

Damian SŁOTA

Katedra Metod Matematycznych w Technice i Informatyce, Wydział Matematyki Stosowanej,
Politechnika Śląska, ul. Kaszubska 23, 44-100 Gliwice

Mathematica – podstawowe instrukcje

Streszczenie. W prezentowanych materiałach dydaktycznych przedstawiono podstawowe instrukcje programu *Mathematica*. W szczególności opisano, jak zacząć pracę w środowisku *Mathematica*. Przedstawiono wybrane instrukcje dotyczące wektorów oraz macierzy, równań i nierówności. Opisano podstawowe instrukcje z zakresu analizy matematycznej oraz równań różniczkowych.

Słowa kluczowe: Mathematica, CAS, obliczenia symboliczne, obliczenia numeryczne.

Wstęp

Mathematica jest programem firmy Wolfram Research, Inc. Jego twórcą jest Stephan Wolfram. Wersja 1.0 programu ukazała się 23 czerwca 1988 r. Zawierała ona 554 instrukcje. W kolejnych wersjach programu dodawano nowe instrukcje oraz ulepszano istniejące. Wersja 14.0 posiada już 6602 instrukcje [4]. Obecnie dystrybuowana jest wersja 14.1. Wyłącznym dystrybutorem programów firmy Wolfram Research, Inc. na terenie Polski jest firma Gambit (mathematica.pl, gambit.net.pl).

Mathematica jest znakomitym środowiskiem zarówno do obliczeń symbolicznych, jak i do opracowywania danych eksperymentalnych, obliczeń przybliżonych czy tworzenia grafiki matematycznej. Łatwość obsługi, intuicyjność oraz bardzo duże możliwości rekomendują program *Mathematica* jako doskonałe środowisko pracy dla studentów i uczniów, jak i pracowników naukowych, stosujących w pracach badawczych zaawansowany aparat matematyczny [4]. W wielu zastosowaniach istotna będzie możliwość bezpośredniego korzystania z bazy wiedzy *Wolfram Knowledgebase* utworzonej przez firmę Wolfram Research. *Mathematica* umożliwi obliczenia w chmurze, za pośrednictwem przeglądarki internetowej, jak również lokalnie na komputerach.

Więcej informacji na temat pakietu *Mathematica* oraz innych produktów firmy Wolfram Research, Inc. można znaleźć na stronie wolfram.com. Można tam znaleźć m.in. przygotowane przez użytkowników projekty (Wolfram Demonstrations Project), a także nowe instrukcje (Wolfram Function Repository).

Pierwsza wersja prezentowanych materiałów dydaktycznych powstała wiele lat temu na potrzeby prowadzonych szkoleń. Z biegiem lat materiały były modyfikowane i poprawiane wraz z rozwojem programu *Mathematica*. Materiały były także wykorzystane na wykładzie z przedmiotu platformy obliczeniowe

prowadzonego dla studentów Wydziału Matematyki Stosowanej Politechniki Śląskiej. Ostatnie szkolenie odbyło się we wrześniu 2024 roku w ramach projektu „F2S Politechnika Śląska uczelnią wspierającą kadrę w drodze do doskonałości”. Materiały zostały opracowane w postaci notatnika (plik *.nb) programu *Mathematica*. Na dalszych stronach niniejszego pliku zawarty jest wydruk do pliku pdf przygotowanego notatnika. Dlatego też styl następnych stron trochę się różni od standardów czasopisma MINUT. Na stronie czasopisma oprócz pliku pdf dostępny jest także plik notatnika (nb). Notatnik może być rozpowszechniany zgodnie z licencją CC BY-NC-SA 4.0. Obliczenia wykonano w wersji 14.1 pakietu. Notatnik zawiera podstawowe instrukcje, które będą działać także we wcześniejszych wersjach pakietu. Przetestowano go także w wersjach 14.0, 13.2, 12.3, 12.0 oraz 11.3. Różnice pojawiają się jedynie w wyglądzie graficznym kilku komórek wyników (Output).

Literatura

1. H. Gliński, R. Grzymkowski, A. Kapusta, D. Słota, *Mathematica 8*, WPKJS, Gliwice 2012.
2. R. Grzymkowski, A. Kapusta, T. Kuboszek, D. Słota, *Mathematica 6*, WPKJS, Gliwice 2008.
3. J.A. Mischczak, *Short Lessons on Wolfram Language*, Quoqa Pubs, Knurów 2024.
4. Strona: *mathematica.pl*, dostęp 09.10.2024.
5. S. Wolfram, *The Mathematica Book*, 5th ed., Wolfram Media/Cambridge University Press, Champaign/Cambridge 2003.
6. S. Wolfram, *An Elementary Introduction to the Wolfram Language*, Wolfram Media, Champaign 2017.

1. Podstawowe informacje

1.1. Uruchamianie programu

Przy pierwszym uruchomieniu programu *Mathematica* zobaczymy następujące elementy: belkę narzędziową oraz okno powitania, w którym możemy wybrać, czy chcemy otworzyć nowy dokument (zwany notatnikiem: Notebook), istniejący już notatnik lub przeglądnąć pomoc pakietu.

Polecenie dla systemu wpisujemy z klawiatury, na koniec naciskając jednocześnie klawisze `SHIFT` i `ENTER` lub klawisz `ENTER` z bocznej klawiatury numerycznej.

Po wpisaniu polecenia `2+3` i naciśnięciu klawiszy `SHIFT+ENTER` nasze polecenie zostanie wykonane:

```
In[1]:=
      2 + 3
Out[1]=
      5
```

Oznaczenia *In[numer]* oraz *Out[numer]* wprowadza *Mathematica*. Aby ułatwić nam pracę, numeruje ona nasze polecenia. Wynik *n*-tego polecenia możemy następnie wykorzystać przez podanie jego numeru poprzedzonego znakiem `%`.

Pakiet *Mathematica* składa się z dwóch programów: *Front end* i *Kernel*. Pierwszy z nich to program (interfejs graficzny) służący do komunikacji między użytkownikiem a jądrem pakietu (*Kernel*), które wykonuje wszystkie obliczenia.

1.2. Zachowanie i otwieranie dokumentu

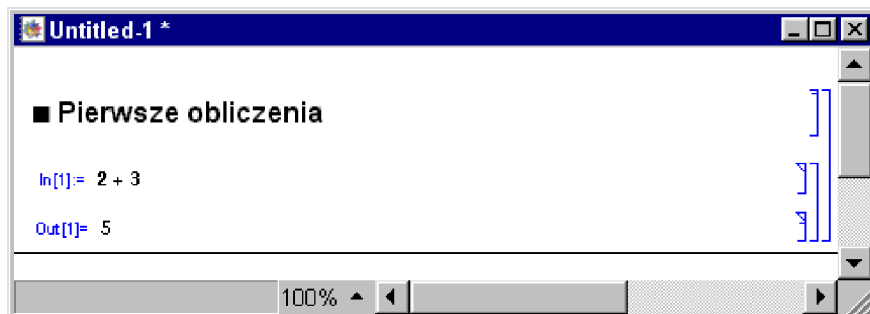
Jeśli chcemy zachować wyniki naszej pracy, przed zamknięciem programu wybieramy z menu *File* opcję *Save as* lub *Save*. Ukaże się wtedy okienko dialogowe (takie samo jak w innych programach windowsowych), w którym możemy podać własną nazwę dla pliku (ewentualnie wraz ze ścieżką dostępu), w którym chcemy przechować wyniki naszej pracy. Po ponownym uruchomieniu programu zapisaną sesję wczytujemy wybierając z menu *File* opcję *Open* lub wybierając nazwę notatnika z listy ostatnio otwieranych dokumentów dostępnej w menu *File/Open Recent*. Po wybraniu opcji *Open* w otwartym okienku dialogowym musimy podać nazwę (i ewentualnie ścieżkę dostępu) pliku, który chcemy otworzyć.

1.3. Budowa notatnika

Notatnik (dokument programu *Mathematica*) składa się z komórek. Komórki mogą zawierać: tekst, grafikę, formy wejściowe (*Input*) i wyjściowe (*Output*) oraz inne obiekty. Komórki możemy łączyć w grupy, które mogą zawierać powiązany z sobą materiał np. tworzący rozdziały, podrozdziały, części obliczeń itp.

Komórki oznaczane są separatorem - klamrą znajdującą się z prawej strony ekranu.

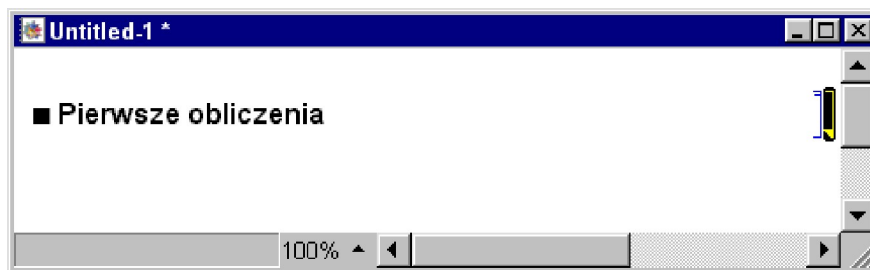
Na poniższym rysunku widzimy dokument z trzema komórkami: komórką zawierającą nazwę, komórką wejściową i komórką wyjściową. Z prawej strony umieszczonych jest kilka separatorów komórek.



Wskazując kursorem myszy separator komórki lub grupy i jednokrotnie naciskając lewy jej klawisz, spowodujemy zaznaczenie danej komórki lub grupy komórek. Zaznaczenie komórki jest uwidaczniane zmianą wyglądu separatora komórki (niebieski prostokąt). Stworzenie grupy komórek spowoduje pojawienie się na ekranie nowego separatora obejmującego wszystkie komórki wchodzące w skład tej grupy. Na naszym „ekranie” widzimy trzy komórki, z których druga i trzecia są połączone w grupę, oraz jest utworzona kolejna grupa zawierająca pierwszą komórkę i poprzednią grupę. Pierwsza grupa jest wynikiem standardowego działania programu. Łączy on zawsze w jedną grupę komórkę wejściową i odpowiadającą jej komórkę wyjściową. Tworząc grupę komórek, możemy na początku napisać nazwę i komórkę z nią przekształcić na komórkę typu *Title*, *Section*, itd. (aby to wykonać, korzystamy z podmenu *Style* w menu *Format* lub z rozwijalnej listy umieszczonej na belce narzędziowej, którą włączamy opcją *Toolbar/Formatting* z menu *Window*). Określenie typu komórki powoduje, że zawarty w niej tekst zostanie wypisany w odpowiedniej postaci (krój i rozmiar czcionki, oznaczniki). Na poprzednim rysunku widzieliśmy komórkę typu *Section* (pierwsza komórka). Kwadraty z lewej strony komórek to wspomniane wcześniej oznaczniki wprowadzane przez program.

Aby zaznaczyć większą liczbę komórek, za pomocą myszy zaznaczamy jedną komórkę, następnie przytrzymując jej lewy klawisz, przechodzimy kursorem do następnego separatora.

Jeśli chcemy utworzyć grupę komórek, to zaznaczamy je wszystkie, a następnie łączymy je za pomocą polecenia *Group Cells/Group Together* (dostępnego po wybraniu z menu *Cell/Grouping*). Aby dokument zajmował mniej miejsca na ekranie, możemy poszczególne grupy zamykać. Po zamknięciu, z całej grupy komórek na ekranie będzie widoczna tylko komórka pierwsza.



W celu zamknięcia lub otwarcia grupy musimy kursor myszy ustawić nad separatorem grupy (kursor przekształci się wtedy w strzałkę) i dwukrotnie nacisnąć jej lewy klawisz.

1.4. Wykorzystanie palet

Palety umożliwiają szybkie umieszczanie we własnych notatnikach instrukcji, symboli operacji matematycznych, stałych matematycznych, liter alfabetu greckiego, itp. Przy pierwszym uruchomieniu programu automatycznie zostaje otwarta jedna paleta *Basic Math Input*.

Na początku każdej następnej sesji są otwierane te palety, które były otwarte podczas zamykania poprzedniej sesji.

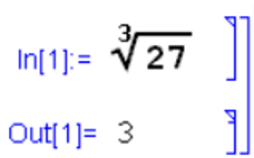
W celu wstawienia instrukcji z palety do notatnika, najpierw umieszczamy kursor w miejscu, gdzie ma się pojawić nowa instrukcja lub symbol, klikamy w tym miejscu (pojawi się kursor tekstowy w postaci pionowej kreski), a następnie klikamy odpowiedni przycisk na paletce.



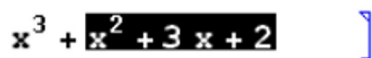
W miejsce kursora tekstowego zostanie wstawiona wybrana instrukcja lub symbol, a sam kursor zostanie przesunięty na koniec (jeśli wstawialiśmy symbol) lub ustawiony w miejscu przeznaczonym na uzupełnianie instrukcji (czarny kwadrat).



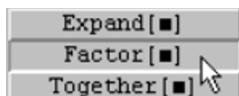
Instrukcje zawarte w paletach posiadają jedno lub więcej miejsc, które należy przed ich wykonaniem uzupełnić. Miejsca te są oznaczone kwadratami (■, □), jeden kwadrat jest czarny, pozostałe (jeśli występują) puste. Czarny kwadrat oznacza miejsce, w którym ustawiony zostanie kursor po wczytaniu instrukcji do notatnika. Po uzupełnieniu instrukcji (między miejscami do wpisywania danych możemy poruszać się, wykorzystując klawisz tabulacji lub mysz) możemy wydać polecenie obliczenia wpisanego wyrażenia.



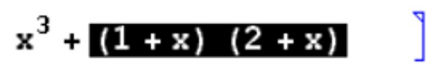
Dzięki paletom mamy możliwość wykonania w jednej linii części operacji (m.in. algebraicznych). Zilustrujmy to na przykładzie. Otwórzmy najpierw paletę z instrukcjami algebraicznymi (z menu *Palettes* wybieramy opcję *Other*, a następnie paletę *Algebraic Manipulation*). Przypuśćmy, że chcemy część wyrażenia zapisać w postaci iloczynowej. W tym celu musimy zaznaczyć interesujący nas fragment wyrażenia.



Następnie w otwartej paletce kliknąć przycisk z instrukcją *Factor*.



Po tej operacji wyrażenie jest obliczane i w miejsce pierwotnego wyrażenia jest podstawiany wynik obliczeń. Nie pojawia się w tym przypadku nowa komórka wyjściowa (*Out[]*).



1.5. Posługiwanie się systemem pomocy

Program *Mathematica* wyposażony jest w dwa rodzaje pomocy. Pierwszy jest wbudowany w jądro programu, natomiast drugi jest dostępny w wersji okienkowej. W pierwszym z tych dwóch rodzajów pomocy informacje o interesujących nas instrukcjach możemy uzyskać w następujący sposób:

? instrukcja	informacje o instrukcji;
?? instrukcja	rozszerzone informacje o instrukcji;
? "abc*"	informacje o instrukcjach zaczynających się od ciągu liter abc.

Sprawdźmy informacje dotyczące instrukcji *Plot*.

In[2]:=

?Plot

Out[2]=

Symbol i

`Plot[f, {x, xmin, xmax}]` generates a plot of f as a function of x from x_{min} to x_{max} .

`Plot[{f1, f2, ...}, {x, xmin, xmax}]` plots several functions f_i .

`Plot[{..., w[fi], ...}, ...]` plots f_i with features defined by the symbolic wrapper w .

`Plot[..., {x} ∈ reg]` takes the variable x to be in the geometric region reg .

▼

Zobaczmy jeszcze, jakie instrukcje zaczynają się od ciągu `Abs` (musimy pamiętać, że w programie *Mathematica* rozróżniane są duże i małe litery).

In[3]:=

? "Abs*"

Out[3]=

▼ System`

Abs	AbsoluteCorrelationFunction	AbsolutePointSize
AbsArg	AbsoluteCurrentValue	AbsoluteThickness
AbsArgPlot	AbsoluteDashing	AbsoluteTime
Absolute	AbsoluteFileName	AbsoluteTiming
AbsoluteCorrelation	AbsoluteOptions	

Klikając myszą na dowolną z wypisanych nazw, wyświetlone zostaną informacje dotyczące wybranej instrukcji.

Drugi ze wspomnianych systemów pomocy jest dostępny z menu *Help* lub poprzez naciśnięcie klawisza **F1** (gdy kursor jest między komórkami). Otworzy się wtedy okno, w którym możemy przeglądać dokumentację programu lub wpisać interesującą nas instrukcję. Po wybraniu interesującej nas instrukcji, w dolnej części zostaną wyświetlony odnośniki do stron dotyczących wpisanego polecenia.

Jeśli chcemy uzyskać informacje na temat instrukcji, którą mamy wpisaną w notatniku, możemy zaznaczyć ją i nacisnąć klawisz **F1**, otworzy się wtedy okno pomocy z interesującym nas opisem. To samo okno się otworzy, gdy klikniemy na znaku i w opisie uzyskiwanym z jądra programu.

