%,s);
```
(%o12) $-\left.\frac{d}{d\,x}y(x)\right|_{x=0}+s^2\,\mathrm{laplace}(y(x),x,s)-y(0)\,s$

(%o13)/R/ $\mathrm{laplace}(y(x),x,s)\,s^2-y(0)\,s-\left.\frac{d}{d\,x}y(x)\right|_{x=0}$

Przykład 15

Stosując transformatę Laplace'a wyznaczymy rozwiązanie szczególne równania $y'-y=x^2$ spełniające warunek początkowy $y(0)=1$.

Najpierw definiujemy równanie:

```
(%i1) rr:'diff(y(x),x)-y(x)=x^2;
```
(%o1) $\frac{d}{d\,x}y(x)-y(x)=x^2$

Polecenie **printprops** pozwala na wyświetlenie warunku początkowego.

```
(%i2) atvalue(y(x),x=0,1)$ printprops(y,atvalue)$
```
$y(0)=1$

Następnie stosujemy przekształcenie Laplace'a do obu stron równania *rr*.

```
(%i4) r1:laplace(rr,x,s);
```
(%o4) $s\,\mathrm{laplace}(y(x),x,s)-\mathrm{laplace}(y(x),x,s)-1=\frac{2}{s^3}$

```
(%i5) r2:linsolve(r1,'laplace(y(x),x,s));
```
(%o5) $[\mathrm{laplace}(y(x),x,s)=\frac{s^3+2}{s^4-s^3}]$

Korzystając z polecenia **linsolve** wyznaczamy $L[y(x)]$ z równania $r1$.

```
(%i6) r3:r2[1],factor;
```
(%o6) $\mathrm{laplace}(y(x),x,s)=\frac{s^3+2}{(s-1)\,s^3}$

Stosujemy **factor**, aby otrzymać postać iloczynową prawej strony równania $r2$.

```
(%i7) r4:partfrac(r3,s);
```
(%o7) $\mathrm{laplace}(y(x),x,s)=-\frac{2}{s}-\frac{2}{s^2}-\frac{2}{s^3}+\frac{3}{s-1}$

Rozkładamy prawą stronę $r3$ na ułamki proste.

```
(%i8) ilt(r4,s,x);
```
(%o8) $y(x)=3\,\%e^x-x^2-2\,x-2$

Stosujemy transformatę odwrotną do równania $r4$ i uzyskujemy rozwiązanie.

Przykład 16

Zgodnie z prawem stygnięcia (prawem Newtona):
„Szybkość, z jaką układ (np. ciało, ciecz) stygnie, jest wprost proporcjonalna do różnicy temperatur pomiędzy układem a otoczeniem.”
Rozważmy jako układ - kubek z wodą nagrzaną do temperatury $80\,°C$ oraz przyjmijmy, że temperatura otoczenia wynosi $25\,°C$. Rozwiązując odpowiednie równanie różniczkowe, wyznaczymy w tym układzie zależność temperatury od czasu.

Prawo Newtona możemy zapisać jako następujące równanie różniczkowe:

$$\frac{dT}{dt} = -k\,(T - T_o),$$

gdzie zmienna t oznacza czas, T - temperaturę wody ($T = T(t)$), T_o - temperaturę otoczenia. Temperaturę początkową $80\,°C$ oznaczymy przez T_p i uwzględnimy, przy wyznaczaniu rozwiązania zagadnienia początkowego.

```
(%i1) depends(T,t);
(%o1) [T(t)]
(%i2) r:diff(T,t)=-k*(T-To);
```
$$(r)\quad \frac{d}{d\,t}\,T = -(T - To)\,k$$
```
(%i3) ode2(r,T,t),expand;
```
$$(\%o3)\quad T = \%c\,\%e^{-k\,t} + To$$
```
(%i5) To:25$ Tp:80$
(%i6) ode2(r,T,t),expand;
```
$$(\%o6)\quad T = \%c\,\%e^{-k\,t} + 25$$
```
(%i8) ic1(%,t=0,T=Tp)$
      rozwiązanie:%,expand;
```
$$(rozwiązanie)\quad T = 55\,\%e^{-k\,t} + 25$$

Rozwiązanie to zawiera stałą k, która może być znana (ustalona) dla danego układu albo można ją wyznaczyć wykonując dodatkowy pomiar temperatury po pewnym ustalonym czasie. Przyjmijmy, że otrzymany wynik to temperatura $T_{10} = 60\,°C$ otrzymana po 10 minutach.

Podstawiamy wartości i z otrzymanego równania wyznaczymy „krok po kroku” stałą k.

(%i10) T10:60$ fpprintprec:3$

(%i11) rozwiązanie,T=T10,t=10;

(%o11) $60 = 55\,\%e^{-10\,k} + 25$

(%i16) %-25;%/55;log(%);%/-10;wart_k:float(%);

(%o12) $35 = 55\,\%e^{-10\,k}$

(%o13) $\frac{7}{11} = \%e^{-10\,k}$

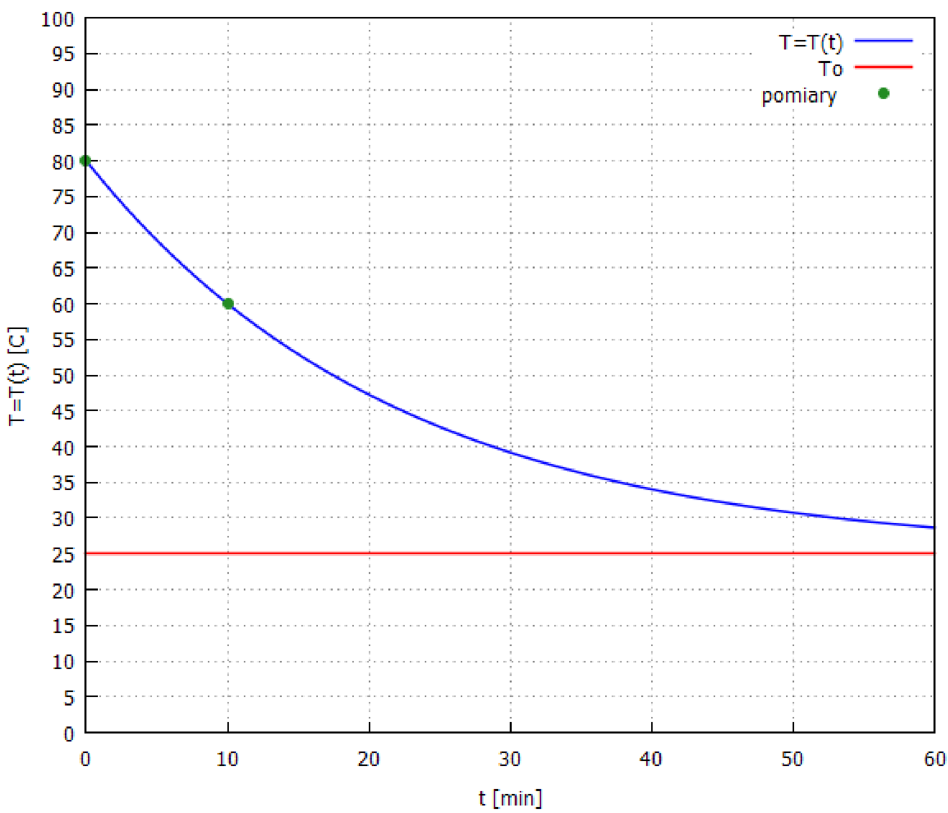
(%o14) $\log\left(\frac{7}{11}\right) = -10\,k$

(%o15) $-\frac{\log\left(\frac{7}{11}\right)}{10} = k$

(wart_k) $0.0452 = k$

(%i17) rys:rozwiązanie,k=lhs(wart_k);

(rys) $T = 55\,\%e^{-0.0452\,t} + 25$