1.6. Przerwanie obliczeń

Aktualnie wykonywane obliczenia możemy przerwać, wybierając opcję *Abort Evaluation* z menu *Evaluation* lub jednocześnie naciskając klawisze `[ALT]` i kropkę (`[ALT].`).

2. Pierwsze obliczenia

Podstawowe działania i relacje są w programie *Mathematica* zapisywane tak jak w matematyce. A zatem, do zapisania dodawania i odejmowania używamy symboli "+" oraz "-", mnożenia "*" lub odstęp (spacji). Do zapisania dzielenia użyjemy ukośnika "/" albo gotowego szablonu z palety *Basic Math Input* - $\frac{\square}{\square}$. Potęgowanie zapisujemy używając

szablonu \square^{\square} albo symbolu "^". Kolejność wykonywania działań jest też taka, jaką przyjmuje matematyka. Jeżeli priorytety działań nie określają jednoznacznie wyniku, wyrażenie obliczane jest zgodnie z ogólnie przyjętą zasadą od lewej do prawej. Do zmiany kolejności wykonywania działań służą nawiasy (wyłącznie!) okrągłe "(" i ")". Nawiasy te mogą być zagnieżdżone jedne w drugich.

Uwaga. Po zapisaniu wyrażenia przeznaczonego do obliczenia nie piszemy "=", lecz naciskamy klawisze `SHIFT+ENTER` lub `ENTER` na bocznej klawiaturze numerycznej.

A oto przykłady obliczania wyrażeń liczbowych:

In[4]:=

$(2 + 1) * 5$

Out[4]=

15

In[5]:=

$\frac{1 + 3^3}{23}$

Out[5]=

$\frac{28}{23}$

In[6]:=

$16 / 4 * 3 * 5^2$ (* Uwaga: Tu mnożenie zaznaczone jest spacją! *)

Out[6]=

300

W ostatnim przykładzie został użyty komentarz. Jest to dowolny ciąg znaków umieszczony pomiędzy dwuznakami "(" i "*). *Mathematica* go ignoruje, a służy on do umieszczania różnorodnych wyjaśnień.

Możemy odwoływać się do ostatnio uzyskanego wyniku za pomocą operatora %, albo za pomocą %nr do wyniku uzyskanego w komórce wyjściowej o numerze nr (Out[nr]).

Dodamy do ostatniego wyniku (300) wynik piąty ($\frac{28}{23}$) oraz liczbę 4:

In[7]:=

$% + \%5 + 4$

Out[7]=

$\frac{7020}{23}$

In[8]:=

$Out[6] + Out[5] + 4$

Out[8]=

$\frac{7020}{23}$

W wyrażeniach możemy używać funkcji. Ich nazwy piszemy z dużej litery, argumenty umieszczamy w nawiasach kwadratowych:

<code>Sin[x]</code>	oblicza sinus kąta o mierze łukowej x;
<code>Cos[x]</code>	oblicza cosinus kąta o mierze łukowej x;

Tan [x]	oblicza tangens kąta o mierze łukowej x;
Cot [x]	oblicza cotangens kąta o mierze łukowej x;
Log [a, x]	oblicza logarytm o podstawie a z liczby x;
Abs [x]	podaje wartość bezwzględną liczby x;
n!	liczy n!;
N [wyr]	podaje wartość numeryczną wyrażenia wyr; z domyślnie określoną dokładnością;
N [wyr, n]	podaje wartość numeryczną wyrażenia wyr; z dokładnością do n cyfr znaczących.

Policzmy $\cos \frac{\pi}{6}$ oraz $\sin \pi$:

In[9]:=

Cos $\left[\frac{\pi}{6} \right]$

Out[9]=

$\frac{\sqrt{3}}{2}$

In[10]:=

Sin [Pi]

Out[10]=

0

Uwaga. Liczbę π uzyskujemy, używając jej symbolu z palety *Basic Math Input* albo pisząc Pi. W przypadku liczby Eulera (podstawy logarytmu naturalnego) również możemy posłużyć się symbolem z palety (*e*) albo napisać E. Liczby te możemy również wstawić do notatnika naciskając kolejno klawisze `ESC` pi `ESC` dla liczby π oraz `ESC` ee `ESC` dla liczby Eulera *e* (`ESC` oznacza klawisz Escape).

Mathematica "stara się" podawać wartości dokładne.

Przykład podawania dokładnego wyniku:

In[11]:=

Sin [1]

Out[11]=

Sin [1]

Odczytajmy wartość numeryczną $\sin 1$, z domyślnie ustaloną dokładnością:

In[12]:=

N [Sin [1]]

Out[12]=

0.841471

Dokładność wyniku możemy zmieniać:

In[13]:=

N [10 Sin [1], 400]

Out[13]=

8.41470984807896506652502321630298999622563060798371065672751709991910404391239668948639`.
 7435430526958543490379079206742932591189209918988811934103277292124094807919558267666`.
 0699990776401197840878273256634748480287029865615701796245539489357292467012708648628`.
 1053382030561377218203868449667761674266239013382753397956764255565477963989764824328`.
 6902756964291206300583036515230312782552898532648513981935

In[14]:=

Sin [1. ` 100]

Out[14]=

0.84147098480789650665250232163029899962256306079837106567275170999191040439123966894863`.
 97435430526959

Jak widzimy, separatorem części dziesiętnej jest w tym programie kropka, a nie przecinek.

Oprócz liczb możemy używać zmiennych. Bardzo ułatwia to obliczenia. Zmienną powołujemy do istnienia, nazywając ją. Określenie typu zmiennej następuje w momencie przypisania jej wartości.

Powołajmy do istnienia zmienną o nazwie a:

In[15]:=

a

Out[15]=

a

Zapytamy teraz o podstawową informację o tej zmiennej:

In[16]:=

? a

Out[16]=

Symbol
Global`a
Full Name Global`a
^

A teraz utwórzmy zmienną b i przypiszmy jej wartość 5:

In[17]:=

b = 5

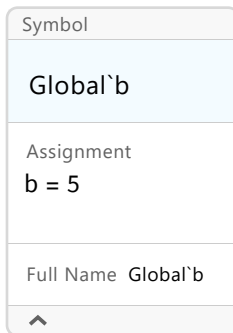
Out[17]=

5

In[18]:=

?b

Out[18]=



In[19]:=

b + 5

Out[19]=

10

Przypisywanie wartości zmiennej i usuwanie przypisania możemy wykonać następująco:

<code>x = wart</code>	przypisanie zmiennej <code>x</code> wartości <code>wart</code> ;
<code>x =. albo Clear[x]</code>	usunięcie przypisania.

Zajmiemy się teraz zapisywaniem relacji. *Mathematica* oprócz wartości liczbowych może posługiwać się między innymi wartościami logicznymi (Boolean): True (prawda) oraz False (fałsz).

<code>x == y</code>	<code>x</code> jest równe <code>y</code> ;
<code>x != y</code> albo <code>x ≠ y</code>	<code>x</code> jest różne od <code>y</code> ;
<code>x > y</code>	<code>x</code> większe od <code>y</code> ;
<code>x < y</code>	<code>x</code> mniejsze od <code>y</code> ;
<code>x >= y</code> albo <code>x ≥ y</code>	<code>x</code> większe bądź równe <code>y</code> ;
<code>x <= y</code> albo <code>x ≤ y</code>	<code>x</code> mniejsze bądź równe <code>y</code> .

Porównajmy dwie wielkości:

In[20]:=

b == 4

Out[20]=

False

Jeśli wartość logiczna wyrażenia nie jest określona, w komórce wyjściowej wyrażenie to zostanie ponownie wypisane:

In[21]:=

a > 3

Out[21]=

a > 3

W powyższym przykładzie użyliśmy zmiennej `a`, której nie przypisano żadnej wartości.

Mathematica może operować również na symbolach. Usuńmy wartość przypisaną zmiennej `b` i policzmy $(a + b)^3$:

In[22]:=

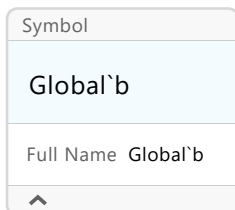
b = .

Sprawdzamy, czy wartość przypisana zmiennej b została usunięta i wykonujemy rachunki:

In[23]:=

?b

Out[23]=



In[24]:=

(a + b)³

Out[24]=

(a + b)³

Użyjemy funkcji `Expand` do rozwinięcia otrzymanego wyniku:

In[25]:=

ww = Expand [%]

Out[25]=

 $a^3 + 3 a^2 b + 3 a b^2 + b^3$

Jeśli mamy jakieś wyrażenie zapisane symbolicznie, możemy łatwo zastąpić którykolwiek z parametrów liczbą albo innym wyrażeniem.

<code>wyr /. a → wart</code>	zastąpienie parametru a w wyrażeniu <code>wyr</code> przez <code>wart</code> ;
<code>ReplaceAll[wyr, {a → wart}]</code>	zastąpienie parametru a w wyrażeniu <code>wyr</code> przez <code>wart</code> ;
<code>wyr /. {a → wart1, b → wart2}</code>	zastąpienie a oraz b przez <code>wart1</code> oraz <code>wart2</code> ;
<code>ReplaceAll[wyr, {a → wart1, b → wart2}]</code>	zastąpienie a oraz b przez <code>wart1</code> oraz <code>wart2</code> .

W ostatnim wyrażeniu podstawmy 3 za a :

In[26]:=

ww /. a → 3

Out[26]=

 $27 + 27 b + 9 b^2 + b^3$

In[27]:=

% /. {b → y}

Out[27]=

 $27 + 27 y + 9 y^2 + y^3$

Teraz zastąpimy a przez $p - q$, natomiast b przez $2 p + q$:

In[28]:=

`ww /. {a -> p - q, b -> 2 p + q}`

Out[28]=

$$(p - q)^3 + 3(p - q)^2(2p + q) + 3(p - q)(2p + q)^2 + (2p + q)^3$$

Uprośćmy otrzymany wynik:

In[29]:=

`Simplify[%]`

Out[29]=

 $27 p^3$

Do podstawień możemy wykorzystać także instrukcję `ReplaceAll`.

In[30]:=

`ReplaceAll[ww, {a -> 23, b -> -4}]`

Out[30]=

6859

Do przekształcania wyrażeń służą m.in. następujące instrukcje:

<code>Expand[wyr]</code>	rozwija wyrażenie wyr;
<code>Simplify[wyr]</code>	sprowadza wyrażenie wyr do prostszej postaci;
<code>FullSimplify[wyr]</code>	sprowadza wyrażenie wyr do prostszej postaci.

Uwaga. Wprowadzane instrukcje możemy kończyć średnikiem. Wówczas uzyskany wynik nie będzie wypisany na ekranie. Średnik może służyć także jako separator instrukcji w obrębie jednej komórki.

3. Wektory i macierze

In[31]:=

`Clear["Global`*"]`

3.1. Wektory

Jeżeli mamy do czynienia z obiektem złożonym, możemy próbować go przedstawić za pomocą listy. Lista to uporządkowany zbiór elementów, niekoniecznie tego samego typu. Listę możemy utworzyć, wymieniając jej elementy i ograniczając ją nawiasami "{" oraz "}". Elementy listy oddzielamy przecinkami. Listy służą też do reprezentowania wektorów i macierzy.

<code>{w_x, w_y, w_z}</code>	wektor $[w_x, w_y, w_z]$;
<code>w[[i]]</code>	odwołanie się do i - tej współrzędnej wektora w ;
<code>w + v, w - v</code>	suma oraz różnica wektorów w oraz v ;
<code>α * w</code> lub <code>α w</code>	iloczyn wektora w przez liczbę $α$;
<code>w.v</code> lub <code>Dot[w, v]</code>	iloczyn skalarny wektorów w i v ;
<code>Cross[w, v]</code>	iloczyn wektorowy podanych wektorów.

Zdefiniujmy wektor przestrzeni trójwymiarowej.

```
In[32]:=
u = {x, y, z}
```

```
Out[32]=
{x, y, z}
```

Wyberzmy drugą współrzędną tego wektora:

```
In[33]:=
u[[2]]
```

```
Out[33]=
y
```

Wektory możemy dodawać oraz mnożyć przez skalar. Jeśli do wektora będziemy dodawać niezdefiniowaną wcześniej zmienną q , to Mathematica potraktuje ją jako wektor, którego każda współrzędna jest równa q .

```
In[34]:=
p * u + q
```

```
Out[34]=
{q + p x, q + p y, q + p z}
```

Iloczyn skalarny wektorów oznaczamy kropką (taką samą jak na końcu zdania). Możemy go także policzyć, wykorzystując instrukcję `Dot`.

```
In[35]:=
{x, y} . {p, q}
```

```
Out[35]=
p x + q y
```

```
In[36]:=
Dot [ {x, y}, {p, q} ]
```

```
Out[36]=
p x + q y
```

Możemy także wyznaczać iloczyn wektorowy.

```
In[37]:=
Cross [ {a, b, c}, {x, y, z} ]
```

```
Out[37]=
{-c y + b z, c x - a z, -b x + a y}
```

Kilka dodatkowych instrukcji działających na wektorach.

<code>UnitVector[k]</code>	tworzy wektor jednostkowy w k – tym kierunku w przestrzeni dwuwymiarowej ($k \in \{1, 2\}$);
<code>UnitVector[n, k]</code>	tworzy wektor jednostkowy w k – tym kierunku w przestrzeni n – wymiarowej ($k \in \{1, 2, \dots, n\}$);
<code>VectorQ[w]</code>	sprawdza, czy podana zmienna jest wektorem;
<code>Norm[w]</code>	liczy normę euklidesową wektora w ;
<code>Norm[w, p]</code>	liczy p – normę wektora w , gdzie $p \geq 1$ lub $p = \infty$ (Infinity);

<code>VectorAngle[w, v]</code>	podaje kąt pomiędzy wektorami w i v;
<code>Projection[w, v]</code>	podaje rzut wektora w w kierunku wektora v;
<code>Normalize[w]</code>	podaje unormowanie wektora w czyli wektor $\frac{w}{ w }$;
<code>Orthogonalize[{w₁, w₂, ..., w_n}]</code>	podaje bazę ortonormalną uzyskana poprzez ortogonalizację podanych wektorów.

Określmy dwa wektory jednostkowe.

In[38]:=

UnitVector [1]

Out[38]=

{ 1, 0 }

In[39]:=

UnitVector [5, 3]

Out[39]=

{ 0, 0, 1, 0, 0 }

Jeśli chcemy sprawdzić, czy zmienna jest wektorem, możemy użyć instrukcji `VectorQ`.

In[40]:=

VectorQ [{ 1, 3, 7 }]

Out[40]=

True

In[41]:=

VectorQ [{ 1, { 3 }, 7 }]

Out[41]=

False

Instrukcja `Norm` wywołana z jednym argumentem ($w = (w_1, w_2, \dots, w_n)$) oblicza normę według zależności $||w|| = \sqrt{w \cdot w}$, natomiast z dwoma argumentami liczy p-normę, dla $p \in [1, \infty)$ ze wzoru:

$$||w||_p = \left(\sum_{i=1}^n |w_i|^p \right)^{1/p},$$

natomiast dla $p = \infty$ ze wzoru:

$$||w||_\infty = \max \{ |w_i| : i = 1, 2, \dots, n \}.$$

Zdefiniujmy wektor i policzmy kilka jego norm.

In[42]:=

v = { 1, 5, -1, 3 };

In[43]:=

Norm [v]

Out[43]=

6

In[44]:= **Norm [v, 2]**

Out[44]=
6

In[45]:= **Norm [v, 5]**

Out[45]=
 $3370^{1/5}$

In[46]:= **Norm [v, Infinity]**

Out[46]=
5

Zdefiniujmy teraz trzy wektory.

In[47]:=
**p1 = { 2, 3, 1 };
p2 = { 3, 2, -1 };
p3 = { -2, 1, 2 };**

Znajdźmy bazę ortonormalną generowaną przez te wektory.

In[50]:= **{ e1, e2, e3 } = Orthogonalize [{ p1, p2, p3 }]**

Out[50]=
 $\left\{ \left\{ \sqrt{\frac{2}{7}}, \frac{3}{\sqrt{14}}, \frac{1}{\sqrt{14}} \right\}, \left\{ 2\sqrt{\frac{2}{21}}, -\frac{1}{\sqrt{42}}, -\frac{5}{\sqrt{42}} \right\}, \left\{ -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}} \right\} \right\}$

Sprawdźmy prostopadłość otrzymanych wektorów.

In[51]:= **{ VectorAngle [e1, e2], VectorAngle [e2, e3], VectorAngle [e3, e1] }**

Out[51]=
 $\left\{ \frac{\pi}{2}, \frac{\pi}{2}, \text{ArcCos} \left[-\sqrt{\frac{2}{21}} + \sqrt{\frac{3}{14}} - \frac{1}{\sqrt{42}} \right] \right\}$

In[52]:= **Simplify [%]**

Out[52]=
 $\left\{ \frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2} \right\}$

Policzmy jeszcze normy tych wektorów.

In[53]:= **{ Norm [e1], Norm [e2], Norm [e3] }**

Out[53]=
{ 1, 1, 1 }

3.2. Macierze

In[54]:=

```
Clear["Global`*"]
```

Podobnie jak wektory, macierze są szczególnymi przypadkami list. Macierz możemy traktować jako wektor, którego elementami są wektory o tej samej liczbie współrzędnych.

$\{\{a_{11}, a_{12}\}, \{a_{21}, a_{22}\}\}$	macierz o wierszach (a_{11}, a_{12}) i (a_{21}, a_{22}) ;
$a[[i, j]]$	odwołanie się do elementu stojącego w i – tym wierszu oraz j – tej kolumnie macierzy a ;
$a.b$ lub $\text{Dot}[a, b]$	iloczyn macierzowy macierzy a i b ;
$a + b, a - b$	suma oraz różnica macierzy lub wektorów a oraz b ;
$\alpha * a$ lub αa	iloczyn macierzy albo wektora a przez liczbę α ;
$\text{Dimensions}[a]$	podaje wymiary macierzy;
$\text{MatrixForm}[a]$	wypisanie macierzy w tradycyjny sposób.

Zdefiniujmy macierz a oraz wektor v :

In[55]:=

```
a = {{1, 0, 3}, {4, 5, 1}, {7, 8, 4}};
```

```
v = {0.2, -1, 1};
```

Wypiszmy je w tradycyjny sposób:

In[57]:=

```
MatrixForm[a]
```

Out[57]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 3 \\ 4 & 5 & 1 \\ 7 & 8 & 4 \end{pmatrix}$$

In[58]:=

```
MatrixForm[v]
```

Out[58]//MatrixForm=

$$\begin{pmatrix} 0.2 \\ -1 \\ 1 \end{pmatrix}$$

Wypiszmy element a_{31} macierzy a :

In[59]:=

```
a[[3, 1]]
```

Out[59]=

7

Poszczególne elementy macierzy możemy wykorzystywać do różnych obliczeń.

In[60]:=

```
a[[3, 2]] + 3
```

Out[60]=

11

A także zmieniać ich wartości:

```
In[61]:=
a[[1, 2]] = -4
```

```
Out[61]=
-4
```

Obliczmy iloczyn $a v^T$. *Mathematica* sama domyśli się, że należy wektor v transponować:

```
In[62]:=
a.v
```

```
Out[62]=
{7.2, -3.2, -2.6}
```

```
In[63]:=
MatrixForm [%]
```

```
Out[63]//MatrixForm=

$$\begin{pmatrix} 7.2 \\ -3.2 \\ -2.6 \end{pmatrix}$$

```

Do definiowania macierzy, wektorów oraz ogólniej list możemy wykorzystywać instrukcję `Table`.

`Table[wzór, {licznik, start, koniec, krok}]` tworzy listę, obliczając jej elementy zgodnie z zadaniem wzorem, wykorzystując licznik zmieniający się od wartości start do wartości koniec z zadaniem krokiem.

W instrukcji `Table` można podać kilka liczników. Utworzymy wówczas listę, której elementami będą inne listy.

Utwórzmy wektor złożony z liczb parzystych mniejszych od dwudziestu:

```
In[64]:=
Table[i, {i, 2, 19, 2}]
```

```
Out[64]=
{2, 4, 6, 8, 10, 12, 14, 16, 18}
```

Jeśli krok zmiany licznika ma być równy jeden, to możemy go pominąć:

```
In[65]:=
Table[i, {i, 1, 10}]
```

```
Out[65]=
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Jeśli chcemy utworzyć macierz, musimy w instrukcji `Table` dodać drugi licznik. Pierwszy licznik będzie odpowiadał za wiersze macierzy, a drugi za kolumny macierzy.

```
In[66]:=
b = Table[i + j, {i, 1, 3}, {j, 4, 8, 2}]
```

```
Out[66]=
{{5, 7, 9}, {6, 8, 10}, {7, 9, 11}}
```

In[67]:=

MatrixForm [b]

Out[67]//MatrixForm=

$$\begin{pmatrix} 5 & 7 & 9 \\ 6 & 8 & 10 \\ 7 & 9 & 11 \end{pmatrix}$$

W tym przypadku licznik i przyjmuje trzy wartości (1,2,3), dlatego macierz ma trzy wiersze. Także licznik j przyjmuje trzy wartości (4,6,8), tym samym utworzona macierz ma trzy kolumny.

Możemy także określać macierze prostokątne.

In[68]:=

b = Table [i + j, {i, 0, 3}, {j, 4, 8, 2}]

Out[68]=

{ {4, 6, 8}, {5, 7, 9}, {6, 8, 10}, {7, 9, 11} }

In[69]:=

MatrixForm [b]

Out[69]//MatrixForm=

$$\begin{pmatrix} 4 & 6 & 8 \\ 5 & 7 & 9 \\ 6 & 8 & 10 \\ 7 & 9 & 11 \end{pmatrix}$$

Wymiary macierzy możemy odczytać instrukcją `Dimensions` (pierwsza liczba w wyniku, podaje liczbę wierszy, a druga liczbę kolumn).

In[70]:=

Dimensions [b]

Out[70]=

{4, 3}

Kolejne instrukcje działające na macierzach to:

<code>Det [a]</code>	liczy wyznacznik macierzy;
<code>Tr [a]</code>	liczy ślad macierzy;
<code>MatrixRank [a]</code>	liczy rząd macierzy;
<code>Inverse [a]</code>	znajduje macierz odwrotną;
<code>Transpose [a]</code>	podaje macierz transponowaną;
<code>Minors [a]</code>	podaje listę zawierającą wszystkie minory główne podanej macierzy;
<code>Minors [a, n]</code>	podaje listę zawierającą wszystkie minory stopnia n podanej macierzy;
<code>IdentityMatrix [n]</code>	podaje macierz jednostkową wymiaru $n \times n$;
<code>DiagonalMatrix[{d₁, ..., d_n}]</code>	podaje macierz diagonalną;
<code>MatrixQ [a]</code>	sprawdza, czy argument jest macierzą;
<code>PositiveDefiniteMatrixQ [a]</code>	sprawdza, czy macierz jest dodatnio określona.

Określmy nową macierz b i dodajmy do niej macierz a pomnożoną przez s , oznaczając wynik przez c :

In[71]:=

```
Clear [ b, c, s ];
b = { { 1, 0.5, 3 }, { 0, 1, 0 }, { -1, 2, 0 } };
c = b + s a;
```

In[74]:=

```
MatrixForm [ c ]
```

Out[74]//MatrixForm=

$$\begin{pmatrix} 1+s & 0.5-4s & 3+3s \\ 4s & 1+5s & s \\ -1+7s & 2+8s & 4s \end{pmatrix}$$

Policzmy jeszcze wyznacznik macierzy c:

In[75]:=

```
Det [ c ]
```

Out[75]=

$$3 + 22.5 s + 22.5 s^2 + 39 s^3$$

Sprawdźmy, ile jest równy wyznacznik macierzy a.

In[76]:=

```
Det [ a ]
```

Out[76]=

39

Macierz a jest nieosobliwa, możemy więc wyznaczyć macierz odwrotną do niej.

In[77]:=

```
a1 = Inverse [ a ]
```

Out[77]=

$$\left\{ \left\{ \frac{4}{13}, \frac{40}{39}, -\frac{19}{39} \right\}, \left\{ -\frac{3}{13}, -\frac{17}{39}, \frac{11}{39} \right\}, \left\{ -\frac{1}{13}, -\frac{12}{13}, \frac{7}{13} \right\} \right\}$$

Sprawdźmy poprawność obliczeń.

In[78]:=

```
a1.a == IdentityMatrix [ 3 ] == a.a1
```

Out[78]=

True

Transponujmy macierz a.

In[79]:=

```
Transpose [ a ]
```

Out[79]=

```
{ { 1, 4, 7 }, { -4, 5, 8 }, { 3, 1, 4 } }
```

Policzmy jej ślad:

```
In[80]:=  
Tr [ a ]
```

```
Out[80]=  
10
```

oraz sprawdźmy, czy macierz ta jest dodatnio określona:

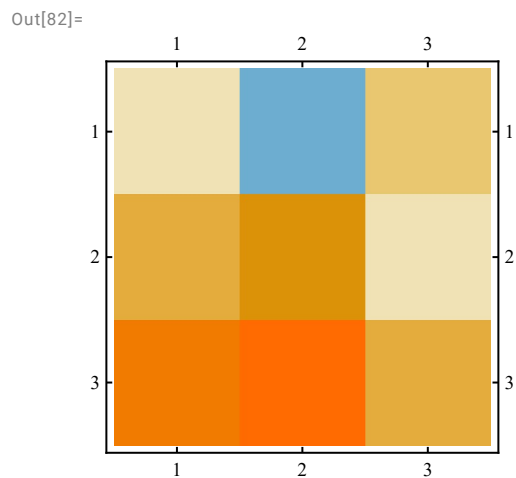
```
In[81]:=  
PositiveDefiniteMatrixQ [ a ]
```

```
Out[81]=  
False
```

`MatrixPlot[a]` rysuje obraz graficzny macierzy.

Zilustrujmy graficznie określoną wcześniej macierz a .

```
In[82]:=  
MatrixPlot [ a, ImageSize → 200 ]
```



Opcja `ImageSize` pozwala określać wielkość rysunku. Można ją stosować w różnych instrukcjach graficznych.

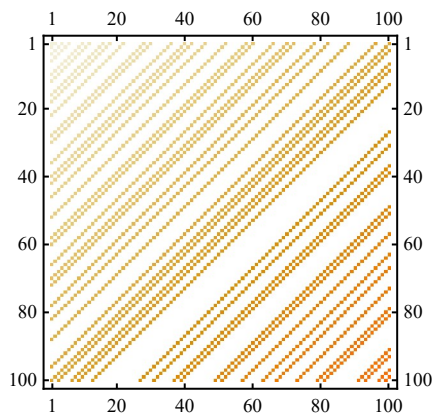
Jeszcze jeden przykład.

```
In[83]:=  
aa = Table [ If [ PrimeQ [ i + j ], i + j, 0 ], { i, 1, 100 }, { j, 1, 100 } ];
```

In[84]:=

MatrixPlot [aa, ImageSize → 200]

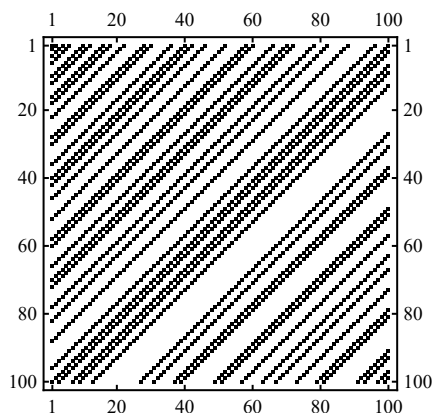
Out[84]=



In[85]:=

MatrixPlot [aa, ColorFunction → "Monochrome", ImageSize → 200]

Out[85]=



3.3. Wartości i wektory własne macierzy

In[86]:=

Clear ["Global`*"]

W pakiecie *Mathematica* możemy także wyznaczać wartości i wektory własne macierzy.

CharacteristicPolynomial [a, x]	podaje wielomian charakterystyczny (ze zmienną niezależną x) macierzy a;
Eigenvalues [a]	wyznacza wartości własne macierzy a;
Eigenvectors [a]	wyznacza wektory własne macierzy a;
Eigensystem [a]	podaje listę zawierającą wartości i wektory własne macierzy a.

Zdefiniujmy macierz i wyznaczmy jej wielomian charakterystyczny.

In[87]:=

a = { { 1, 0, -1 }, { 0, 1, 2 }, { 1, 1, -1 } };

In[88]:=

wiel = CharacteristicPolynomial [a, x]

Out[88]=

$$-2 + 2x + x^2 - x^3$$

Znajdźmy pierwiastki otrzymanego wielomianu, czyli wartości własne macierzy a .

In[89]:=

Solve [wiel == 0, x]

Out[89]=

$$\left\{ \left\{ x \rightarrow 1 \right\}, \left\{ x \rightarrow -\sqrt{2} \right\}, \left\{ x \rightarrow \sqrt{2} \right\} \right\}$$

Teraz wyznaczmy te same wartości instrukcją **Eigenvalues**.

In[90]:=

Eigenvalues [a]

Out[90]=

$$\left\{ -\sqrt{2}, \sqrt{2}, 1 \right\}$$

Wektory własne tej samej macierzy.

In[91]:=

Eigenvectors [a]

Out[91]=

$$\left\{ \left\{ \frac{1}{1 + \sqrt{2}}, -\frac{2}{1 + \sqrt{2}}, 1 \right\}, \left\{ -\frac{1}{-1 + \sqrt{2}}, \frac{2}{-1 + \sqrt{2}}, 1 \right\}, \{-1, 1, 0\} \right\}$$

Do tej samej macierzy zastosujemy teraz instrukcję **Eigensystem**. Wynikiem działania instrukcji **Eigensystem** jest lista $\{w, v\}$, gdzie w jest listą zawierającą obliczone wartości własne, a v jest listą zawierającą wektory własne.

In[92]:=

Eigensystem [a]

Out[92]=

$$\left\{ \left\{ -\sqrt{2}, \sqrt{2}, 1 \right\}, \left\{ \left\{ \frac{1}{1 + \sqrt{2}}, -\frac{2}{1 + \sqrt{2}}, 1 \right\}, \left\{ -\frac{1}{-1 + \sqrt{2}}, \frac{2}{-1 + \sqrt{2}}, 1 \right\}, \{-1, 1, 0\} \right\} \right\}$$

JordanDecomposition [a] rozkład Jordana macierzy.

Instrukcja ta podaje listę $\{c, j\}$, której elementami są: macierz j będąca postacią Jordana podanej macierzy a oraz macierz c będąca macierzą przejścia z bazy standardowej do bazy Jordana. Zachodzi wtedy wzór $a = c j c^{-1}$. Poszukajmy macierzy Jordana dla podanej poniżej macierzy a .

In[93]:=

a = { { 1, 0, -1 }, { 0, 1, 1 }, { 1, 1, -1 } };

In[94]:=

{ c, j } = JordanDecomposition [a]

Out[94]=

$$\left\{ \left\{ \{ 1, -1, 2 \}, \{ -1, 1, 0 \}, \{ 2, 0, 1 \} \right\}, \left\{ \{ -1, 0, 0 \}, \{ 0, 1, 1 \}, \{ 0, 0, 1 \} \right\} \right\}$$

```
In[95]:=
j // MatrixForm
```

```
Out[95]//MatrixForm=

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

```

```
In[96]:=
c // MatrixForm
```

```
Out[96]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}$$

```

```
In[97]:=
a == c.j.Inverse [ c ]
```

```
Out[97]=
True
```

4. Równania i nierówności

4.1. Zapis równania i jego rozwiązanie

```
In[98]:=
Clear [ "Global`*" ]
```

4.1.1. Zapis równania

W programie *Mathematica* równanie postaci $2x+5=0$ zapisujemy następująco:

```
In[99]:=
2 x + 5 == 0
```

```
Out[99]=
5 + 2 x == 0
```

Do znalezienia rozwiązania równania lub układu równań, możemy wykorzystać instrukcję `Solve`.

`Solve[równanie, zmienna]` rozwiązuje równanie względem podanej zmiennej;
`Solve[równanie, zmienna, dziedzina]` poszukuje rozwiązań w podanej dziedzinie (Reals, Integers, Complexes).

Rozwiążmy poprzednio podane równanie liniowe.

```
In[100]:=
roz = Solve [ 2 x + 5 == 0, x ]
```

```
Out[100]=

$$\left\{ \left\{ x \rightarrow -\frac{5}{2} \right\} \right\}$$

```

4.1.2. Podstawienia

Wynik, jaki otrzymujemy, jest zapisany w postaci zbioru tzw. reguł podstawienia, a interpretować można go w ten sposób, że jeżeli w rozwiązywanym równaniu w miejsce zmiennej x podstawimy $-\frac{5}{2}$, to równanie będzie spełnione.

Dzięki temu zapisowi możemy łatwo wykorzystywać znalezione rozwiązanie w innych obliczeniach.

wyrażenie /. {zmienna → wartość} zastępuje w wyrażeniu każde wystąpienie podanej zmiennej zadaną wartością;
 wyrażenie /. {z1 → w1, z2 → w2, ...} wykonuje kilka podstawień jednocześnie.

Do podstawień możemy wykorzystywać także instrukcję `ReplaceAll` w postaci `ReplaceAll[wyrażenie,{zmienna→wartość}]` lub `ReplaceAll[wyrażenie,{z1→w1,z2→w2,...}]`.

Wykonajmy proste podstawienie:

```
In[101]:=
  x + y + 34 /. {x → 6}
Out[101]=
  40 + y
```

A teraz dwa podstawienia jednocześnie.