```
(%i19)  load(draw)$
        draw2d( terminal = wxt, dimensions = [600,500],
                xrange = [0,60], yrange = [0,100],
                xlabel = "t [min]", ylabel = "T=T(t) [C]",
                ytics = 5, grid = true, line_width = 1.5,
                key = "T=T(t)", explicit(rhs(rys),t,0,60),
                color = red, key = "To", explicit(To,t,0,60),
                color = forest-green, key = "pomiary ", point_type = 7,
                points([0,10],[Tp,T10]) )$
```

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Znaleźć rozwiązanie ogólne równania różniczkowego I rzędu:

a) $y' = e^{2x}\, y^2$,

b) $y' = \frac{2\sqrt{y}}{x}$,

c) $y' = \frac{\cos x}{y}$,

d) $y' - 2y = (4x-1)e^{2x}$,

e) $y' = \frac{2y-x}{x}$,

f) $y' - \frac{y}{x} = \frac{1}{\ln x}$.

Zadanie 2

Znaleźć rozwiązanie szczególne równania różniczkowego I rzędu spełniające podany warunek początkowy:

a) $y' + y = 2\sin x,\ y(0) = -1$

b) $y' + 2xy = 2x^3,\ y(0) = 2$

Zadanie 3

Znaleźć rozwiązanie ogólne równania różniczkowego II rzędu:

a) $y'' + y' = 2e^x$,

b) $y'' + y = \sin x$.

Zadanie 4

Znaleźć rozwiązanie szczególne równania różniczkowego II rzędu spełniające podane warunki początkowe:

a) $y'' - y = 2x + 1,\ \ y(0) = 1,\ \ y'(0) = -2;$

b) $y'' + 2y' + y = 2e^{-x},\ \ y(0) = -1,\ \ y'(0) = 3.$

Zadanie 5

Znaleźć rozwiązanie szczególne równania różniczkowego II rzędu spełniające podane warunki brzegowe:

a) $y'' - 4y' + 5y = 0,\ \ y(0) = 1,\ \ y\left(\frac{\pi}{2}\right) = 0;$

b) $y'' - 4y' + 3y = 8e^{-x},\ \ y(0) = 1,\ \ y(-1) = e.$

Odpowiedzi

1. a) $-\frac{1}{y} = \frac{e^{2x}}{2} + C$, a po przekształceniu $y = -\frac{2}{e^{2x}+2C}$, oraz $y = 0$; b) $\sqrt{y} = \ln x + C$, dla $x > 0$, przy założeniu $\ln x + C > 0$, a po przekształceniu $y = (\ln x + C)^2$, oraz $y = 0$;
c) $y^2 = \sin x + C$, przy założeniu $\sin x + C \neq 0$; d) $y = (2x^2 - x + C)\, e^{2x}$;
e) $y = \left(\frac{1}{x} + C\right) x^2$, dla $x \neq 0$; f) $y = x\left(C - \frac{1}{\ln x}\right)$, dla $x \in (0,1) \cup (1, \infty)$.
W powyższych rozwiązaniach, o ile nie ma innej informacji, stała $C \in \mathbb{R}$.

2. a) $y = \sin x - \cos x$, b) $y = 3e^{-x^2} + x^2 - 1$.

3. a) $y = e^x + C_1 + C_2 e^{-x}$, $C_1, C_2 \in \mathbb{R}$; b) $y = -\frac{x\cos x}{2} + C_1 \sin x + C_2 \cos x$, $C_1, C_2 \in \mathbb{R}$;

4. a) $y = e^x + e^{-x} - 2x - 1$, b) $y = (x^2 + 2x - 1)e^{-x}$.

5. a) $y = e^{2x} \cos x$, b) $y = e^{-x}$.

19. Równania różnicowe

Tabela 19.1. Składnia poleceń pakietu **solve_rec**

POLECENIA	OPIS
solve_rec(r,y[n]**)**	rozwiązanie ogólne liniowego równania różnicowego r, w którym y_n jest szukanym ciągiem
solve_rec(r,y[n],war**)**	rozwiązanie szczególne liniowego równania różnicowego r rzędu m spełniające warunki brzegowe war: $y_0 = t_0, y_1 = t_1, \dots, y_{m-1} = t_{m-1}$, w którym y_n jest szukanym ciągiem

ZADANIA PRZYKŁADOWE

Na początku każdego przykładu wczytamy pakiet **solve_rec**.
Podobnie jak w rozwiązaniach ogólnych równań różniczkowych, tak i tu pojawią się stałe: $\%k_1$, $\%k_2$, $\%k_3$, $\%k_4$.

Przykład 1

Znajdziemy rozwiązania ogólne równań różnicowych jednorodnych:

a) $y_{n+1} - 7y_n = 0,$

b) $2y_{n+2} + y_{n+1} - 6y_n = 0,$

c) $y_{n+2} + 49y_n = 0,$

d) $y_{n+3} + y_{n+2} - 8y_{n+1} - 12y_n = 0,$

e) $y_{n+4} - 2y_{n+3} + 2y_{n+1} - y_n = 0.$

```
(%i1) load(solve_rec)$
```

```
(%i2) y[n+1]-7*y[n]=0;solve_rec(%,y[n]) /* 1 a) */;
```

(%o2) $y_{n+1} - 7\,y_n = 0$

(%o3) $y_n = \%k_1\, 7^n$

```
(%i4) 2*y[n+2]+y[n+1]-6*y[n]=0;solve_rec(%,y[n]) /* 1 b) */;
```

(%o4) $2\,y_{n+2} + y_{n+1} - 6\,y_n = 0$

(%o5) $y_n = \dfrac{\%k_1\, 3^n}{2^n} + \%k_2\,(-2)^n$

```
(%i6) y[n+2]+49*y[n]=0;solve_rec(%,y[n])  /* 1 c) */;
```

(%o6) $y_{n+2} + 49\,y_n = 0$

(%o7) $y_n = \%k_2\,(-1)^{n/2}\, 7^n + (-\%i)^n\, \%k_1\, 7^n$

Rozwiązanie ($\%o7$) przekształcimy, stosując polecenia **rectform** oraz **facsum** (patrz tabela 4.5, tabela 5.1).

```
(%i8) rectform(%o7)$ facsum(%,sin(n*%pi/2),cos(n*%pi/2));
```

$$(\%o9)\quad y_n = \%i\,(\%k_2 - \%k_1)\,7^n \sin\left(\frac{\pi n}{2}\right) + (\%k_2 + \%k_1)\,7^n \cos\left(\frac{\pi n}{2}\right)$$

Przyjmując w powyższym, że $C_1 = \%i\,(\%k_2 - \%k_1)$ oraz $C_2 = \%k_2 + \%k_1$, ostatecznie otrzymujemy: $y_n = C_1\, 7^n \sin(\frac{\pi n}{2}) + C_2\, 7^n \cos(\frac{\pi n}{2})$.

```
(%i10) y[n+3]+y[n+2]-8*y[n+1]-12*y[n]=0;solve_rec(%,y[n])  /* 1 d) */;
```

$$(\%o10)\quad y_{n+3} + y_{n+2} - 8\,y_{n+1} - 12\,y_n = 0$$

$$(\%o11)\quad y_n = \%k_1\, 3^n + (\%k_3\, n + \%k_2)(-2)^n$$

```
(%i12) y[n+4]-2*y[n+3]+2*y[n+1]-y[n]=0;solve_rec(%,y[n])  /* 1 e) */;
```

$$(\%o12)\quad y_{n+4} - 2\,y_{n+3} + 2\,y_{n+1} - y_n = 0$$

$$(\%o13)\quad y_n = \%k_1 (-1)^n + \%k_4\, n^2 + \%k_3\, n + \%k_2$$

Przykład 2

Znajdziemy rozwiązania szczególne równań różnicowych jednorodnych spełniających podane warunki brzegowe:

a) $y_{n+1} + 3y_n = 0,\ y_0 = 5$; b) $y_{n+2} - 36y_n = 0,\ y_0 = 1,\ y_1 = -2$;

c) $y_{n+2} + 25y_n = 0,\ y_0 = 1,\ y_1 = -1$.