```
In[102]:=
  x + y + 34 /. {x → 4, y → -22}
Out[102]=
  16
```

Możemy także podstawiać symbole lub inne wyrażenia.

```
In[103]:=
  x + y == 1 /. {x → t, y → (s + 1)^2}
Out[103]=
  (1 + s)^2 + t == 1
```

Sprawdźmy, czy wynik otrzymany przy rozwiązywaniu równania, jest poprawny.

```
In[104]:=
  2 x + 5 == 0 /. roz
Out[104]=
  { True }
```

4.1.3. Wybieranie fragmentu wyrażenia

Ponieważ rozwiązanie zawierało podwójne nawiasy klamrowe (bo jest to zbiór reguł podstawienia), więc jedna para nawiasów pojawiła się w ostatnim wyniku. Jeśli chcemy pozbyć się tej pary nawiasów, musimy wykorzystać tylko część otrzymanego rozwiązania.


wyrażenie_[numer] podaje wybraną część wyrażenia;
 wyrażenie_[numer1,numer2,...] podaje wybraną część podwyrażenia danego wyrażenia.

Zobaczmy, z jakich części składa się nasze rozwiązanie.

```
In[105]:=
  roz[[1]]
Out[105]=
  { x → -5/2 }
```

In[106]:=

roz[[2]]

Part: Part 2 of $\left\{\left\{x \rightarrow -\frac{5}{2}\right\}\right\}$ does not exist. 

Out[106]=

$\left\{\left\{x \rightarrow -\frac{5}{2}\right\}\right\} \llbracket 2 \rrbracket$

Możemy także pisać następująco.

In[107]:=

roz[[1]]

Out[107]=

$\left\{x \rightarrow -\frac{5}{2}\right\}$

Każde wyrażenie w programie *Mathematica* ma budowę grafu rozgałęziającego się w dół (jak korzeń). W powyższym przykładzie na pierwszym poziomie wyrażenie składa się tylko z jednego podwyrażenia (stąd komunikat, że część druga danego wyrażenia nie istnieje).

Zobaczmy, z jakich części zbudowane jest to podwyrażenie.

In[108]:=


roz[[1,1]]

Out[108]=

$x \rightarrow -\frac{5}{2}$

In[109]:=

roz[[1,2]]

Part: Part 2 of $\left\{x \rightarrow -\frac{5}{2}\right\}$ does not exist. 

Out[109]=

$\left\{\left\{x \rightarrow -\frac{5}{2}\right\}\right\} \llbracket 1, 2 \rrbracket$

Ponownie mamy tylko jedną część. Zobaczmy, co jest na trzecim poziomie.

In[110]:=

roz[[1,1,1]]

Out[110]=

x

In[111]:=


roz[[1,1,2]]

Out[111]=

$-\frac{5}{2}$

In[112]:=

roz_[1,1,3]

Part: Part 3 of $x \rightarrow -\frac{5}{2}$ does not exist. 

Out[112]=

$$\left\{ \left\{ x \rightarrow -\frac{5}{2} \right\} \right\} \llbracket 1, 1, 3 \rrbracket$$

Tym razem mamy dwie składowe. Przy okazji widzimy, w jaki sposób możemy uzyskać wartość liczbową naszego rozwiązania.

Teraz już wiemy, jak pozbyć się zbędnych nawiasów.

In[113]:=

2 x + 5 == 0 / . roz_[1]

Out[113]=

True

In[114]:=

2 x + 5 == 0 / . roz_[1]

Out[114]=

True

4.1.4. Rozwiązywanie równań (ciąg dalszy)

Możemy rozwiązywać równania z parametrem.

In[115]:=

Solve [**a * x + b == c, x**]

Out[115]=

$$\left\{ \left\{ x \rightarrow \frac{-b + c}{a} \right\} \right\}$$

Oczywiście w takim przypadku sami musimy zastanowić się, dla jakich wartości parametru znalezione rozwiązanie ma sens (w powyższym przykładzie dla $a \neq 0$).

W przypadku, gdy rozwiązujemy równania wielomianowe wyższych stopni, *Mathematica* poda wszystkie rozwiązania (nawet zespolone), jeśli jest w stanie je znaleźć. Dla równań do rzędu czwartego włącznie istnieją wzory na pierwiastki tych równań, natomiast dla równań wyższych stopni takich ogólnych wzorów już nie ma i stąd się biorą problemy z rozwiązywaniem takich równań.

Rozwiążmy równanie kwadratowe.

In[116]:=

Solve [**3 x^2 + x - 2 == 0, x**]

Out[116]=

$$\left\{ \left\{ x \rightarrow -1 \right\}, \left\{ x \rightarrow \frac{2}{3} \right\} \right\}$$

Rozwiążmy równanie czwartego stopnia.

In[117]:=

Solve[$x^4 + x^3 - 5x^2 + x - 6 == 0, x$]

Out[117]=

 $\{ \{x \rightarrow -3\}, \{x \rightarrow -i\}, \{x \rightarrow i\}, \{x \rightarrow 2\} \}$

Jeśli interesują nas tylko pierwiastki rzeczywiste, możemy ograniczyć rozwiązanie do liczb rzeczywistych.

In[118]:=

Solve[$x^4 + x^3 - 5x^2 + x - 6 == 0, x, \text{Reals}$]

Out[118]=

 $\{ \{x \rightarrow -3\}, \{x \rightarrow 2\} \}$

In[119]:=

Solve[$x^4 + x^3 - 5x^2 + x - 6 == 0, x \in \text{Reals}$]

Out[119]=

 $\{ \{x \rightarrow -3\}, \{x \rightarrow 2\} \}$

Jeśli *Mathematica* nie potrafi podać dokładnego rozwiązania równania wielomianowego, lub jego postać jest złożona, otrzymamy wynik w poniższej postaci.

In[120]:=

Solve[$x^3 - x^2 + 1 == 0, x$]

Out[120]=

 $\{ \{x \rightarrow \sqrt[3]{-0.755...}\}, \{x \rightarrow \sqrt[3]{0.877... - 0.745... i}\}, \{x \rightarrow \sqrt[3]{0.877... + 0.745... i}\} \}$

W przypadku wielomianu stopnia trzeciego możemy wypisać pierwiastki korzystając z opcji *Cubics*.

In[121]:=

Solve[$x^3 - x^2 + 1 == 0, x, \text{Cubics} \rightarrow \text{True}$]

Out[121]=

$$\left\{ \left\{ x \rightarrow \frac{1}{3} \left(1 - \left(\frac{2}{25 - 3\sqrt{69}} \right)^{1/3} - \left(\frac{1}{2} (25 - 3\sqrt{69}) \right)^{1/3} \right) \right\}, \right.$$

$$\left. \left\{ x \rightarrow \frac{1}{3} + \frac{1}{6} (1 + i\sqrt{3}) \left(\frac{1}{2} (25 - 3\sqrt{69}) \right)^{1/3} + \frac{1 - i\sqrt{3}}{3 \times 2^{2/3} (25 - 3\sqrt{69})^{1/3}} \right\}, \right.$$

$$\left. \left\{ x \rightarrow \frac{1}{3} + \frac{1}{6} (1 - i\sqrt{3}) \left(\frac{1}{2} (25 - 3\sqrt{69}) \right)^{1/3} + \frac{1 + i\sqrt{3}}{3 \times 2^{2/3} (25 - 3\sqrt{69})^{1/3}} \right\} \right\}$$

W przypadku wielomianów stopnia czwartego, możemy wykorzystać opcję *Quartics*. Możemy także szukać numerycznie przybliżonej wartości pierwiastka równania (co opiszemy w jednym z następnych podrozdziałów).

Rozwiążmy równanie trygonometryczne.

In[122]:=

Solve[Sin[2 x] == Sin[x], x]

Out[122]=

$$\left\{ \left\{ x \rightarrow 2\pi c_1 \text{ if } c_1 \in \mathbb{Z} \right\}, \left\{ x \rightarrow -\frac{\pi}{3} + 2\pi c_1 \text{ if } c_1 \in \mathbb{Z} \right\}, \right. \\ \left. \left\{ x \rightarrow \frac{\pi}{3} + 2\pi c_1 \text{ if } c_1 \in \mathbb{Z} \right\}, \left\{ x \rightarrow \pi + 2\pi c_1 \text{ if } c_1 \in \mathbb{Z} \right\} \right\}$$

Możemy także wykorzystać instrukcję Reduce.

In[123]:=

Reduce[Sin[2 x] == Sin[x], x]


Out[123]=

$$c_1 \in \mathbb{Z} \ \&\& \ \left(x == 2\pi c_1 \mid \mid x == -\frac{\pi}{3} + 2\pi c_1 \mid \mid x == \frac{\pi}{3} + 2\pi c_1 \mid \mid x == \pi + 2\pi c_1 \right)$$

Może się zdarzyć, że *Mathematica* nie potrafi rozwiązać jakiegoś równania:

In[124]:=

Solve[Log[x²] + (Log[x])² == 2, x]

 **Solve:** This system cannot be solved with the methods available to Solve. Try Reduce or FindInstance instead. 

Out[124]=

Solve[Log[x]² + Log[x²] == 2, x]

ale zastosowanie odpowiedniego podstawienia (w tym przypadku $t = \ln(x)$), umożliwi znalezienie poszukiwanego rozwiązania.

In[125]:=

rr = Solve[2 t + t² == 2, t]

Out[125]=

$$\left\{ \left\{ t \rightarrow -1 - \sqrt{3} \right\}, \left\{ t \rightarrow -1 + \sqrt{3} \right\} \right\}$$

W celu uzyskania rozwiązania wyjściowego równania, musimy jeszcze rozwiązać równania $\ln(x) = -1 - \sqrt{3}$ i $\ln(x) = -1 + \sqrt{3}$.

In[126]:=

Solve[Log[x] == rr_[1,1,2], x]

Out[126]=

$$\left\{ \left\{ x \rightarrow e^{-1 - \sqrt{3}} \right\} \right\}$$

In[127]:=

Solve[Log[x] == rr_[2,1,2], x]

Out[127]=

$$\left\{ \left\{ x \rightarrow e^{-1 + \sqrt{3}} \right\} \right\}$$

Wyjściowe równanie posiada więc dwa rozwiązania, którymi są liczby $e^{-1 - \sqrt{3}}$ i $e^{-1 + \sqrt{3}}$.

4.2. Układy równań i ich rozwiązanie

In[128]:=

```
Clear["Global`*"]
```

Do rozwiązywania układów równań także możemy wykorzystać instrukcję Solve:

`Solve[{równanie1, równanie2, ...}, zmienne]` rozwiązuje układ równań względem podanych zmiennych.

Znajdźmy rozwiązanie układu trzech równań liniowych:

In[129]:=

```
Solve[{x + y - z == 2, 2 x - 5 y + z == 1, 3 y + 2 z == -1}, {x, y, z}]
```

Out[129]=

$$\left\{ \left\{ x \rightarrow \frac{27}{23}, y \rightarrow \frac{3}{23}, z \rightarrow -\frac{16}{23} \right\} \right\}$$

oraz układu dwóch równań nieliniowych:

In[130]:=

```
Solve[{x^2 + y^2 == 4, x^2 + y^2 - 4 x + 3 == 0}, {x, y}]
```

Out[130]=

$$\left\{ \left\{ x \rightarrow \frac{7}{4}, y \rightarrow -\frac{\sqrt{15}}{4} \right\}, \left\{ x \rightarrow \frac{7}{4}, y \rightarrow \frac{\sqrt{15}}{4} \right\} \right\}$$

4.3. Instrukcja SolveAlways

In[131]:=

```
Clear["Global`*"]
```

<code>SolveAlways[równanie, zmienna]</code>	podaje wartości parametrów występujących w równaniu, dla których równanie jest zawsze prawdziwe;
<code>SolveAlways[{równanie1, równanie2, ...}, zmienne]</code>	działa tak samo tylko w przypadku układu równań.

Instrukcję tę możemy wykorzystać na przykład do znalezienia środka okręgu oraz jego promienia, gdy mamy dane równanie ogólne postaci:

$$x^2 + y^2 + a x + b y + c = 0$$

Wykonując to "ręcznie", musielibyśmy zastosować tzw. uzupełnianie do kwadratu lub pamiętać odpowiednie wzory. W programie *Mathematica* możemy wykorzystać instrukcję `SolveAlways` w celu znalezienia takich wartości parametrów s_x , s_y i r , aby równanie:

$$(x - s_x)^2 + (y - s_y)^2 - r^2 = x^2 + y^2 + a x + b y + c$$

było tożsamością.

Znajdźmy środek i promień okręgu o równaniu ogólnym $x^2 + y^2 + 2 x + 4 y + 1 = 0$.

In[132]:=

```
SolveAlways[(x - sx)^2 + (y - sy)^2 - r^2 == x^2 + y^2 + 2 x + 4 y + 1, {x, y}]
```

Out[132]=

$$\left\{ \left\{ r \rightarrow -2, s_x \rightarrow -1, s_y \rightarrow -2 \right\}, \left\{ r \rightarrow 2, s_x \rightarrow -1, s_y \rightarrow -2 \right\} \right\}$$

Otrzymaliśmy dwa rozwiązania, ale ponieważ r oznacza promień (który musi być dodatni), więc pierwsze z nich odrzucamy. Ostatecznie nasz okrąg ma środek w punkcie $S(-1, -2)$ i promień $r = 2$.

Wykorzystując implikację (instrukcja `Implies`) oraz instrukcję `SolveAlways`, możemy poszukiwać takich wartości parametrów, aby para liczb (x, y) będąca rozwiązaniem pierwszego równania była także rozwiązaniem drugiego równania.

Dobierzmy tak parametry a i b , aby liczby spełniające równanie $x - 2y = 0$ spełniały także równanie $ax^2 = by^2 + xy$.

In[133]:=

```
Clear[a, b];
```

In[134]:=

```
SolveAlways[Implies[x - 2y == 0, a * x^2 == b * y^2 + x * y], {x, y}]
```

Out[134]=

```
{{a -> 1/2 + b/4}}
```

4.4. Przybliżone wartości pierwiastków

In[135]:=

```
Clear["Global`*"]
```

Jak już wspomnieliśmy wcześniej, nie zawsze jest możliwe znalezienie dokładnej wartości rozwiązania równania lub układu równań, w takiej sytuacji możemy wykorzystać metody numeryczne poszukiwania tych rozwiązań. W programie *Mathematica* jest dostępnych kilka instrukcji znajdujących przybliżone wartości rozwiązań równań lub układów równań.

<code>NSolve[równanie, zmienna]</code>	numeryczne rozwiązanie równania;
<code>NSolve[równanie, zmienna, Reals]</code>	numeryczne rozwiązanie równania w dziedzinie rzeczywistej;
<code>NSolve[{równanie1, równanie2, ...}, zmienna]</code>	numeryczne rozwiązanie układu równań;
<code>FindRoot[równanie, {x, x₀}</code>	numeryczne rozwiązanie równania, startujące z punktu x_0 ;
<code>FindRoot[{rów1, rów2, ...}, {x, x₀}, {y, y₀}, ...]</code>	numeryczne rozwiązanie układu równań, startujące z punktu $\{x_0, y_0, \dots\}$.

Instrukcja `NSolve` jest przeznaczona (przede wszystkim) do rozwiązywania równań, w których obie strony są wielomianami, ale jeśli jest w stanie, to poda wszystkie pierwiastki. Instrukcja `FindRoot` pozwala natomiast znajdować przybliżoną wartość pierwiastka dowolnego równania, ale tylko jednego. Żeby poszukać kolejnego pierwiastka (o ile istnieje), musimy zmienić punkt startowy x_0 i wykonać ponowne obliczenia. W obliczeniach wykorzystano metodę Newtona (w przypadku jednowymiarowym zwaną także metodą stycznych).

Znajdźmy przybliżone wartości pierwiastków równania czwartego stopnia.

In[136]:=

```
NSolve[2x^4 - x^3 - 1 == 0, x]
```

Out[136]=

```
{{x -> -0.738984}, {x -> 0.119492 - 0.813835 i}, {x -> 0.119492 + 0.813835 i}, {x -> 1.}}
```

A teraz znajdziemy tylko pierwiastki rzeczywiste tego samego równania.

In[137]:=

```
NSolve[ $2x^4 - x^3 - 1 == 0$ ,  $x$ , Reals]
```

Out[137]=

```
{ {  $x \rightarrow -0.738984$  }, {  $x \rightarrow 1.$  } }
```

Rozwiążmy układ równań.

In[138]:=

```
NSolve[ $\{x^2 - y == 2, x^2 + y^2 == 2\}$ , {  $x, y$  }]
```

Out[138]=

```
{ {  $x \rightarrow -1.$ ,  $y \rightarrow -1.$  }, {  $x \rightarrow -1.41421$ ,  $y \rightarrow 0.$  }, {  $x \rightarrow 1.41421$ ,  $y \rightarrow 0.$  }, {  $x \rightarrow 1.$ ,  $y \rightarrow -1.$  } }
```

Poszukajmy instrukcją FindRoot pierwiastek równania trygonometrycznego.

In[139]:=

```
roz1 = FindRoot [  $\text{Sin}[x] == \text{Cos}[3x]$ , {  $x, 0$  } ]
```

Out[139]=

```
{  $x \rightarrow 0.392699$  }
```

Sprawdźmy z jaką dokładnością zostało rozwiązane równanie.

In[140]:=

```
Sin [  $x$  ] - Cos [  $3x$  ] /. roz1
```

Out[140]=

```
 $7.77156 \times 10^{-16}$ 
```

Możemy też zwiększyć dokładność wyniku.

In[141]:=

```
roz1a = FindRoot [  $\text{Sin}[x] == \text{Cos}[3x]$ , {  $x, 0$  }, WorkingPrecision  $\rightarrow 200$  ]
```

Out[141]=

```
{  $x \rightarrow$   

  0.3926990816987241548078304229099378605246461749218882276218680740384770507857761248`.  

  2850435316776463349776851081416028833088673057619382277896566992601606013968128551`.  

  283774231513819495557786861936629775 }
```

In[142]:=

```
Sin [  $x$  ] - Cos [  $3x$  ] /. roz1a
```

Out[142]=

```
 $0. \times 10^{-200}$ 
```

In[143]:=

```
Precision [ roz1[1, 2]]
```

Out[143]=

```
MachinePrecision
```

In[144]:=

```
N [%]
```

Out[144]=

```
15.9546
```

Znajdźmy inny pierwiastek tego samego równania.

```
In[145]:=
roz2 = FindRoot [ Sin [ x ] == Cos [ 3 x ] , { x , 2 } ]
```

```
Out[145]=
{ x → 1.9635 }
```

```
In[146]:=
Sin [ x ] - Cos [ 3 x ] /. roz2
```

```
Out[146]=
2.22045 × 10-16
```

Na koniec poszukamy przybliżonego rozwiązania układu równań trygonometrycznych.

```
In[147]:=
roz = FindRoot [ { Sin [ x ] == Cos [ y ] , Tan [ x ] == 3 Sin [ x ] } , { x , 1 } , { y , 1 } ]
```

```
Out[147]=
{ x → 1.23096 , y → 0.339837 }
```

```
In[148]:=
{ Sin [ x ] - Cos [ y ] , Tan [ x ] - 3 Sin [ x ] } /. roz
```

```
Out[148]=
{ 0. , 8.88178 × 10-16 }
```

4.5. Rozwiązywanie nierówności

```
In[149]:=
Clear [ "Global`*" ]
```

Do przekształcania nierówności i układów nierówności możemy wykorzystać instrukcję `Reduce`.

<code>Reduce[nierówność, zmienna]</code>	upraszcza podaną nierówność;
<code>Reduce[{n1, n2, ...}, x]</code>	upraszcza układ nierówności jednej zmiennej;
<code>Reduce[{n1, n2, ...}, {x1, x2, ...}]</code>	upraszcza układ nierówności wielu zmiennych;
<code>Reduce[nierówności, zmienne, zbiór]</code>	upraszcza nierówności zakładając, że zmienne należą do podanego zbioru, którym może być zbiór Reals, Complexes lub Integers.

Rozwiążmy nierówność kwadratową.

```
In[150]:=
Reduce [ x2 - 2 x - 2 ≤ 0 , x ]
```

```
Out[150]=
1 - √3 ≤ x ≤ 1 + √3
```

```
In[151]:=
Reduce [ x2 - 2 x + 4 > 0 , x ]
```

```
Out[151]=
x ∈ ℝ
```

Nierówność trygonometryczne.

In[152]:=

$$\text{Reduce}\left[\left\{-6\pi < x < 2\pi, \sin[x] > \frac{\sqrt{3}}{2}\right\}, x\right]$$

Out[152]=

$$-\frac{17\pi}{3} < x < -\frac{16\pi}{3} \quad || \quad -\frac{11\pi}{3} < x < -\frac{10\pi}{3} \quad || \quad -\frac{5\pi}{3} < x < -\frac{4\pi}{3} \quad || \quad \frac{\pi}{3} < x < \frac{2\pi}{3}$$

Układ nierówności.

In[153]:=

$$\text{Reduce}\left[\{x^2 + y^2 \leq 2, 2x - 3y < 1\}, \{x, y\}\right]$$

Out[153]=

$$\left(x == -\sqrt{2} \ \&\& \ y == 0\right) \quad || \quad \left(-\sqrt{2} < x < -1 \ \&\& \ -\sqrt{2-x^2} \leq y \leq \sqrt{2-x^2}\right) \quad || \quad \left(-1 \leq x < \frac{17}{13} \ \&\& \ \frac{1}{3}(-1+2x) < y \leq \sqrt{2-x^2}\right)$$

Sprawdźmy, jakie liczby całkowite spełniają powyższy układ nierówności.

In[154]:=

$$\text{Reduce}\left[\{x^2 + y^2 \leq 2, 2x - 3y < 1\}, \{x, y\}, \text{Integers}\right]$$

Out[154]=

$$(x == -1 \ \&\& \ y == 0) \quad || \quad (x == -1 \ \&\& \ y == 1) \quad || \quad (x == 0 \ \&\& \ y == 0) \quad || \quad (x == 0 \ \&\& \ y == 1) \quad || \quad (x == 1 \ \&\& \ y == 1)$$

W rozwiązywanych nierównościach możemy wykorzystywać także wartość bezwzględną (moduł).

In[155]:=

$$\text{Reduce}\left[\text{Abs}[x^2 - 7] > 1, x\right]$$

Out[155]=

$$\text{Re}[x] < -2\sqrt{2} \quad || \quad \left(-2\sqrt{2} \leq \text{Re}[x] \leq -\sqrt{6} \ \&\& \ \left(\text{Im}[x] < -\sqrt{-7 - \text{Re}[x]^2 + \sqrt{1 + 28\text{Re}[x]^2}} \quad || \quad \text{Im}[x] > \sqrt{-7 - \text{Re}[x]^2 + \sqrt{1 + 28\text{Re}[x]^2}}\right)\right) \quad || \quad -\sqrt{6} < \text{Re}[x] < \sqrt{6} \quad || \quad \left(\sqrt{6} \leq \text{Re}[x] \leq 2\sqrt{2} \ \&\& \ \left(\text{Im}[x] < -\sqrt{-7 - \text{Re}[x]^2 + \sqrt{1 + 28\text{Re}[x]^2}} \quad || \quad \text{Im}[x] > \sqrt{-7 - \text{Re}[x]^2 + \sqrt{1 + 28\text{Re}[x]^2}}\right)\right) \quad || \quad \text{Re}[x] > 2\sqrt{2}$$

In[156]:=

$$\text{Reduce}\left[\text{Abs}[x^2 - 7] > 1, x, \text{Reals}\right]$$

Out[156]=

$$x < -2\sqrt{2} \quad || \quad -\sqrt{6} < x < \sqrt{6} \quad || \quad x > 2\sqrt{2}$$

Nierówność wymierna.

In[157]:=

$$\text{Reduce}\left[\frac{(x+1)(x+2)^2}{x^4 + x^3 - 5x^2 + x - 6} \geq 0, x\right]$$

Out[157]=

$$-3 < x \leq -1 \quad || \quad x > 2$$

Możemy także rozwiązywać nierówności podwójne.

In[158]:=

$$\text{Reduce}\left[1 \leq \frac{x+1}{x^2+x-1} < 2, x\right]$$

Out[158]=

$$-\frac{3}{2} < x \leq -\sqrt{2} \quad || \quad 1 < x \leq \sqrt{2}$$

Nierówność wykładnicza.

In[159]:=

$$\text{Reduce}\left[2^x \geq 16, x\right]$$

Out[159]=

$$c_1 \in \mathbb{Z} \ \&\& \ \text{Im}[x] = \frac{2\pi c_1}{\text{Log}[2]} \ \&\& \ \text{Re}[x] \geq \frac{\text{Log}[16 \text{Sec}[\text{Im}[x] \text{Log}[2]]]}{\text{Log}[2]}$$

In[160]:=

$$\text{Reduce}\left[2^x \geq 16, x, \text{Reals}\right]$$

Out[160]=

$$x \geq 4$$

In[161]:=

$$\text{Reduce}\left[2^x \geq 16, x, \text{Integers}\right]$$

Out[161]=

$$x \in \mathbb{Z} \ \&\& \ x \geq 4$$

Do ilustracji graficznej zbiorów rozwiązań nierówności możemy wykorzystać instrukcję RegionPlot.

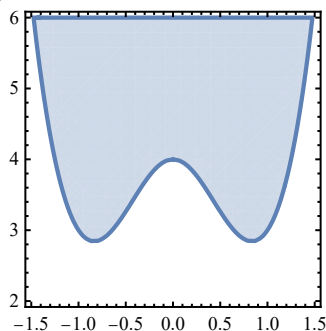
`RegionPlot[nierówność, {x, xmin, xmax}, {y, ymin, ymax}]` zaznacza w podanym prostokącie zbiór punktów spełniających zadaną nierówność.

Pierwszy przykład.

In[162]:=

$$\text{RegionPlot}\left[x^4 + \frac{4}{1+x^2} \leq y, \{x, -1.5, 1.5\}, \{y, 2, 6\}, \text{ImageSize} \rightarrow 150\right]$$

Out[162]=

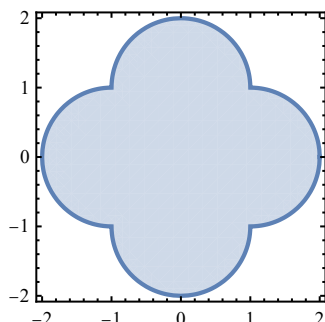


Drugi przykład. Tym razem kilka nierówności połączonych spójnikiem logicznym “lub”.

In[163]:=

```
RegionPlot[x2 + y2 + 2x ≤ 0 ∨ x2 + y2 - 2x ≤ 0 ∨ x2 + y2 + 2y ≤ 0 ∨ x2 + y2 - 2y ≤ 0,
{x, -2, 2}, {y, -2, 2}, ImageSize → 150]
```

Out[163]=

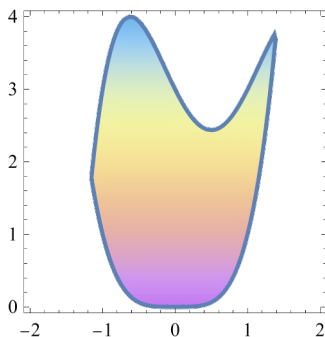


Możemy także zmieniać kolorowanie wykreślanego obszaru.

In[164]:=

```
RegionPlot[y > x4 ∧ y < -x4 + 2x3 + x2 - 2x + 3, {x, -2, 2},
{y, 0, 4}, ColorFunction → ColorData["Pastel"], ImageSize → 150]
```

Out[164]=



4.6. Równania rekurencyjne

In[165]:=

```
Clear["Global`*"]
```

Mathematica umożliwia także rozwiązywanie równań rekurencyjnych, czyli znalezienia wzoru na ogólny wyraz ciągu danego w postaci rekurencyjnej.

<code>RSolve[równanie, a[n], n]</code>	podaje rozwiązanie równania rekurencyjnego ze względu na $a[n]$;
<code>RSolve[równanie, a, n]</code>	rozwiązanie jest podawane w postaci funkcji prymitywnej;
<code>RSolve[równania, {a1, a2, ...}, n]</code>	podaje rozwiązanie układu równań rekurencyjnych.

Instrukcja `RSolve` podaje wynik w postaci reguły podstawiania, natomiast bezpośrednio sam wzór na n -ty wyraz ciągu możemy uzyskać instrukcją `RSolveValue`.

Bez podania wartości początkowej uzyskamy rozwiązanie zależne od stałej dowolnej.

In[166]:=

RSolve [a [n] == 2 a [n - 1] + 3, a [n], n]

Out[166]=

 $\{ \{ a [n] \rightarrow -3 (1 - 2^n) + 2^{-1+n} c_1 \} \}$

Podajmy pierwszą wartość ciągu:

In[167]:=

RSolve [{ a [n] == 2 a [n - 1] + 3, a [1] == 1 }, a [n], n]

Out[167]=

 $\{ \{ a [n] \rightarrow -3 + 2^{1+n} \} \}$

A teraz inny przykład:

In[168]:=

RSolve [{ a [n] == a [n - 1] + a [n - 2], a [1] == 1, a [2] == 1 }, a [n], n]

Out[168]=

 $\{ \{ a [n] \rightarrow \text{Fibonacci} [n] \} \}$

Jeszcze jeden przykład. Tym razem rozwiązanie wyznaczmy w postaci funkcji prymitywnej.

In[169]:=

ciag = { b [n] == 3 b [n - 1] + 2, b [1] == 1 }

Out[169]=

 $\{ b [n] == 2 + 3 b [-1 + n], b [1] == 1 \}$

In[170]:=

rr = RSolve [ciag, b, n]

Out[170]=

 $\{ \{ b \rightarrow \text{Function} [\{ n \}, \frac{1}{3} (-3 + 2 \times 3^n)] \} \}$

Ułatwia to wykorzystanie wyniku w dalszych obliczeniach:

In[171]:=

ciag /. rr[[1]] // Simplify

Out[171]=

 $\{ \text{True}, \text{True} \}$

Wartości ciągu możemy policzyć na dwa sposoby:

In[172]:=

Table [b [n] /. rr[[1]], { n, 1, 12 }]

Out[172]=

 $\{ 1, 5, 17, 53, 161, 485, 1457, 4373, 13121, 39365, 118097, 354293 \}$

In[173]:=

Table [rr[[1, 1, 2]][n], { n, 1, 12 }]

Out[173]=

 $\{ 1, 5, 17, 53, 161, 485, 1457, 4373, 13121, 39365, 118097, 354293 \}$

Wykorzystując instrukcję RSolveValue dostaniemy bezpośrednio wzór na n-ty wyraz ciągu.

```
In[174]:= RSolveValue[{a[n] == 2 a[n - 1] + 3, a[1] == 1}, a[n], n]
```

```
Out[174]= -3 + 21+n
```

```
In[175]:= RSolveValue[{a[n] == 2 a[n - 1] + 3, a[1] == 1}, a, n]
```

```
Out[175]= Function[{n}, -3 + 21+n]
```

5. Analiza matematyczna

5.1. Granice

```
In[176]:= Clear["Global`*"]
```

Do wyznaczenia granicy ciągu możemy wykorzystać instrukcję `Limit`.

`Limit[an, n → ∞]` liczy granicę a_n, gdy n → ∞.

Jeżeli *Mathematica* nie potrafi policzyć granicy, powtórz w komórce wyjściowej treść wejścia.

Policzmy $\lim_{n \rightarrow \infty} \frac{n^2 - n}{n^3 + n^2 + n + 2}$:

```
In[177]:= Limit[ $\frac{n^2 - n}{n^3 + n^2 + n + 2}$ , n → ∞]
```

```
Out[177]= 0
```

Policzmy granicę ciągu związanego z liczbą Eulera:

```
In[178]:= Limit[(1 +  $\frac{1}{n}$ )n, n → ∞]
```

```
Out[178]= e
```

oraz jeszcze jedną granicę:

```
In[179]:= Limit[ $\sqrt{n^2 + 2n + 3} - n + 4$ , n → ∞]
```

```
Out[179]= 5
```

Możemy także liczyć granice funkcji.

`Limit[f[x], x → x0]` liczy granicę funkcji f w punkcie x₀.

Policzmy granicę $\lim_{x \rightarrow 0} \frac{\sin x}{x}$.

In[180]:=

```
Limit [Sin [x] / x, x → 0]
```

Out[180]=

1

Dla instrukcji Limit istnieje opcja, pozwalająca liczyć granice jednostronne.

Direction → -1 wymusza liczenie granicy prawostronnej;
 Direction → 1 wymusza liczenie granicy lewostronnej.

Jeśli chcemy policzyć granicę prawostronną, zamiast wartości "-1" możemy także podać "FromAbove". W przypadku granicy lewostronnej, zamiast wartości "1" możemy podać "FromBelow".

Zdefiniujmy funkcję, którą będziemy badać:

In[181]:=

```
Clear [y];
y [x_] :=  $\frac{x^2 - x - 4}{x - 2}$ ;
```

Policzmy granice na krańcach dziedziny funkcji:

In[183]:=

```
Limit [y [x], x → -∞]
```

Out[183]=

-∞

In[184]:=

```
Limit [y [x], x → ∞]
```

Out[184]=

∞

In[185]:=

```
Limit [y [x], x → 2, Direction → -1]
```

Out[185]=

-∞

In[186]:=

```
Limit [y [x], x → 2, Direction → 1]
```

Out[186]=

∞

In[187]:=

```
Limit [y [x], x → 2, Direction → "FromAbove"]
```

Out[187]=

-∞

```
In[188]:= Limit[y[x], x → 2, Direction → "FromBelow"]
```

```
Out[188]= ∞
```

Kolejny przykład.