```
(%i1) load(solve_rec)$
```

```
(%i2) y[n+1]+3*y[n]=0; solve_rec(%,y[n],y[0]=5)  /* 2 a) */;
```

$$(\%o2)\quad y_{n+1} + 3\,y_n = 0$$

$$(\%o3)\quad y_n = 5\,(-3)^n$$

```
(%i4) y[n+2]-36*y[n]=0; solve_rec(%,y[n],y[0]=1,y[1]=-2)  /* 2 b) */;
```

$$(\%o4)\quad y_{n+2} - 36\,y_n = 0$$

$$(\%o5)\quad y_n = \frac{6^n}{3} + \frac{2\,(-6)^n}{3}$$

```
(%i6) y[n+2]+25*y[n]=0; solve_rec(%,y[n],y[0]=1,y[1]=-1) /* 2 c) */;
      ev(%,rectform,ratexpand);
```

$$(\%o6)\quad y_{n+2} + 25\,y_n = 0$$

$$(\%o7)\quad y_n = \frac{(\%i+5)(-1)^{n/2}\, 5^{n-1}}{2} - \frac{(\%i-5)(-\%i)^n\, 5^{n-1}}{2}$$

$$(\%o8)\quad y_n = 5^n \cos\left(\frac{\pi n}{2}\right) - 5^{n-1} \sin\left(\frac{\pi n}{2}\right)$$

W powyższym przypadku korzystaliśmy z komend upraszczających.

Przykład 3

Jaką kwotę powinniśmy otrzymać po 5 latach, jeżeli kapitał w wysokości tysiąca złotych ulokowaliśmy na procent składany $i = 2\%$ rocznie?

Oznaczmy przez y_n kwotę uzyskaną z procentu składanego po n latach.
Aby rozwiązać zadanie, skorzystamy z jednorodnego równania różnicowego postaci $y_{n+1} - (1+i)y_n = 0$ z warunkiem początkowym $y_0 = 1\,000$.
Następnie obliczymy wartość $y(5)$.

```
(%i1) load(solve_rec)$
(%i2) y[n+1]-(1+2/100)*y[n]=0;
      solve_rec(%,y[n],y[0]=1000);
```

(%o2) $y_{n+1} - \frac{51\,y_n}{50} = 0$

(%o3) $y_n = 20\,50^{1-n}\,51^n$

```
(%i4) y5:%,n=5;
      fpprintprec:6$ y5,numer;
```

(%o4) $y_5 = \frac{345025251}{312500}$

(%o6) $y_5 = 1104.08$

Po 5 latach otrzymamy 1 104.08 zł.

Uwaga 1

Przy wprowadzaniu równań różnicowych będziemy stosować ułamki zwykłe, gdyż Maxima wykonuje obliczenia symbolicznie, a taki zapis pozwala działać na liczbach całkowitych. Jeśli wpiszemy ułamki dziesiętne, zostaną zamienione na zwykłe i pojawi się stosowny komunikat.
Prezentowana wyżej metoda jest alternatywna w stosunku do metody przedstawionej w rozdziale 11 (przykład 6).

Przykład 4

Znajdziemy rozwiązania ogólne równań różnicowych niejednorodnych:

a) $y_{n+1} - y_n = 3n^2 - n + 2$, b) $y_{n+2} + y_{n+1} - 6y_n = 30(-3)^n$,
c) $y_{n+2} + 9y_n = 50n + 10$, d) $y_{n+2} + y_n = \cos n\pi + i \sin n\pi$,
e) $y_{n+3} - y_{n+2} - 8y_{n+1} + 12y_n = 40 \cdot 2^n$,
f) $y_{n+4} - 13y_{n+2} + 36y_n = 84 \cdot 4^n + 90 \cdot 3^n$.

```
(%i1) load(solve_rec)$
(%i2) y[n+1]-y[n]=3*n^2-n+2; solve_rec(%,y[n]) /* 4 a)*/;
```

(%o2) $y_{n+1} - y_n = 3\,n^2 - n + 2$

(%o3) $y_n = n\,(n^2 - 2\,n + 3) + \%k_1$

```
(%i4) y[n+2]+y[n+1]-6*y[n]=30*(-3)^n; solve_rec(%,y[n]) /* 4 b)*/;
```

(%o4) $y_{n+2} + y_{n+1} - 6\,y_n = -10\,(-3)^{n+1}$

(%o5) $y_n = \%k_2\,2^n + 2\,n\,(-3)^n + \%k_1\,(-3)^n$

Podobnie jak w przykładzie 1 c), również w przykładzie 4 c) wynik poddamy przekształceniom.

```
(%i6) y[n+2]+9*y[n]=50*n+10; solve_rec(%,y[n])   /* 4 c)*/;
      ev(%,rectform)$ facsum(%,sin(n*%pi/2),cos(n*%pi/2));
```

(%o6) $y_{n+2}+9\,y_n=50\,n+10$

(%o7) $y_n=\%k_2\,(-1)^{n/2}\,3^n+(-\%i)^n\,\%k_1\,3^n+5\,n$

(%o9) $y_n=\%i\,(\%k_2-\%k_1)\,3^n\sin\left(\frac{\pi\,n}{2}\right)+(\%k_2+\%k_1)\,3^n\cos\left(\frac{\pi\,n}{2}\right)+5\,n$

Przyjmując w powyższym, że $C_1 = \%i\ (\%k_2 - \%k_1)$ oraz $C_2 = \%k_2 + \%k_1$, ostatecznie otrzymujemy: $y_n = C_1\ 3^n \sin(\frac{\pi\, n}{2}) + C_2\ 3^n \cos(\frac{\pi\, n}{2}) + 5n$.

Ponieważ funkcja **rec_solve** rozwiązuje niejednorodne równania różnicowe jedynie wtedy, gdy zespolony czynnik wymuszający jest zapisany w postaci wykładniczej, najpierw przekształcimy prawą stronę równania.

```
(%i10) y[n+2]+y[n]=cos(n*%pi)+%i*sin(n*%pi);
       ev(%,exponentialize,expand);
       solve_rec(%,y[n]);
```

(%o10) $y_{n+2}+y_n=\%i\sin(\pi\,n)+\cos(\pi\,n)$

(%o11) $y_{n+2}+y_n=\%e^{\%i\,\pi\,n}$

(%o12) $y_n=\dfrac{\%e^{\%i\,\pi\,n}}{\%e^{2\,\%i\,\pi}+1}+\%k_2\,(-1)^{n/2}+(-\%i)^n\,\%k_1$

```
(%i13) ev(%,rectform,ratsimp);
```

(%o13) $y_n=$

$$\frac{\%i\sin(\pi\,n)+\cos(\pi\,n)+(2\,\%i\,\%k_2-2\,\%i\,\%k_1)\sin\left(\frac{\pi\,n}{2}\right)+(2\,\%k_2+2\,\%k_1)\cos\left(\frac{\pi\,n}{2}\right)}{2}$$

```
(%i14) y[n+3]-y[n+2]-8*y[n+1]+12*y[n]=40*2^n; solve_rec(%,y[n]) /* 4 e) */;
```

(%o14) $y_{n+3}-y_{n+2}-8\,y_{n+1}+12\,y_n=5\,2^{n+3}$

(%o15) $y_n=n^2\,2^n+(\%k_3\,n+\%k_2)\,2^n+\%k_1\,(-3)^n$

```
(%i16) y[n+4]-13*y[n+2]+36*y[n]=84*4^n+90*3^n  /* 4 f) */;
       solve_rec(%,y[n]);
```

(%o16) $y_{n+4}-13\,y_{n+2}+36\,y_n=10\,3^{n+2}+21\,4^{n+1}$

(%o17) $y_n=4^n+n\,3^n+\%k_4\,3^n+\%k_2\,2^n+\%k_1\,(-2)^n+\%k_3\,(-3)^n$

Przykład 5

Znajdziemy rozwiązania szczególne równań różnicowych niejednorodnych spełniających podane warunki brzegowe:

a) $y_{n+1} + 6y_n = 10 \cdot 4^n$, $y_0 = 4$; b) $y_{n+2} - 4y_n = 27n^2 - 3$, $y_0 = 1$, $y_1 = 2$;

c) $y_{n+2} + 2y_{n+1} + 2y_n = 25n$, $y_0 = 1$, $y_1 = -1$;

d) $y_{n+2} + 4y_n = 3\left(\cos\frac{n\pi}{2} + i\sin\frac{n\pi}{2}\right)$, $y_0 = 1$, $y_1 = 2$;

e) $y_{n+3} - 2y_{n+2} - 16y_{n+1} + 32y_n = 96\,(-4)^n$, $y_0 = -1$, $y_1 = 3$, $y_2 = 1$.