```
In[189]:= Clear[y];
y[x_] :=  $\frac{e^x}{x+1} + 2$ ;
```

Zbadajmy granice na krańcach dziedziny funkcji:

```
In[191]:= Limit[y[x], x → -∞]
```

```
Out[191]= 2
```

```
In[192]:= Limit[y[x], x → ∞]
```

```
Out[192]= ∞
```

```
In[193]:= Limit[y[x], x → -1, Direction → -1]
```

```
Out[193]= ∞
```

```
In[194]:= Limit[y[x], x → -1, Direction → 1]
```

```
Out[194]= -∞
```

5.2. Różniczkowanie

```
In[195]:= Clear["Global`*"]
```

Przejdziemy do liczenia pochodnych.

$D[f, x]$	podaje pochodną funkcji f względem zmiennej x ($\frac{df}{dx}$); w przypadku funkcji wielu zmiennych, podaje odpowiednią pochodną cząstkową ($\frac{\partial f}{\partial x}$);
$D[f, x, y]$	podaje pochodną cząstkową $\frac{\partial^2 f}{\partial x \partial y}$;
$D[f, \{x, n\}]$	podaje pochodną $\frac{d^n f}{dx^n}$ albo $\frac{\partial^n f}{\partial x^n}$;
$D[f, x, \text{NonConstants} \rightarrow y]$	podaje pochodną funkcji $f(x, y(x))$.

Policzmy pierwszą pochodną funkcji $x^2 \cos x$:

In[196]:=

D[$x^2 \text{Cos}[x]$, x]

Out[196]=

 $2x \text{Cos}[x] - x^2 \text{Sin}[x]$

A teraz czwartą pochodną tej funkcji:

In[197]:=

D[$x^2 \text{Cos}[x]$, { x , 4}]

Out[197]=

 $-12 \text{Cos}[x] + x^2 \text{Cos}[x] + 8x \text{Sin}[x]$

Tak samo możemy różniczkować funkcje napisane przez użytkownika. Pokażemy to demonstrując liczenie pochodnych cząstkowych funkcji dwu (wielu) zmiennych. W tym celu zdefiniujemy funkcję:

In[198]:=

Clear[f]; $f[x_, y_] := x^2 y + 2 y^2 - 3 x y + 4 x + 7;$

Policzmy pochodną $\frac{\partial^3 f(x,y)}{\partial x^2 \partial y}$:

In[200]:=

D[$f[x, y]$, { x , 2}, y]

Out[200]=

2

Założmy, że y jest funkcją zmiennej x , czyli $y=y(x)$ i teraz zróżniczkujmy funkcję f :

In[201]:=

D[$f[x, y]$, x , **NonConstants** → y]

Out[201]=

 $4 - 3y + 2xy - 3xD[y, x, \text{NonConstants} \rightarrow \{y\}] + x^2 D[y, x, \text{NonConstants} \rightarrow \{y\}] + 4y D[y, x, \text{NonConstants} \rightarrow \{y\}]$

W powyższej odpowiedzi $D[y, x, \text{NonConstants} \rightarrow \{y\}]$ oznacza nieznaną pochodną $\frac{dy(x)}{dx}$.

Alternatywny sposób oznaczania operacji różniczkowania (dostępny z palety *Basic Math Input*).

$\partial_x f$ odpowiednik $D[f, x]$;

$\partial_{x,y} f$ odpowiednik $D[f, x, y]$.

Skorzystajmy z "tradycyjnego" sposobu oznaczania operacji różniczkowania:

In[202]:=

 $\partial_x \text{Sin}[x]$

Out[202]=

 $\text{Cos}[x]$

In[203]:= $\partial_{x,y}(x^2 y + 2 y^4 - x + 7 y)$

Out[203]=
2 x

5.3. Całka nieoznaczona i oznaczona

In[204]:= `Clear["Global`*"]`

Poniższe instrukcje pozwalają liczyć całkę nieoznaczoną lub oznaczoną.

<code>Integrate[f, x]</code>	liczy całkę nieoznaczoną funkcji f względem x ; można zamiennie używać operatora $\int \square \times d\square$ z palety Basic Math Input;
<code>Integrate[f, {x, a, b}]</code>	liczy całkę oznaczoną funkcji f względem x , a, b – granice całkowania; można zamiennie używać operatora $\int_a^b \square \times d\square$ z palety Basic Math Input;
<code>Integrate[f, {x, a, b}, {y, c, d}]</code> <code>NIntegrate[f, {x, a, b}]</code>	liczy całkę funkcji $f(x, y)$ po prostokącie $a \leq x \leq b$, $c \leq y \leq d$; wartość całki wyznaczana jest numerycznie.

Policzmy całkę nieoznaczoną z funkcji $3x^2 - 2x + \cos x$:

In[205]:= $\int (3x^2 - 2x + \text{Cos}[x]) dx$

Out[205]=
 $-x^2 + x^3 + \text{Sin}[x]$

Należy zauważyć, że w wyniku nie jest wypisywana stała całkowania.

Możemy całkować funkcje z parametrem:

In[206]:= $\int (ax^4 - bx + 4) dx$

Out[206]=
 $4x - \frac{bx^2}{2} + \frac{ax^5}{5}$

Policzmy teraz całkę oznaczoną:

In[207]:= $\int_0^\pi (3x^2 - 2x + \text{Cos}[x]) dx$

Out[207]=
 $(-1 + \pi) \pi^2$

Policzmy całkę podwójną po prostokącie $0 \leq x \leq 2$, $1 \leq y \leq 3$ z funkcji $z(x, y) = 4x^2 y^2 + x - 3y + 1$:

In[208]:= `Integrate[4 x^2 y^2 + x - 3 y + 1, {x, 0, 2}, {y, 1, 3}]`

Out[208]=
$$\frac{688}{9}$$

Jeśli *Mathematica* "nie potrafi" policzyć całki oznaczonej, powtarza wejście na wyjściu:

In[209]:= `Integrate[2^Sin[x], {x, 0, 1}]`

Out[209]=
$$\int_0^1 2^{\sin[x]} dx$$

Możemy wtedy spróbować policzyć taką całkę numerycznie:

In[210]:= `NIntegrate[2^Sin[x], {x, 0, 1}]`

Out[210]= 1.3954

Możemy zmieniać dokładność obliczeń.

In[211]:= `NIntegrate[2^Sin[x], {x, 0, 1}, WorkingPrecision -> 50]`

Out[211]= 1.3953964083274320677743488544483104235401142563613

5.4. Zastosowania całki oznaczonej

In[212]:= `Clear["Global`*"]`

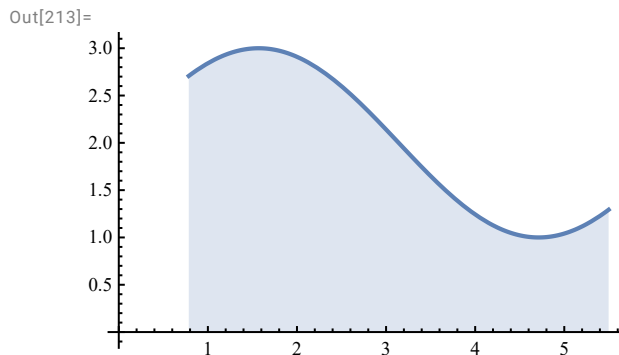
5.4.1. Pole obszaru ograniczonego krzywymi

Przykład 1

Obliczyć pole obszaru ograniczonego krzywą $y = 2 + \sin x$, prostymi $x = \frac{\pi}{4}$, $x = \frac{7}{4}\pi$ oraz osią Ox .

Wykonajmy rysunek do naszego zadania:

In[213]:= `Plot[2 + Sin[x], {x, $\frac{\pi}{4}$, $\frac{7\pi}{4}$ }, Filling -> Axis, AxesOrigin -> {0, 0}, ImageSize -> 250]`



Pole można obliczyć, posługując się całką:

In[214]:=

$$\text{pole} = \int_{\frac{\pi}{4}}^{\frac{7}{4}\pi} (2 + \sin[x]) dx$$

Out[214]=

3π

Przykład 2

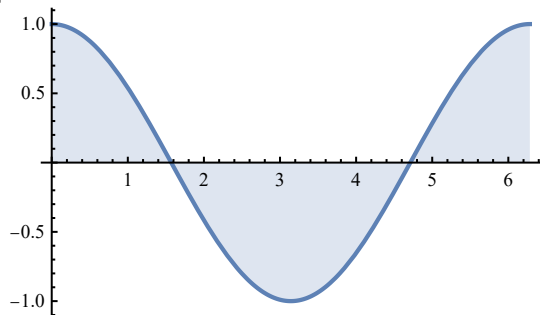
Obliczyć pole obszaru ograniczonego krzywą $y = \cos x$ i prostymi $x = 0$, $x = 2\pi$ oraz osią Ox .

Jak poprzednio, zaczniemy od rysunku:

In[215]:=

`Plot[Cos[x], {x, 0, 2π}, Filling → Axis, ImageSize → 250]`

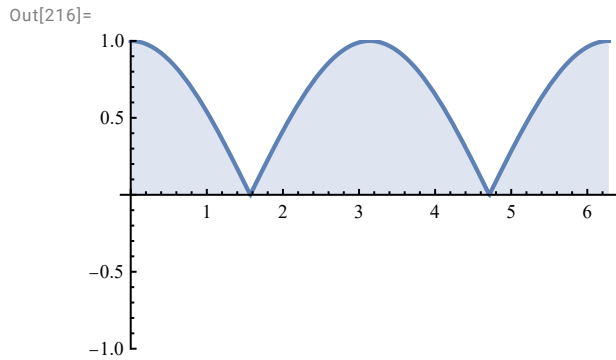
Out[215]=



Łatwo zauważyć, że pole to jest równe polu, jakie przy tych samych pozostałych ograniczeniach wyznaczy wykres funkcji $y_1(x) = |y(x)| = |\cos x|$. Użyjemy zatem funkcji `Abs`:

In[216]:=

`Plot[Abs[Cos[x]], {x, 0, 2π}, PlotRange → {-1, 1}, Filling → Axis, ImageSize → 250]`



Policzmy więc pole:

In[217]:=

$$\text{pole} = \int_0^{2\pi} \text{Abs}[\text{Cos}[x]] \, dx$$

Out[217]=

4

Przykład 3

Obliczyć pole ograniczone wykresami krzywych $y = |x|$ i $y = -x^2 + x + 2$.

Musimy najpierw wyznaczyć punkty, w których krzywe przecinają się:

In[218]:=

$$\text{roz} = \text{Solve}[\{y == \text{Abs}[x], y == -x^2 + x + 2\}, \{x, y\}]$$

Out[218]=

$$\left\{ \left\{ x \rightarrow \sqrt{2}, y \rightarrow \sqrt{2} \right\}, \left\{ x \rightarrow 1 - \sqrt{3}, y \rightarrow -1 + \sqrt{3} \right\} \right\}$$

In[219]:=

$$\text{roz}[[2, 1, 2]]$$

Out[219]=

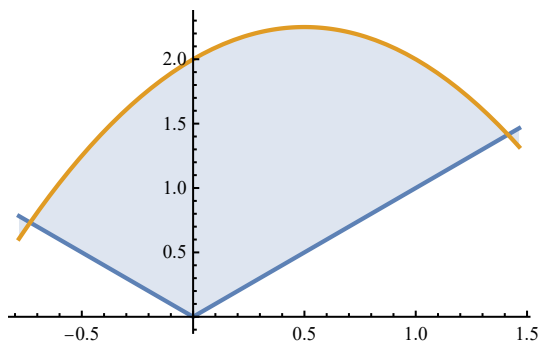
$$1 - \sqrt{3}$$

Teraz możemy wykonać rysunek do naszego zadania:

In[220]:=

$$\text{Plot}[\{\text{Abs}[x], -x^2 + x + 2\}, \{x, \text{roz}[[2, 1, 2]] - 0.05, \text{roz}[[1, 1, 2]] + 0.05\}, \\ \text{Filling} \rightarrow \{1 \rightarrow \{2\}\}, \text{ImageSize} \rightarrow 250]$$

Out[220]=



Wiemy, że aby policzyć pole, należy całkować różnicę: "ograniczenie górne" minus "ograniczenie dolne", pomiędzy odciętymi punktów przecięcia się wykresów, zatem przystępujemy do obliczeń:

In[221]:=

$$\text{pole} = \int_{\text{roz}[2,1,2]}^{\text{roz}[1,1,2]} ((-x^2 + x + 2) - \text{Abs}[x]) dx$$

Out[221]=

$$\frac{2}{3} (-4 + 2\sqrt{2} + 3\sqrt{3})$$

In[222]:=

N [pole]

Out[222]=

2.68305

5.4.2. Objętość i pole powierzchni bocznej brył obrotowych

Przykład 1

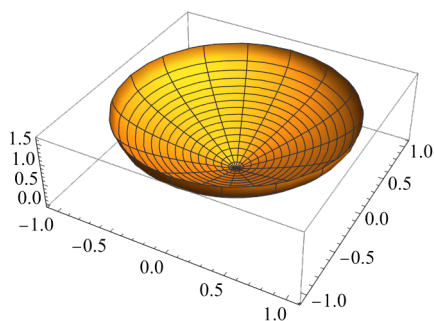
Policzyć pole powierzchni oraz objętość bryły ograniczonej powierzchnią powstałą przez obrót krzywej $y = \sin x$, $x \in [0, \pi]$, dookoła osi Ox .

Najpierw wykonamy rysunek:

In[223]:=

p1 = RevolutionPlot3D [ArcSin [x], { x, 0, 1 }, ImageSize → 200]

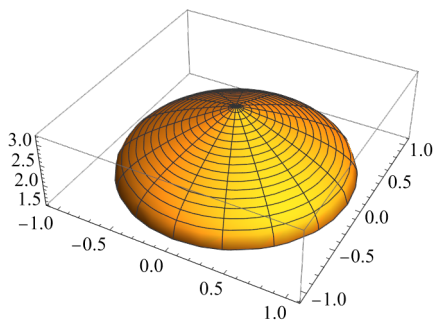
Out[223]=



In[224]:=

p2 = RevolutionPlot3D [π + ArcSin [x], { x, -1, 0 }, ImageSize → 200]

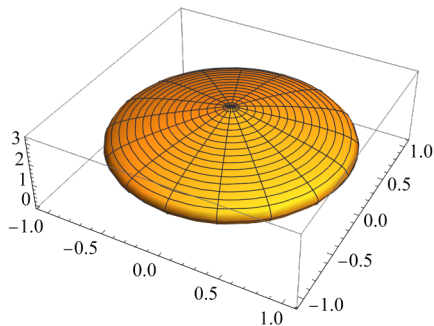
Out[224]=



In[225]:=

Show [p1, p2, PlotRange → All, ImageSize → 200]

Out[225]=



Aby narysować interesujący nas obiekt, musieliśmy skorzystać z funkcji arcsin, bowiem instrukcja RevolutionPlot3D obraca wykres wokół osi pionowej.

Policzymy pole, korzystając ze wzoru $P = 2\pi \int_a^b f(x) \sqrt{1 + (f'(x))^2} dx$, gdy obracamy krzywą $y = f(x)$, $x \in [a, b]$:

In[226]:=

$$\text{pole} = 2\pi \int_0^{\pi} \text{Sin}[x] \sqrt{1 + \text{Cos}[x]^2} dx$$

Out[226]=

$$2\pi \left(\sqrt{2} + \text{ArcCoth}[\sqrt{2}] \right)$$

Odczytamy wartość numeryczną wyniku:

In[227]:=

N [%]

Out[227]=

14.4236

Objętość bryły ($V = \pi \int_a^b [f(x)]^2 dx$):

In[228]:=

$$\text{objętość} = \pi \int_0^{\pi} \text{Sin}[x]^2 dx$$

Out[228]=

$$\frac{\pi^2}{2}$$

In[229]:=

N [%]

Out[229]=

4.9348

Przykład 2

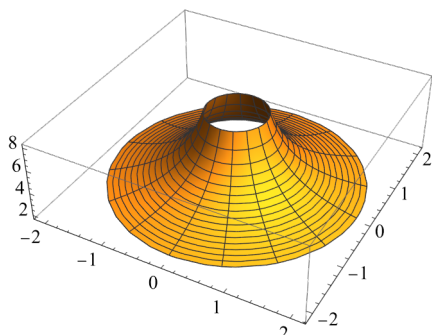
Policzyć pole powierzchni bocznej oraz objętość bryły ograniczonej powierzchnią powstałą przez obrót krzywej $y = \frac{4}{x}$, $x \in [0.5, 2]$, dookoła osi OY.

W tym zadaniu wykonanie rysunku nie stanowi problemu:

In[230]:=