```
(%i1) load(solve_rec)$
```

```
(%i2) y[n+1]+6*y[n]=10*4^n; solve_rec(%,y[n],y[0]=4) /* 5 a) */;
```

(%o2) $y_{n+1} + 6\,y_n = 10\,4^n$

(%o3) $y_n = 4^n + 3\,(-6)^n$

```
(%i4) y[n+2]-4*y[n]=27*n^2-3; solve_rec(%,y[n],y[0]=1,y[1]=2) /* 5 b) */;
```

(%o4) $y_{n+2} - 4\,y_n = 27\,n^2 - 3$

(%o5) $y_n = 41\,2^{n-1} + (-2)^{n-1} - 9\,n^2 - 12\,n - 19$

```
(%i6) y[n+2]+2*y[n+1]+2*y[n]=25*n /* 5 c) */;
      solve_rec(%,y[n],y[0]=1,y[1]=-1);
      ev(%,rectform, ratsimp);
```

(%o6) $y_{n+2} + 2\,y_{n+1} + 2\,y_n = 25\,n$

(%o7) $y_n = 5\,n + \dfrac{(-\%i-1)^n\,(3\,\%i+5)}{2} - \dfrac{(\%i-1)^n\,(3\,\%i-5)}{2} - 4$

(%o8) $y_n = 3\,2^{n/2}\sin\left(\dfrac{3\,\pi\,n}{4}\right) + 5\,2^{n/2}\cos\left(\dfrac{3\,\pi\,n}{4}\right) + 5\,n - 4$

```
(%i9) y[n+2]+4*y[n]=3*(cos(n*%pi/2)+%i*sin(n*%pi/2))  /* 5 d) */;
      ev(%,exponentialize,expand);
      solve_rec(%,y[n],y[0]=1,y[1]=2);
      ev(%,rectform, ratexpand);
```

(%o9) $y_{n+2} + 4\,y_n = 3\left(\%i\sin\left(\dfrac{\pi\,n}{2}\right) + \cos\left(\dfrac{\pi\,n}{2}\right)\right)$

(%o10) $y_{n+2} + 4\,y_n = 3\,\%e^{\frac{\%i\,\pi\,n}{2}}$

(%o11) $y_n = \dfrac{3\,\%e^{\frac{\%i\,\pi\,n}{2}}}{\%e^{\%i\,\pi} + 4} - (2\,\%i+1)(-1)^{n/2}\,2^{n-2} + (-\%i)^n\,(2\,\%i+1)\,2^{n-2}$

(%o12) $y_n = 2^n\sin\left(\dfrac{\pi\,n}{2}\right) - \%i\,2^{n-1}\sin\left(\dfrac{\pi\,n}{2}\right) + \%i\sin\left(\dfrac{\pi\,n}{2}\right) + \cos\left(\dfrac{\pi\,n}{2}\right)$

```
(%i13) y[n+3]-2*y[n+2]-16*y[n+1]+32*y[n]=96*(-4)^n /* 5 e) */;
       solve_rec(%,y[n],y[0]=-1,y[1]=3,y[2]=1);
```

(%o13) $y_{n+3}-2\,y_{n+2}-16\,y_{n+1}+32\,y_n=6\,(-4)^{n+2}$

(%o14) $y_n=-\dfrac{n\,(-4)^n}{2}+27\,4^{n-2}-11\,2^{n-2}+(-4)^{n-2}$

Przykład 6

Jaka musi być roczna stała spłata z dołu kredytu w wysokości 100 000 *zł* przy rocznej stopie procentowej $i = 7\%$, by został on spłacony w ciągu 5 lat?

Oznaczmy przez y_n pozostałą do spłacenia część kredytu po zapłaceniu n-tej raty. Schemat spłaty kredytu może być opisany niejednorodnym równaniem różnicowym

$$y_{n+1}-(1+i)y_n=-R$$

z warunkiem początkowym $y_0 = 100\,000$, gdzie R jest nieznaną wysokością rocznej raty (czyli rocznej spłaty).

```
(%i1) load(solve_rec)$
(%i2) y[n+1]-(1+7/100)*y[n]=-R;
      solve_rec(%,y[n],y[0]=100000);
```

(%o2) $y_{n+1}-\dfrac{107\,y_n}{100}=-R$

(%o3) $y_n=-\dfrac{100^{1-n}\,107^n\,R}{7}+\dfrac{100\,R}{7}+10\,100^{2-n}\,107^n$

Mając rozwiązanie tego równania zdefiniujemy funkcję y zmiennej n, a następnie wyznaczymy R z równania $y(5)=0$, gdyż po pięciu latach kredyt ma być spłacony.

```
(%i4) y(n):=''(rhs(%))$
      R:solve(y(5)=0,R);
      fpprec:5$ bfloat(R);
```

(%o5) $[R=\dfrac{14025517307000}{575073901}]$

(%o7) $[R=2.4389b4]$

Szukana roczna spłata R jest więc w przybliżeniu równa 24 389 *zł*.

Przykład 7

Kredyt wysokości 20 000 zł należy spłacić w 12 ratach po 2 010 zł płaconych na koniec każdego miesiąca. Ile wynosi miesięcznie oprocentowanie kredytu?

Oznaczmy przez y_n wartość kredytu po n latach.
Schemat spłaty kredytu opisuje równanie różnicowe niejednorodne postaci

$$y_{n+1}-(1+i)y_n=-2\,010$$

z warunkiem początkowym $y_0 = 20\,000$, gdzie i – nieznana stopa procentowa.

```
(%i1) load(solve_rec)$
```

```
(%i2) y[n+1]-(1+i)*y[n]=-2010;
      solve_rec(%,y[n],y[0]=20000);
```

(%o2) $y_{n+1}-(i+1)\,y_n=-2010$

(%o3) $y_n=-\frac{2010\,(i+1)^n}{i}+20000\,(i+1)^n+\frac{2010}{i}$

Podobnie jak w poprzednim przykładzie, zdefiniujemy funkcję y zmiennej n opisującą wartość kredytu po n latach. Następnie wyznaczymy i z równania $y(12)=0$, gdyż po dwunastu miesiącach kredyt ma być spłacony.

```
(%i4) y(n):=''(rhs(%))$
```

```
(%i5) rozw:realroots(y(12)=0);
```

(%o5) $[i=-\frac{59837589}{33554432},i=\frac{1008755}{33554432}]$

Otrzymaliśmy dwa pierwiastki, z których tylko drugi jest liczbą dodatnią, a jego przybliżona wartość wynosi $3\cdot 10^{-2}$.

```
(%i6) fpprec:2$ bfloat(rozw[2]);
```

(%o7) $i=3.0b-2$

Szukana miesięczna stopa procentowa wynosi zatem około 3%.

Przykład 8

Kredyt w wysokości 40 000 *zł* został zaciągnięty przy rocznej stopie procentowej $i=10\%$. Po ilu latach zostanie on spłacony, jeżeli początkowa roczna spłata kredytu wynosi 7 000 *zł* i maleje z każdym rokiem o 50 zł ?

Oznaczmy przez y_n wartość kredytu po n latach.
Schemat spłaty kredytu może być przedstawiony za pomocą równania różnicowego niejednorodnego postaci

$$y_{n+1}-(1+i)\,y_n=50n-7\,000$$

z warunkiem $y_0=40\,000$.

```
(%i1) load(solve_rec)$
```

Znajdujemy rozwiązanie równania różnicowego z warunkiem początkowym, a następnie definiujemy otrzymaną funkcję $y=y(n)$ opisującą wartość kredytu po n latach.