```
RevolutionPlot3D[ $\frac{4}{x}$ , {x, 0.5, 2}, PlotPoints -> 30, ImageSize -> 200]
```

Out[230]=



Aby wyznaczyć pole powierzchni bocznej i objętość tej bryły, musimy znaleźć funkcję odwrotną. Łatwo wykazać, że jest to funkcja $x(y) = \frac{4}{y}$, $y \in [2, 8]$.

In[231]:=

$$\text{pole} = 2 \pi \int_2^8 \frac{4}{y} \sqrt{1 + \frac{16}{y^4}} dy$$

Out[231]=

$$2 \pi \left(2 \sqrt{2} - \frac{\sqrt{257}}{8} - 2 \text{ArcTanh} \left[\frac{1}{\sqrt{2}} \right] + 2 \text{ArcTanh} \left[\frac{16}{\sqrt{257}} \right] \right)$$

In[232]:=

```
N[pole]
```

Out[232]=

37.6689

In[233]:=

$$\text{obje} = \pi \int_2^8 \frac{16}{y^2} dy$$

Out[233]=

6 π

In[234]:=

```
N[obje]
```

Out[234]=

18.8496

5.5. Poszukiwanie ekstremów

In[235]:=

```
Clear["Global`*"]
```

Możemy poszukiwać przybliżonych wartości minimów i maksimów lokalnych, korzystając z poniższych instrukcji programu.

<code>FindMinimum[f, {x, x0}]</code>	poszukuje minimum lokalnego funkcji f, startując z podanego punktu x0;
<code>FindMinimum[f, {x, x0, x1}]</code>	poszukuje minimum lokalnego funkcji f, używając x0 i x1 jako pierwszych dwu wartości zmiennej x; postać stosowana, gdy nie można policzyć symbolicznie pochodnej funkcji f;
<code>FindMinimum[f, {{x, x0}, {y, y0}, ...}]</code>	poszukuje minimum lokalnego funkcji wielu zmiennych.

Do poszukiwania maksimum lokalnego możemy wykorzystać instrukcję `FindMaximum`, która ma takie same argumenty jak instrukcja `FindMinimum`.

Poszukajmy minimum lokalnego funkcji postaci:

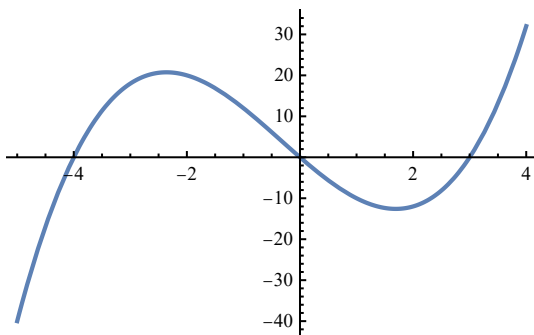
In[236]:=

```
f[x_] := x3 + x2 - 12 x
```

In[237]:=

```
Plot[f[x], {x, -5, 4}, ImageSize -> 250]
```

Out[237]=



In[238]:=

```
FindMinimum[f[x], {x, 2}]
```

Out[238]=

```
{-12.5972, {x -> 1.69425}}
```

A teraz poszukajmy maksimum lokalne tej samej funkcji.

In[239]:=

```
FindMaximum[f[x], {x, -2}]
```

Out[239]=

```
{20.7453, {x -> -2.36092}}
```

Przykład funkcji dwóch zmiennych.

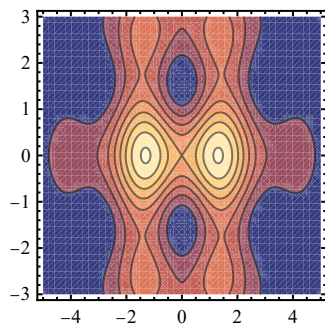
In[240]:=

```
h[x_, y_] := Exp[-(x2 + y2)/10] * ((Sin[x])2 + (Cos[y])2)
```

In[241]:=

```
ContourPlot[h[x, y], {x, -5, 5}, {y, -3, 3}, PlotPoints -> 50, ImageSize -> 150]
```

Out[241]=



In[242]:=

```
FindMinimum[h[x, y], {{x, 1}, {y, 1}}]
```

Out[242]=

```
{6.59629 × 10-21, {x -> 3.83718 × 10-11, y -> 1.5708}}
```

In[243]:=

```
FindMaximum[h[x, y], {{x, 1}, {y, 1}}]
```

Out[243]=

```
{1.62863, {x -> 1.3063, y -> 1.35438 × 10-9}}
```

Możemy poszukiwać także minimum i maksimum globalne funkcji w zadanym obszarze lub w całej dziedzinie.

Minimize[f, x]	poszukuje minimum globalnego funkcji jednej zmiennej;
Minimize[f, {x, y, ...}]	poszukuje minimum globalnego funkcji wielu zmiennych;
Minimize[{f, ograniczenia}, x]	poszukuje minimum globalnego funkcji jednej zmiennej, w obszarze określonym przez podane ograniczenia;
Minimize[{f, ograniczenia}, {x, y, ...}]	poszukuje minimum globalnego funkcji wielu zmiennych, w obszarze określonym przez podane ograniczenia.

Poszukajmy minimum globalnego funkcji kwadratowej.

In[244]:=

```
Minimize[x2 + 3 x - 5, x]
```

Out[244]=

```
{-29/4, {x -> -3/2}}
```

W przypadku, gdy funkcja nie posiada maksimum (lub minimum) globalnego otrzymamy odpowiedni komunikat.

In[245]:=

```
Maximize[x2 + 3 x - 5, x]
```

☞ **Maximize**: The maximum is not attained at any point satisfying the given constraints.

Out[245]=

```
{∞, {x -> -∞}}
```

Poszukiwania ograniczmy do obszaru $|x| \leq 2$.

In[246]:=

```
Maximize[{x^2 + 3 x - 5, Abs[x] ≤ 2}, x]
```

Out[246]=

```
{5, {x → 2}}
```

Kolejne przykłady.

In[247]:=

```
Minimize[{2 x + 3 y - 4 z, y - z ≥ 4 && x + z ≤ 5 && x ≥ 0 && y ≥ 0 && z ≥ 0}, {x, y, z}]
```

Out[247]=

```
{7, {x → 0, y → 9, z → 5}}
```

In[248]:=

```
Minimize[{2 x + 3 y - 4 z, y - z ≥ 4, x + z ≤ 5, x ≥ 0, y ≥ 0, z ≥ 0}, {x, y, z}]
```

Out[248]=

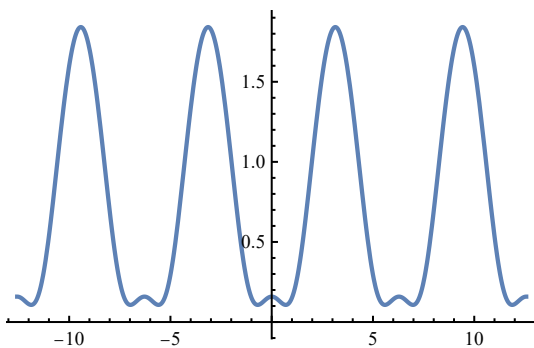
```
{7, {x → 0, y → 9, z → 5}}
```

Istnieją także instrukcje `NMinimize` i `NMaximize`, które pozwalają wyznaczać numerycznie ekstrema globalne.

In[249]:=

```
Plot[Cos[Sin[x]] - Sin[Cos[x]], {x, -4 π, 4 π}, ImageSize → 250]
```

Out[249]=



In[250]:=

```
NMinimize[Cos[Sin[x]] - Sin[Cos[x]], x]
```

Out[250]=

```
{0.107127, {x → 0.692729}}
```

In[251]:=

```
NMinimize[{Cos[Sin[x]] - Sin[Cos[x]], x < 0}, x]
```

Out[251]=

```
{0.107127, {x → -0.692729}}
```

In[252]:=

```
NMaximize[{Cos[Sin[x]] - Sin[Cos[x]], 8 < x < 12}, x]
```

Out[252]=

```
{1.84147, {x → 9.42478}}
```

Poszukiwanie ekstremów globalnych funkcji liniowej, w przypadku gdy ograniczenia są określone także przez funkcje liniowe jest nazywane programowaniem liniowym.

```
LinearProgramming[c, m, b]   poszukuje wektora x, który minimalizuje iloczyn c x
                             oraz spełnia nierówności  $m x \geq b$  i  $x \geq 0$ ;
LinearProgramming[c, m, b, p] poszukuje wektora x, który minimalizuje iloczyn c x
                             oraz spełnia nierówności  $m x \geq b$  i  $x \geq p$ .
```

Powyższa instrukcja jest formą macierzową instrukcji programowania liniowego. Każde zadanie programowania liniowego, zapisane za pomocą funkcji i układu nierówności, możemy przedstawić w postaci macierzowej. Rozwińmy przykład programowania liniowego. W tym celu zdefiniujemy najpierw potrzebne macierze.

```
In[253]:=
c = {2, 3, -4}; m = {{0, 1, -1}, {-1, 0, -1}}; b = {4, -5};
```

```
In[254]:=
xr = LinearProgramming[c, m, b]
```

```
Out[254]=
{0, 9, 5}
```

```
In[255]:=
c.xr
```

```
Out[255]=
7
```

```
In[256]:=
xr1 = LinearProgramming[c, m, b, {1, 1, 1}]
```

```
Out[256]=
{1, 8, 4}
```

```
In[257]:=
c.xr1
```

```
Out[257]=
10
```

5.6. Aproksymacja i interpolacja

```
In[258]:=
Clear["Global`*"]
```

Zadanie znalezienia kombinacji funkcji aproksymujących dane liczbowe możemy rozwiązać za pomocą instrukcji Fit.

```
Fit[dane, funkcje, x]   aproksymuje podane dane przy użyciu
                         kombinacji liniowej funkcji zmiennej x.
```

Utwórzmy listę wartości.

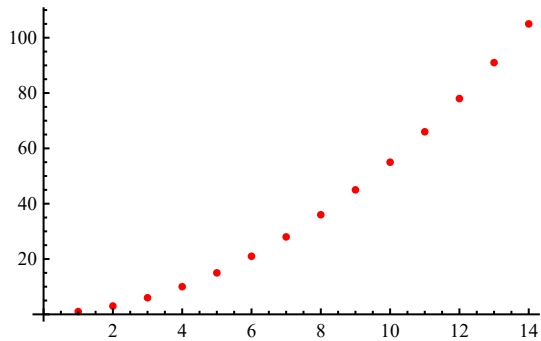
```
In[259]:=
dane = Table[Binomial[n, 2], {n, 2, 15}]
```

```
Out[259]=
{1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105}
```

In[260]:=

```
wykres = ListPlot [ dane, PlotStyle → { PointSize [ 0.015 ], Red }, ImageSize → 250 ]
```

Out[260]=



Aproksymujmy nasze dane funkcją liniową.

In[261]:=

```
q = Fit [ dane, { 1, x }, x ]
```

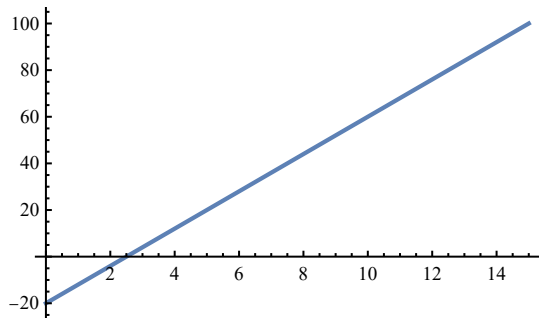
Out[261]=

```
- 20. + 8. x
```

In[262]:=

```
apro = Plot [ q, { x, 0, 15 }, ImageSize → 250 ]
```

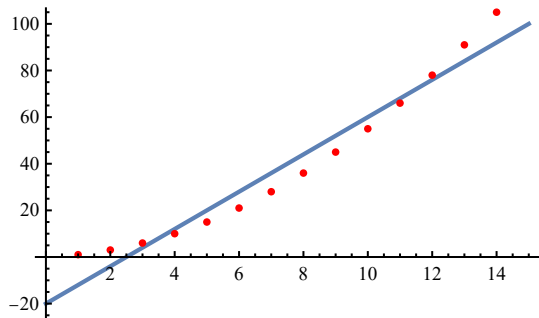
Out[262]=



In[263]:=

```
Show [ apro, wykres, ImageSize → 250 ]
```

Out[263]=



A teraz wielomianem trzeciego stopnia.

In[264]:=

```
aap = Fit[dane, {1, x, x^2, x^3}, x]
```

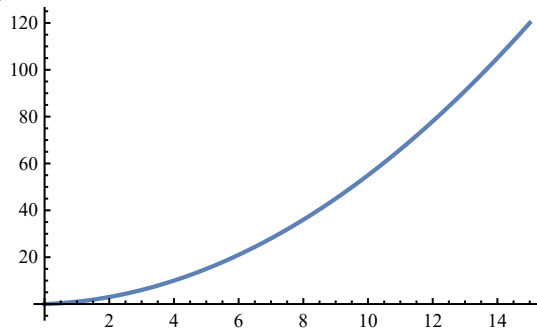
Out[264]=

$$-2.9314 \times 10^{-14} + 0.5 x + 0.5 x^2 + 9.41736 \times 10^{-17} x^3$$

In[265]:=

```
apro2 = Plot[aap, {x, 0, 15}, ImageSize -> 250]
```

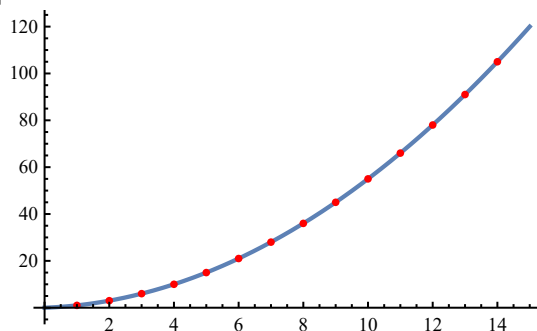
Out[265]=



In[266]:=

```
Show[apro2, wykres, ImageSize -> 250]
```

Out[266]=



Kolejny przykład.

In[267]:=

```
dane = Table[{x, N[ $\frac{3 + \text{Cos}[x]}{\text{Cos}[x]}$ ]}, {x, 0, 1.5, 0.1}]
```

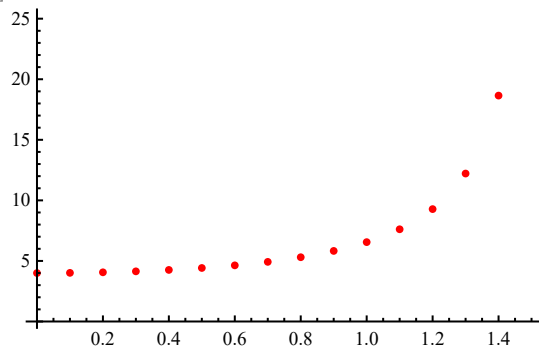
Out[267]=

```
{ {0., 4.}, {0.1, 4.01506}, {0.2, 4.06102}, {0.3, 4.14025}, {0.4, 4.25711}, {0.5, 4.41848},
  {0.6, 4.63488}, {0.7, 4.92238}, {0.8, 5.30597}, {0.9, 5.82618}, {1., 6.55245},
  {1.1, 7.61381}, {1.2, 9.27911}, {1.3, 12.215}, {1.4, 18.6505}, {1.5, 43.4105} }
```

In[268]:=

```
p1 = ListPlot [ dane, PlotStyle → { PointSize [ 0.015 ], Red }, ImageSize → 250 ]
```

Out[268]=



In[269]:=

```
a1 = Fit [ dane, { 1,  $\frac{1}{\text{Cos}[x]}$  }, x ]
```

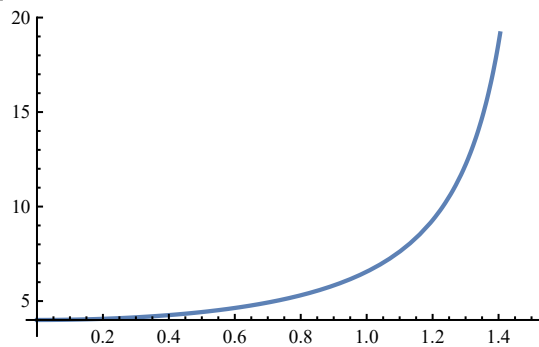
Out[269]=

```
1. + 3. Sec [ x ]
```

In[270]:=

```
p2 = Plot [ a1, { x, 0, 1.5 }, ImageSize → 250 ]
```

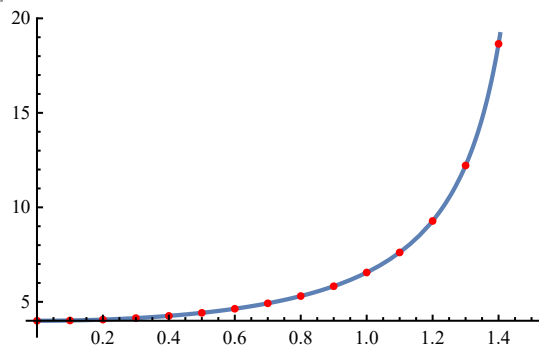
Out[270]=



In[271]:=

```
Show [ p2, p1, ImageSize → 250 ]
```

Out[271]=



In[272]:=

```
a2 = Fit[dane, {1,  $\frac{1}{\text{Cos}[x]}$ , Sin[x], Exp[-x]}, x]
```

Out[272]=

```
1. - 9.83344 × 10-14 e-x + 3. Sec[x] - 5.88808 × 10-14 Sin[x]
```

In[273]:=

```
Chop [%]
```

Out[273]=

```
1. + 3. Sec[x]
```

Aproksymacja funkcją dwóch zmiennych.

In[274]:=

```
dane = Flatten[Table[{x, y, -3 y + 2 x * y - x2}, {x, 3}, {y, 3}], 1]
```

Out[274]=

```
{ {1, 1, -2}, {1, 2, -3}, {1, 3, -4}, {2, 1, -3},  
  {2, 2, -2}, {2, 3, -1}, {3, 1, -6}, {3, 2, -3}, {3, 3, 0} }
```

In[275]:=

```
aw = Fit[dane, {1, x, y, x * y, x2, y2}, {x, y}]
```

Out[275]=

```
-1.67453 × 10-15 - 1.1973 × 10-15 x - 1. x2 - 3. y + 2. x y - 2.93012 × 10-16 y2
```

In[276]:=

```
Chop [aw]
```

Out[276]=

```
-1. x2 - 3. y + 2. x y
```

Dostępna jest także instrukcja pozwalająca dokonywać aproksymacji funkcją zależną w sposób nieliniowy od dobieranych parametrów.