```
(%i2) y[n+1]-(1+1/10)*y[n]=50*n-7000;
      solve_rec(%,y[n],y[0]=40000)$
      y(n):=''(rhs(%));
```

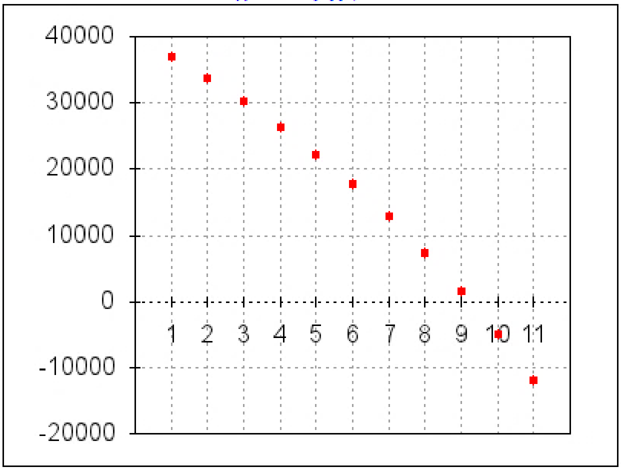
(%o2) $y_{n+1}-\frac{11\,y_n}{10}=50\,n-7000$

(%o4) $y(n):=4\,10^{4-n}\,11^n-65\,10^{3-n}\,11^n-500\,(n-130)$

Funkcja y opisuje wartość kredytu po n latach. Wysokości kwot pozostałych do spłacenia zilustrujemy graficznie.

```
(%i5) s:makelist(y(n),n,1,11)$
      load(draw)$
      wxdraw2d(dimensions=[400,300],grid=true,
               xrange=[0,12],yrange=[-20000,40000],
               xaxis=true,yaxis=true,
               xtics_axis=true,xtics=[1,1,11],
               ytics_axis=true,ytics=[-2.e4,1.e4,1.e5],
               point_size=1,point_type=filled_circle,
               color=red,points(s))$
```

(%t7)




```
(%i8) fpprec:4$
      'y(9)=bfloat(y(9));
      'y(10)=bfloat(y(10));
(%o9) y(9)=1.551b3
(%o10) y(10)=-4.844b3
```

Łatwo zauważyć, że ciąg (y_n) jest malejący dla $n = 1, 2, 3, \ldots, 11$.
Ponadto zachodzą nierówności:
$y_1 > y_2 > \cdots > y_8 > y_9$ i $y_{10} < 0$, więc kredyt zostanie spłacony po 10 latach.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Znaleźć rozwiązanie ogólne równania różnicowego jednorodnego:

a) $4y_{n+1} + 5y_n = 0$; b) $y_{n+2} - 9y_n = 0$; c) $y_{n+2} + 10y_{n+1} + 25y_n = 0$;
d) $y_{n+2} - 2y_{n+1} + y_n = 0$; e) $y_{n+2} + 2y_{n+1} + 2y_n = 0$;
f) $y_{n+3} - 3y_{n+2} - 10y_{n+1} + 24y_n = 0$; g) $y_{n+4} + 8y_{n+2} + 16y_n = 0$.

Zadanie 2

Znaleźć rozwiązanie szczególne równania różnicowego jednorodnego spełniające podane warunki brzegowe:

a) $2y_{n+1} + 5y_n = 0,\ y_0 = 3$; b) $y_{n+2} - y_{n+1} - 6y_n = 0,\ y_0 = 5,\ y_1 = -5$;

c) $y_{n+2} - \sqrt{3}y_{n+1} + y_n = 0,\ y_0 = -1,\ y_1 = 1$.

Zadanie 3

Jaka kwotę powinniśmy otrzymać po 5 latach, jeżeli kapitał w wysokości 30 000 *zł* ulokowaliśmy na procent składany w wysokości $i = 3\%$ rocznie?

Zadanie 4

Znaleźć rozwiązanie ogólne równania różnicowego niejednorodnego:

a) $y_{n+1} - y_n = 6n^2 + 2n$, b) $y_{n+2} - y_{n+1} - 6y_n = 30\,(-2)^n$;

c) $y_{n+2} + 4y_n = 25n + 5$, d) $y_{n+2} + 4y_n = 17\left(\cos\frac{n\pi}{4} + i\sin\frac{n\pi}{4}\right)$;

e) $y_{n+3} + 7y_{n+2} + 16y_{n+1} + 12y_n = 40\,(-2)^n$;

f) $y_{n+4} + 9y_{n+3} + 30y_{n+2} + 44y_{n+1} + 24y_n = 48\,(-2)^n + 6\,(-3)^n$.

Zadanie 5

Znaleźć rozwiązanie szczególne równania różnicowego niejednorodnego spełniające podane warunki brzegowe:

a) $y_{n+1} - 5y_n = 15\ 2^n,\ y_0 = 6$; b) $y_{n+2} - y_n = 6n^2 - 2,\ y_0 = -3,\ y_1 = 6$;

c) $y_{n+2} - 2y_{n+1} + 2y_n = 25n,\ y_0 = -1,\ y_1 = 1$;

d) $y_{n+2} + 4y_n = 6\left(\cos\frac{3n\pi}{2} + i\sin\frac{3n\pi}{2}\right),\ y_0 = 1,\ y_1 = 2$;

e) $y_{n+3} - 4y_{n+2} + 5y_{n+1} - 2y_n = 2\cdot 2^n,\ y_0 = 1,\ y_1 = 2,\ y_2 = -1$.

Zadanie 6

Jaka musi być roczna stała spłata z dołu kredytu w wysokości 30 000 *zł* przy rocznej stopie procentowej $i = 8\%$, by został on spłacony w ciągu 2 lat?

Zadanie 7

Na zakup telewizora został wzięty kredyt w wysokości 4 800 *zł*, który należy spłacić w 24 ratach po 300 *zł* płaconych na koniec każdego miesiąca. Ile wynosi miesięczne oprocentowanie tego kredytu?

Zadanie 8

Kredyt w wysokości 30 000 *zł* został zaciągnięty przy rocznej stopie procentowej $i = 8\%$. Po ilu latach zostanie on spłacony, jeżeli początkowa roczna spłata kredytu wynosi 5 000 *zł* i maleje z każdym rokiem o 50 *zł* ?

Zadanie 9

Kredyt w wysokości 40 000 *zł* został zaciągnięty przy rocznej stopie procentowej $i = 9\%$. Początkowa roczna spłata kredytu wynosi:

a) 7 000 *zł* i maleje z każdym rokiem o 50 *zł*,

b) 8 200 *zł* i maleje z każdym rokiem o 300 *zł*.

Czy kredyt zostanie spłacony wcześniej przy warunkach spłaty a) czy b) ?

Odpowiedzi

1. a) $y_n = \%k_1 \left(-\frac{5}{4}\right)^n$; b) $y_n = \%k_2\, 3^n + \%k_1(-3)^n$; c) $y_n = \%k_2\, n\, (-5)^n + \%k_1\, (-5)^n$;
d) $y_n = \%k_2\, n + \%k_1$; e) $y_n = C_1 2^{\frac{n}{2}} \sin\left(\frac{3\pi n}{4}\right) + C_2\, 2^{\frac{n}{2}} \cos\left(\frac{3\pi n}{4}\right)$,
gdzie $C_1 = i(\%k_2 - \%k_1)$, $C_2 = \%k_2 + \%k_1$, natomiast $\%k_1$, $\%k_2$ są pewnymi stałymi;
f) $y_n = \%k_1 4^n + \%k_2 2^n + \%k_3\, (-3)^n$;
g) $y_n = C_1\, n\, 2^n \sin\left(\frac{\pi n}{2}\right) + C_2 2^n \sin\left(\frac{\pi n}{2}\right) + C_3 n 2^n \cos\left(\frac{\pi n}{2}\right) + C_4 2^n \cos\left(\frac{\pi n}{2}\right)$,
gdzie $C_1 = i\%k_4 - \%k_2$, $C_2 = i\%k_3 - \%k_1$, $C_3 = \%k_4 + \%k_2$, $C_4 = \%k_3 + \%k_1$,
natomiast $\%k_1$, $\%k_2$, $\%k_3$, $\%k_4$ są pewnymi stałymi.
2. a) $y_n = 3\left(\frac{-5}{2}\right)^n$; b) $y_n = 4(-2)^n + 3^n$; c) $y_n = \left(\sqrt{3} + 2\right) \sin\left(\frac{\pi n}{6}\right) - \cos\left(\frac{\pi n}{6}\right)$.
3. 34778 zł.
4. a) $y_n = 2n^3 - 2n^2 + \%k_1$; b) $y_n = \%k_1 3^n + 3n(-2)^n + \%k_2(-2)^n$;
c) $y_n = C_1 2^n \sin\left(\frac{\pi n}{2}\right) + C_2 2^n \cos\left(\frac{\pi n}{2}\right) + 5n - 1$, gdzie $C_1 = i(\%k_2 - \%k_1)$,
$C_2 = \%k_2 + \%k_1$, natomiast $\%k_1$, $\%k_2$ są pewnymi stałymi;
d) $y_n = (4i + 1) \sin\left(\frac{\pi n}{4}\right) + (4 - i) \cos\left(\frac{\pi n}{4}\right) + C_1 2^n \sin\left(\frac{\pi n}{2}\right) + C_2 2^n \cos\left(\frac{\pi n}{2}\right)$,
gdzie $C_1 = i(\%k_2 - \%k_1)$, $C_2 = \%k_2 + \%k_1$, natomiast $\%k_1$, $\%k_2$ są pewnymi stałymi;
e) $y_n = 5n^2(-2)^n + (\%k_3 n + \%k_2)(-2)^n + \%k_1(-3)^n$, gdzie $\%k_1$, $\%k_2$, $\%k_3$
są pewnymi stałymi;
f) $y_n = -n^3(-2)^n + (\%k_4 n^2 + \%k_3 n + \%k_2)(-2)^n + 2n(-3)^n + \%k_1(-3)^n$,
gdzie $\%k_1$, $\%k_2$, $\%k_3$, $\%k_4$ są pewnymi stałymi.
5. a) $y_n = 11 \cdot 5^n - 5 \cdot 2^n$, b) $y_n = -5(-1)^n + n^3 - 3n^2 + n + 2$,
c) $y_n = -23 \cdot 2^{\frac{n}{2}} \sin\left(\frac{\pi n}{4}\right) - 2^{\frac{n}{2}} \cos\left(\frac{\pi n}{4}\right) + 25n$,
d) $y_n = 2i \sin\left(\frac{3\pi n}{2}\right) + 2 \cos\left(\frac{3\pi n}{2}\right) + i2^n \sin\left(\frac{\pi n}{2}\right) + 2^n \sin\left(\frac{\pi n}{2}\right) - 2^n \cos\left(\frac{\pi n}{2}\right)$,
e) $y_n = -8 \cdot 2^n + n2^n + 7n + 9$.
6. 16 823 zł.
7. $i \approx 3.5\%$.
8. Kredyt zostanie spłacony po 9 latach.
9. Kredyt zostanie spłacony wcześniej przy warunkach spłaty b).

20. Pakiet Simplex

Pakiet **simplex** służy do rozwiązywania zagadnień programowania liniowego. Zagadnienia takie dotyczą zazwyczaj optymalnego wykorzystania środków, przy spełnieniu określonych wymagań.

Tabela 20.1. Polecenia pakietu **simplex**

POLECENIA	OPIS
minimize_lp(f(x),[w1,w2…]**)**	minimalizuje funkcję f przy ograniczeniach $w1, w2$ …
maximize_lp(f(x),[w1,w2…]**)**	maksymalizuje funkcję f przy ograniczeniach $w1, w2$ …
nonegative_lp = true	ustawia wartości zmiennych decyzyjnych jako nieujemne
linear_program(A, b, c**)**	rozwiązuje zadanie minimalizacji funkcji kryterium o parametrach c, przy warunkach $Ax = b$ i wektorze zmiennych decyzyjnych $x \geq 0$

Model programowania liniowego można zapisać następująco:

$f(x) = c^T x \to max$ (maksymalizacja funkcji kryterium),

$Ax \leq b$ (warunki ograniczające),

$x \geq 0$ (warunki brzegowe),

gdzie:

$x = [x_1, \dots, x_n]^T$ - wektor n zmiennych decyzyjnych,
A - macierz parametrów warunków ograniczających o wymiarach $m \times n$,
$c = [c_1, \dots, c_n]^T$ - wektor n parametrów funkcji kryterium,
$b = [b_1, \dots, b_m]^T$ - wektor m ograniczeń.

Rozwiązanie przedstawionego powyżej zagadnienia polega na znalezieniu takich wartości zmiennych decyzyjnych x_i, $(i = 1,2, \dots, n)$, dla których wartość funkcji kryterium jest największa, przy spełnieniu warunków ograniczających i brzegowych (czyli warunków wyznaczających zbiór dopuszczalny).
Zadanie programowania liniowego można również sformułować alternatywnie:
$f(x) = c^T x \to min,\ Ax \geq b,\ x \geq 0$.

ZADANIA PRZYKŁADOWE

Przykład 1

Firma krawiecka produkuje dwa rodzaje ubrań ochronnych: typu A i typu B. Ubrania różnią się wytrzymałością w poszczególnych fragmentach stroju.
Do wyprodukowania ubrania typu A potrzeba 1 m materiału wyższej jakości i 0.5 m - niższej. Do wyprodukowania ubrania typu B potrzeba 0.7 m materiału wyższej jakości i 1.1 m - niższej. Firma posiada zapas 50 m materiału wyższej jakości i 40 m - niższej. Ubranie typu A kosztuje 100 zł, typu B - 90 zł. Ile ubrań typu A i typu B powinna produkować firma, aby zmaksymalizować zyski?

Model zadania: x_1 - liczba ubrań typu A, x_2 - liczba ubrań typu B

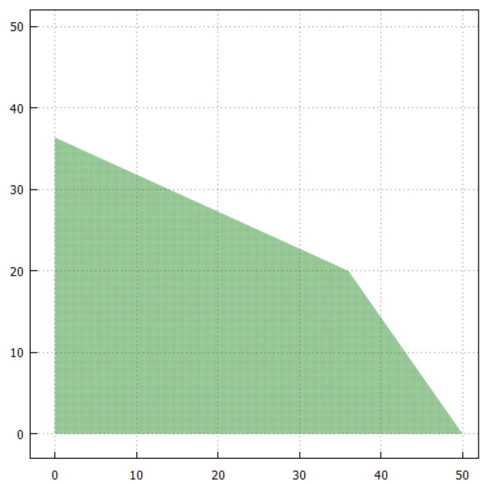
$$f(x_1,x_2) = 100x_1 + 90x_2 \to \max$$
$$x_1 + 0.7x_2 \le 50$$
$$0.5x_1 + 1.1x_2 \le 40$$
$$x_1 \ge 0\,, x_2 \ge 0$$

Rozwiązanie znajdziemy przy pomocy poleceń pakietu simplex.
Na początku ustalimy porządek zmiennych (w tym wypadku jest on domyślny), wczytamy pakiety oraz wprowadzimy ustawienia dla polecenia **draw**.

```
(%i1)    ordergreat (x1,x2)$

(%i3)    load(simplex)$  load(draw)$

(%i4)    set_draw_defaults(terminal=wxt, grid=true, proportional_axes=xy,
                          user_preamble="set style fill transparent solid 0.5 noborder",
                          x_voxel=200, y_voxel=200)$
```

Następnie wczytamy warunki ograniczające i warunki brzegowe oraz narysujemy zbiór rozwiązań dopuszczalnych.

```
(%i5)    w:apply("and", [x1+0.7*x2<=50, 0.5*x1+1.1*x2<=40, x1>=0, x2>=0])$
(%i6)    draw2d(fill_color=forest-green, region(w,x1,-3,52,x2,-3,52))$
```



Zastosujemy teraz polecenie **maximize_lp**.

```
(%i7)  maximize_lp( 100*x1+90*x2,
                  [x1+7/10*x2<=50, 5/10*x1+11/10*x2<=40],
                  nonegative_lp=true);
(%o7)  [5400,[x1=36,x2=20]]
```

Zauważmy, że jeśli chcemy uzyskać wynik dokładny, to parametry liczbowe należy wprowadzić w formie ułamków zwykłych, a nie dziesiętnych.

Rozwiązaniem optymalnym jest wyprodukowanie 36 ubrań typu A i 20 typu B. Zatem, przy spełnieniu warunków ograniczających i brzegowych, wartość funkcji kryterium f jest największa dla $x_1 = 36$ oraz $x_2 = 20$ i wynosi wtedy 5400 zł.

Sprawdzimy również działanie polecenia **linear_program**. Zadanie maksymalizacji musimy teraz sprowadzić do minimalizacji funkcji $-f(x_1, x_2)$.

```
(%i11)  A: matrix([1,7/10], [1/2,11/10])$
        b: [50,40]$
        c: [-100,-90]$
        linear_program(A, b, c);
(%o11)  [[36,20],-5400]
```

Rozważmy jeszcze następujące sytuacje:

Oba ubrania będą kosztowały po 100 zł. Jaka wtedy będzie maksymalna wartość funkcji kryterium?

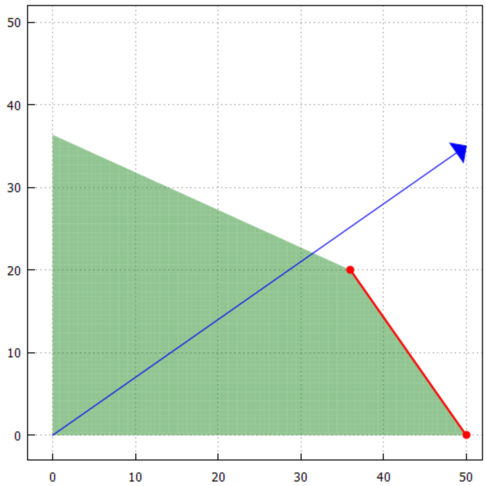
```
(%i12)  maximize_lp(100*x1+100*x2,
                   [x1+7/10*x2<=50,5/10*x1+11/10*x2<=40],
                   nonegative_lp=true);
(%o12)  [5600,[x1=36,x2=20]]
```

Oczywiście, w tym przypadku zysk wzrośnie do 5600 zł.

Natomiast, jeśli cena ubrania typu B spadnie do 70 zł, rozwiązaniem będą wszystkie punkty leżące na odcinku o końcach (36,20) i (50,0) czyli jest nieskończenie wiele rozwiązań optymalnych. Wartość funkcji kryterium w każdym punkcie tego odcinka wynosi 5000 zł.

```
(%i13)  maximize_lp( 100*x1+70*x2,
                   [x1+7/10*x2<=50,5/10*x1+11/10*x2<=40]),
                   nonegative_lp=true;
(%o13)  [5000,[x1=36,x2=20]]
```

```
(%i14)  draw2d(fill_color=forest-green, region(w,x1,-3,52,x2,-3,52),
              vector([0,0],1/2*[100,70]),
              color=red, points_joined=true, line_width=2, point_type=7,
              points([[50,0],[36,20]]))$
```



Wektor o początku w punkcie (50,0) i końcu (36,20) ma współrzędne $[-14{,}20]$ i jest on prostopadły do gradientu funkcji celu [100,70] .
Rzeczywiście,

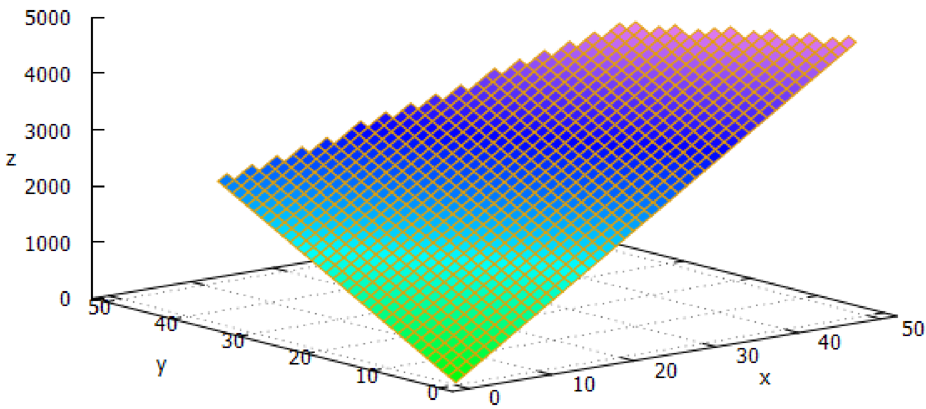
```
(%i15) [36-50,20-0].[100,70];
(%o15) 0
```

iloczyn skalarny tych wektorów wynosi 0.

Powyższy rysunek i obliczenia pokazują, że wspomniany odcinek jest prostopadły do gradientu funkcji celu (w przypadku zagadnień programowania liniowego - do wektora parametrów funkcji celu) oraz zawiera wszystkie punkty wysunięte najdalej w kierunku wskazanym przez gradient. Oznacza to, że w każdym punkcie odcinka wartość funkcji celu jest taka sama i jest to wartość maksymalna.

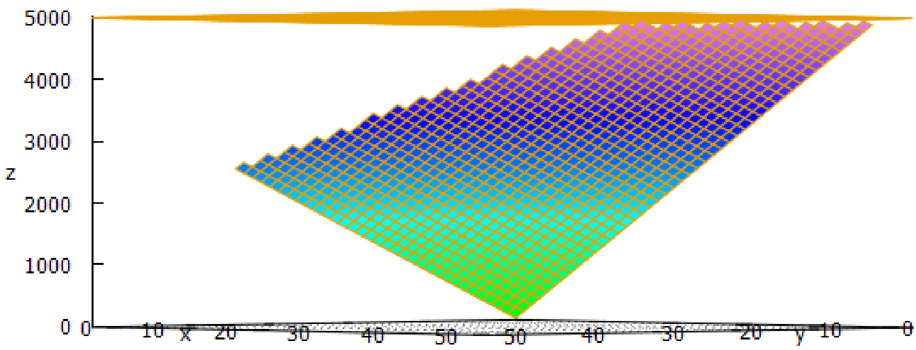
Aby lepiej uwidocznić ten fakt, narysujemy jeszcze część płaszczyzny będącej wykresem funkcji celu określonej na zbiorze dopuszczalnym.

```
(%i15) F(x1,x2):=if x1+0.7*x2<=50 and 0.5*x1+1.1*x2<=40
                 and x1>=0 and x2>=0 then 100*x1+70*x2$

(%i16) plot3d([F(x,y), [x,-2,52], [y,-2,52]], [z,-10,5010], [grid,50,50])$
```



Po zaznaczeniu płaszczyzny poziomej $z = 5000$ widzimy wyraźniej, że wartość maksymalną równą 5000 funkcja celu przyjmuje dla wszystkich punktów odcinka o końcach (36,20), (50,0).

```
(%i17) plot3d([F(x,y), 5000, [x,-2,52], [y,-2,52]], [z,-10,5010], [grid,50,50])$
```



Załóżmy na koniec, że firma krawiecka potrzebuje w pierwszej kolejności opróżnić magazyn, w którym przechowuje materiały do produkcji ubrań oraz że oba materiały potrzebne do produkcji ubrań ochronnych są dostępne w hurtowniach. Wtedy, w naszym modelu, zamiast ograniczeń $x_1 + 0.7x_2 \leq 50$ oraz $0.5x_1 + 1.1x_2 \leq 40$ mielibyśmy $x_1 + 0.7x_2 \geq 50$ oraz $0.5x_1 + 1.1x_2 \geq 40$. Tym samym zbiór rozwiązań dopuszczalnych byłby nieograniczony, a zysk tym większy im większa produkcja.

```
(%i18) maximize_lp( 100*x1+90*x2,
              [x1+7/10*x2>=50,5/10*x1+11/10*x2>=40]),
              nonegative_lp=true;
(%o18) Problem not bounded!
```

Przykład 2

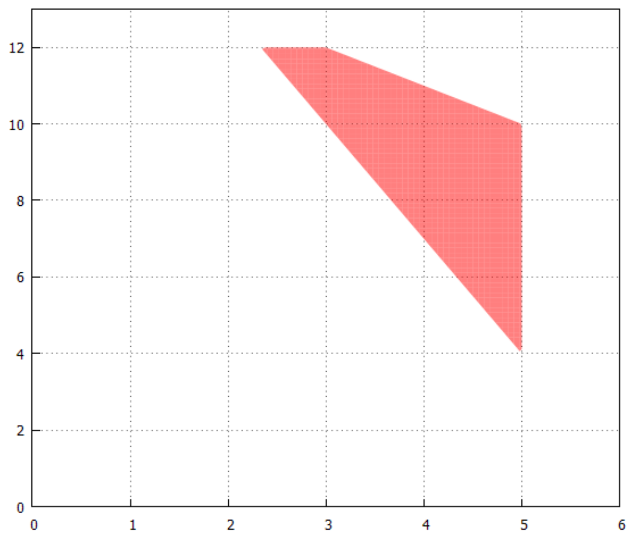
95-osobową grupę turystów należy przetransportować na wschód słońca do stóp wulkanu. Możliwy jest transport minibusami i jeepami. Wynajęcie busa zabierającego 15 osób kosztuje 200 \$, a jeepa zabierającego 5 osób - 70 \$. Ponadto bus wymaga obsługi 2 osób (kierowca i przewodnik), a jeep tylko 1 osoby (kierowca). Ile busów i ile jeepów należy wynająć, aby koszt przewozu był najniższy, jeśli dodatkowo wiadomo, że dyspozycyjnych jest jedynie 5 przewodników, 15 kierowców i wolnych jest 12 jeepów?

Model zadania: x_1 - liczba busów, x_2 - liczba jeepów

$f(x_1, x_2) = 200x_1 + 70x_2 \rightarrow \min$

$15x_1 + 5x_2 \geq 95$
$x_1 \leq 5$
$x_2 \leq 12$
$x_1 + x_2 \leq 15$
$x_1 \geq 0, \quad x_2 \geq 0$

```
(%i2)  load(simplex)$ load(draw)$

(%i3)  set_draw_defaults(terminal=wxt, grid=true, x_voxel=100, y_voxel=100,
                         user_preamble="set style fill transparent solid 0.5 noborder")$

(%i5)  w:apply("and",[15*x1+5*x2>=95, x1+x2<=15, x1<=5, x2<=12,
                      x1>=0, x2>=0])$
       draw2d(region(w,x1,0,6,x2,0,13))$

(%i6)  minimize_lp( 200*x1+70*x2,
                   [15*x1+5*x2>=95, x1+x2<=15, x1<=5, x2<=12]),
                   nonegative_lp=true;
(%o6)  [1280,[x2=4,x1=5]]
```



Rozwiązaniem optymalnym jest wynajęcie 5 busów i 4 jeepów. Koszt przewozu to 1280 zł.

ZADANIA DO SAMODZIELNEGO ROZWIĄZANIA

Zadanie 1

Rozwiązać następujące zadanie programowania liniowego:

$$f(x, y) = 3x_1 + 7x_2 + 5x_3 \to \max$$
$$x_1 + x_2 + x_3 \leq 80$$
$$2x_1 + 3x_2 + x_3 \leq 100$$
$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$$

Zadanie 2

Królewna Fiona planuje przejść na tygodniową dietę składającą się jedynie z bananów i bułek. Jednocześnie chce dostarczać organizmowi minimalną, potrzebną do normalnego funkcjonowania dawkę białek (30 g) i węglowodanów (70 g) nie przekraczając 60 g tłuszczów dziennie. Zamierza jeść przynajmniej 5 razy dziennie. Ile bananów i ile bułek dziennie powinna zjadać, by dostarczyć jak najmniej kalorii?

1 banan - 162 kcal, 1.96 g białka, 36.43 g węglowodanów, 0.59 g tłuszczu.

1 bułka - 260 kcal, 7.79 g białka, 52.91 g węglowodanów, 1.61 g tłuszczu.

Odpowiedzi

1. $x_1 = 0,\ x_2 = 10,\ x_3 = 70$, wartość funkcji kryterium: 420
2. ok. 1.5 bułki + 3.5 banana

21. Elementy programowania

Tabela 21.1

PODSTAWOWE INSTRUKCJE I ICH SKŁADNIA
if warunek **then** akcja **if** warunek **then** akcja1 **else** akcja2 **if** warunek **then** akcja1 **elseif** warunek2 **then** akcja2 **elseif** … **else** akcja0 - instrukcje warunkowe (wykonują akcję, o ile spełniony jest warunek)
for zmienna: wart_poczatkowa **thru** wart_koncowa **step** krok **do** instrukcje **for** zmienna: wart_poczatkowa **while** warunek_pozostania **do** instrukcje **for** zmienna: wart_poczatkowa **unless** warunek_wyjscia **do** instrukcje - instrukcje pętli
block([zm_lokalne],instrukcje, return(wart_koncowa)**)** - zwraca wartość zmiennej wewnątrz polecenia **return**, wartości zmiennych lokalnych nie są dostępne poza poleceniem **block** **block(**[zm_lokalne], wyr_1, …. , wyr_n**)** - podobnie jak wyżej, ale zwraca wyr_n **%% -** odniesienie do poprzedniego wyrażenia wewnątrz polecenia **block**
lambda - anonimowa funkcja (użyta razem z poleceniem **map** działa podobnie do konstrukcji foreach znanej z innych języków programowania)

Przykład 1

Podamy trzy proste przykłady zastosowania instrukcji **for**.

```
(%i1) for i:1 while (i<=4) do display(i^2)$
```

$1^2 = 1$

$2^2 = 4$

$3^2 = 9$

$4^2 = 16$

```
(%i2) for i:1 thru 10 step 3 do display(2^i)$
```

$2^1 = 2$

$2^4 = 16$

$2^7 = 128$

$2^{10} = 1024$

```
(%i3) for k:1 thru 6 do (fpprec:k, print(bfloat(%e)))$
3.0b0
2.7b0
2.72b0
2.718b0
2.7183b0
2.71828b0
```

Przykład 2

Zdefiniujemy dwoma sposobami funkcję obliczającą silnię liczby naturalnej, wykorzystując polecenia z tabeli 21.1.

```
(%i1) silnia1(n):=block([k], k:1, for i:1 thru n do k:k*i, return(k))$
```

```
(%i2) silnia1(7);
(%o2) 5040
```

```
(%i3) silnia2(n):=if n = 0 then 1 else n*silnia2(n-1)$
```

```
(%i4) silnia2(7);
(%o4) 5040
```

Przykład 3

Korzystając z instrukcji **block,** zdefiniujemy polecenie (funkcję) obliczające całki metodą „przez części” i zastosujemy je do całek: $\int x\cos x\,dx$, $\int \operatorname{arctg} t\,dt$.

```
(%i1) calka_czesci(f,h,x):=block([g],g:integrate(h,x),
                                  f*g-'integrate(g*diff(f,x),x) =
                                  ev(f*g-'integrate(g*diff(f,x),x),nouns))$
```

```
(%i2) calka_czesci(x,cos(x),x);
```

(%o2) $x\sin(x)-\int \sin(x)\,dx = x\sin(x)+\cos(x)$

```
(%i3) calka_czesci(atan(t),1,t);
```

(%o3) $t\operatorname{atan}(t)-\int \frac{t}{t^2+1}\,dt = t\operatorname{atan}(t)-\frac{\log(t^2+1)}{2}$

Literatura

[1] Maxima Manual, Ver. 5.40.0
http://maxima.sourceforge.net/docs/manual/en/maxima.pdf

[2] „Maxima.", Wikibooks
http://pl.wikibooks.org/w/index.php?title=Maxima&oldid=200514

[3] http://maxima.sourceforge.net/documentation.html

[4] **De Souza P.N., Fateman R.J., Moses J., Yapp C.:** The Maxima Book,
http://maxima.sourceforge.net/docs/maximabook/maximabook-19-Sept-2004.pdf

[5] **Just A., Walas W., Kondratiuk-Janyska A., Pełczewski J., Małolepszy M., Niedziałkowska A.:** Matematyka dla studentów politechnik. Teoria, przykłady, zadania z wykorzystaniem pakietów matematycznych, Łódź 2013.

[6] **Leydold J., Petry M.:** Introduction to Maxima for Economics, Institute for Statistics and Mathematics, WU Wien, 2011.
http://statmath.wu.ac.at/~leydold/maxima/MaximaSkript.pdf

[7] **Riotorto M. R.:** A Maxima-Gnuplot Interface
http://riotorto.users.sourceforge.net/gnuplot/

[8] **Woollett Edwin L.:** Maxima by Example
www.csulb.edu/~woollett/

POLITECHNIKA ŁÓDZKA
CENTRUM NAUCZANIA MATEMATYKI I FIZYKI

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.

W skrypcie zostały przedstawione rozwiązania zadań przykładowych z wykorzystaniem Maximy 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.

ISBN 978-83-7283-892-6