```
FindFit[dane, funkcja, {p1, p2, ...}, x]      jednowymiarowa aproksymacja nieliniowa;  
FindFit[dane, funkcja, {p1, p2, ...}, {x, y, ...}] wielowymiarowa aproksymacja nieliniowa.
```

Zobaczmy przykład.

In[277]:=

```
dane = Table[{x, 2.1 * Exp[-(x - 0.91)2]}, {x, 0., 3, 0.2}]
```

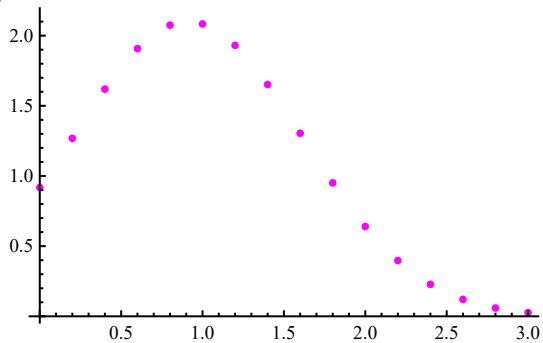
Out[277]=

```
{ {0., 0.917445}, {0.2, 1.2685}, {0.4, 1.61905}, {0.6, 1.90758}, {0.8, 2.07474}, {1., 2.08306},  
  {1.2, 1.93061}, {1.4, 1.65175}, {1.6, 1.30452}, {1.8, 0.951075}, {2., 0.64008},  
  {2.2, 0.397658}, {2.4, 0.228056}, {2.6, 0.120734}, {2.8, 0.0590033}, {3., 0.0266181} }
```

In[278]:=

```
p1 = ListPlot [ dane, PlotStyle → { PointSize [ 0.015 ], Magenta }, ImageSize → 250 ]
```

Out[278]=



In[279]:=

```
Clear [ a, b, x ] ;
```

In[280]:=

```
funkcja = a * Exp [ - ( b - x ) ^ 2 ]
```

Out[280]=

$$a e^{-(b-x)^2}$$

In[281]:=

```
f = FindFit [ dane, funkcja, { a, b }, x ]
```

Out[281]=

```
{ a → 2.1, b → 0.91 }
```

In[282]:=

```
f1 = funkcja / . f
```

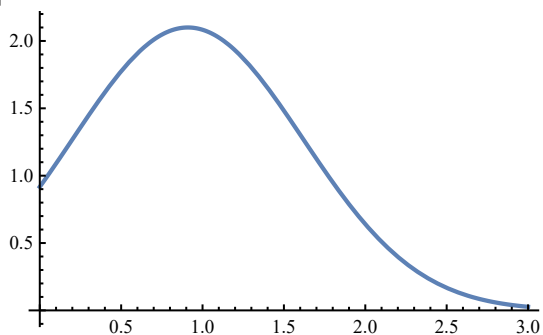
Out[282]=

$$2.1 e^{-(0.91-x)^2}$$

In[283]:=

```
p2 = Plot [ f1, { x, 0, 3 }, ImageSize → 250 ]
```

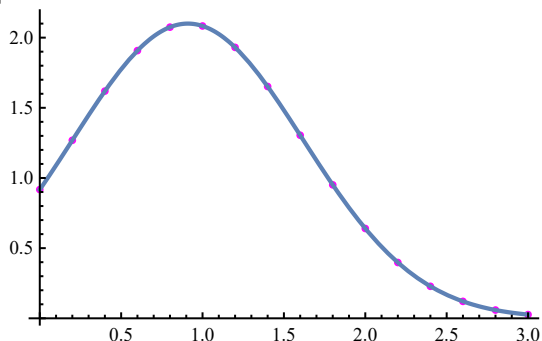
Out[283]=



In[284]:=

Show [p1, p2, ImageSize → 250]

Out[284]=



Mathematica posiada instrukcje służące do poszukiwania wielomianów, które interpolują (w zadanych punktach przyjmują dokładnie takie same wartości) nasze dane.

<code>InterpolatingPolynomial[{f1, f2, ..., fn}, x]</code>	znajduje wielomian zmiennej x , który interpoluje wartości (i, f_i) ;
<code>InterpolatingPolynomial[{{x1, f1}, {x2, f2}, ...}, x]</code>	znajduje wielomian zmiennej x , który interpoluje wartości (x_i, f_i) .

Dla interpolacji wielomianowej n punktów będziemy potrzebować wielomianu stopnia nie większego niż $n-1$. Zadajmy kilka punktów i znajźmy dla nich wielomian interpolacyjny.

In[285]:=

wiel = InterpolatingPolynomial [{ 0, 5, 7, 31, 68, 90 }, x]

Out[285]=

$$\left(5 + \left(-\frac{3}{2} + \left(\frac{25}{6} + \left(-\frac{17}{12} + \frac{1}{8} (-5 + x) \right) (-4 + x) \right) (-3 + x) \right) (-2 + x) \right) (-1 + x)$$

In[286]:=

Expand [wiel]

Out[286]=

$$-82 + \frac{1925x}{12} - \frac{2501x^2}{24} + \frac{695x^3}{24} - \frac{79x^4}{24} + \frac{x^5}{8}$$

In[287]:=

wiel /. x → 2

Out[287]=

5

In[288]:=

wiel /. x → 6

Out[288]=

90

Jeszcze jeden przykład.

```
In[289]:= InterpolatingPolynomial[{{0, 0}, {1, 5}, {2, 7}, {5, 31}, {6, 68}, {10, 90}}, x] // N // Expand
```

```
Out[289]= 4.3375 x + 2.82389 x2 - 2.78299 x3 + 0.662361 x4 - 0.0407639 x5
```

5.7. Funkcje zespolone

```
In[290]:= Clear["Global`*"]
```

Istnieje możliwość wykorzystywania liczb zespolonych. Symbolem jednostki urojonej, w programie *Mathematica*, jest I lub i . Symbol i uzyskamy pisząc `ESCiiESC` lub korzystając z palety *Basic Math Input*. Oprócz samej definicji liczby zespolonej mamy do dyspozycji wszystkie podstawowe funkcje i operacje na liczbach zespolonych (część rzeczywista i urojona, moduł, sprzężenie itd.). Poza tym wszystkie funkcje matematyczne, które są określone w dziedzinie zespolonej, możemy wykorzystywać bez żadnych zmian czy dodatkowych opcji. Wystarczy tylko jako ich argument podać liczbę zespoloną.

<code>x + I y</code>	liczba zespolona $x + iy$;
<code>Re[z]</code>	część rzeczywista liczby z ;
<code>Im[z]</code>	część urojona liczby z ;
<code>Conjugate[z]</code>	liczba sprzężona;
<code>Abs[z]</code>	moduł liczby z ;
<code>Arg[z]</code>	argument liczby z .

Kilka przykładów.

```
In[291]:= Exp[2 + 3 I]
```

```
Out[291]= e2+3 i
```

```
In[292]:= Exp[2 + 3 I] // N
```

```
Out[292]= -7.31511 + 1.04274 i
```

```
In[293]:= Sqrt[-1]
```

```
Out[293]= i
```

```
In[294]:= (1 - 5) / (1 + 1)
```

```
Out[294]= -2 + 3 i
```

In[295]:= **Re [2 - I]**

Out[295]=
2

In[296]:= **Conjugate [2 - I]**

Out[296]=
 $2 + i$

In[297]:= **Abs [2 - I]**

Out[297]=
 $\sqrt{5}$

In[298]:= **Arg [2 - I]**

Out[298]=
 $-\text{ArcTan}\left[\frac{1}{2}\right]$

In[299]:= **Arg [1 + I]**

Out[299]=
 $\frac{\pi}{4}$

Mathematica potrafi znaleźć residuum, gdy może rozwinąć podaną funkcję w szereg potęgowy w pewnym otoczeniu punktu z_0 .

`Residue[funkcja, {z, z0}]` oblicza residuum funkcji w punkcie z_0 .

Przeliczmy kilka przykładów.

In[300]:= **Residue** $\left[\frac{z^2 + 1}{z - 2}, \{z, 2\}\right]$

Out[300]=
5

In[301]:= **Residue** $\left[z + \frac{1}{z}, \{z, 0\}\right]$

Out[301]=
1

In[302]:=

$$\text{Residue}\left[\frac{\text{Exp}[z] + 1}{z - i}, \{z, i\}\right]$$

Out[302]=

$$1 + e^i$$

5.8. Przekształcenie Laplace'a

In[303]:=

```
Clear["Global`*"]
```

Omówimy teraz instrukcje realizujące przekształcenie Laplace'a.

LaplaceTransform[wrażenie, t, s]	znajduje transformatę Laplace'a;
LaplaceTransform[wrażenie, {t1, t2, ...}, {s1, s2, ...}]	znajduje wielowymiarową transformatę Laplace'a;
InverseLaplaceTransform[wrażenie, s, t]	znajduje odwrotną transformatę Laplace'a;
InverseLaplaceTransform[wrażenie, {s1, s2, ...}, {t1, t2, ...}]	znajduje wielowymiarową odwrotną transformatę Laplace'a.

Przykłady.

In[304]:=

```
LaplaceTransform[t^2 * Sin[t], t, s]
```

Out[304]=

$$\frac{2(-1 + 3s^2)}{(1 + s^2)^3}$$

In[305]:=

```
InverseLaplaceTransform[2/(1 - s), s, t]
```

Out[305]=

$$-2e^t$$

In[306]:=

```
LaplaceTransform[Exp[x], x, s]
```

Out[306]=

$$\frac{1}{-1 + s}$$

In[307]:=

```
InverseLaplaceTransform[%, s, x]
```

Out[307]=

$$e^x$$

6. Równania różniczkowe

6.1. Instrukcja DSolve

In[308]:=

```
Clear["Global`*"]
```

Do rozwiązywania równań różniczkowych zwyczajnych możemy wykorzystać instrukcję DSolve.

<code>DSolve[równanie, y[x], x]</code>	podaje rozwiązanie równania różniczkowego ze względu na $y[x]$, przyjmując x jako zmienną niezależną;
<code>DSolve[równanie, y, x]</code>	rozwiązanie jest podawane w postaci funkcji prymitywnej;
<code>DSolve[równania, {y1, y2, ...}, x]</code>	podaje rozwiązanie układu równań różniczkowych.

Musimy wyraźnie zaznaczyć, która zmienna jest zależną, a która niezależną. Jeśli y jest zależne od x , to równanie piszemy w postaci: $y''[x] + y'[x] = y[x]$, nie wystarczy napisać tylko $y'' + y' = y$. Pierwsze dwie przedstawione powyżej postaci instrukcji DSolve różnią się zapisem policzonego wyniku. Pierwsza podaje podstawienie za symbol $y[x]$ ($y[x] \rightarrow \dots$) o jawnie określonym argumencie, natomiast druga podaje podstawienie za symbol y ($y \rightarrow \dots$), któremu jest przyporządkowana funkcja prymitywna postaci `Function[{argument},wzór]` lub w drugiej dopuszczalnej formie, w której argument jest oznaczany symbolem $\#$, a samą funkcję kończy symbol $\&$. Drugi sposób jest wygodny, gdy wynik chcemy podstawiać do wyrażeń zawierających nie tylko funkcję $y[x]$, ale także jej pochodne, wartości dla konkretnego argumentu, itp. Zobaczmy na przykładzie różnicy pomiędzy obu postaciami omawianej instrukcji.

Rozwiążmy równanie różniczkowe $y'(x) = 5y(x)$ (c_1 jest stałą, którą możemy wyznaczyć z warunku początkowego).

In[309]:=

```
DSolve[y'[x] == 5 * y[x], y[x], x]
```

Out[309]=

```
{{y[x] -> e^{5x} c_1}}
```

Jeśli podamy warunek początkowy, to w rozwiązaniu nie będzie nieokreślonej stałej c_1 .

In[310]:=

```
r1 = DSolve[{y'[x] == 5 * y[x], y[0] == 1}, y[x], x]
```

Out[310]=

```
{{y[x] -> e^{5x}}}
```

Rozwiążmy to samo równanie przy pomocy drugiej wersji instrukcji DSolve.

In[311]:=

```
r2 = DSolve[{y'[x] == 5 * y[x], y[0] == 1}, y, x]
```

Out[311]=

```
{{y -> Function[{x}, e^{5x}]}}
```

Podstawmy otrzymane rozwiązania do nowego wyrażenia.

In[312]:=

```
y''[x] - 5 y[x] - y[0] /. r1[[1]]
```

Out[312]=

```
-5 e^{5x} - y[0] + y''[x]
```

In[313]:=

`y''[x] - 5 y[x] - y[0] /. r2[[1]]`

Out[313]=

 $-1 + 20 e^{5x}$

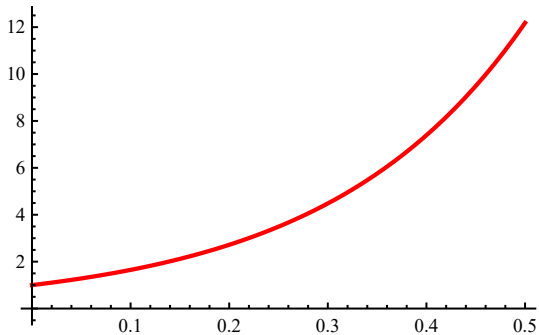
Zgodnie z wcześniejszym opisem, w pierwszym przypadku symbole $y''[x]$ i $y[0]$ nie uległy zmianie.

Znalezione rozwiązanie równania różniczkowego możemy wykreślić (w obu przypadkach postępujemy tak samo).

In[314]:=

`Plot[y[x] /. r1[[1]], {x, 0, 0.5}, PlotStyle -> {Red}, ImageSize -> 250]`

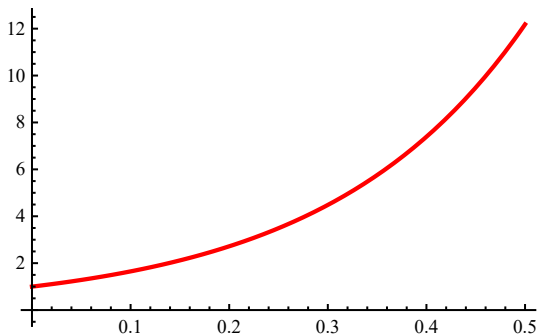
Out[314]=



In[315]:=

`Plot[r1[[1, 1, 2]], {x, 0, 0.5}, PlotStyle -> {Red}, ImageSize -> 250]`

Out[315]=



5.2. Instrukcja NDSolve

In[316]:=

`Clear["Global`*"]`

Wiemy już, jak symbolicznie rozwiązać równanie różniczkowe. Poznamy obecnie metodę uzyskania numerycznej wartości takiego rozwiązania.

<code>NDSolve[równanie, y, {x, x_{min}, x_{max}}]</code>	numeryczne rozwiązanie równania różniczkowego ze względu na y , ze zmienną niezależną x z przedziału $[x_{\min}, x_{\max}]$;
<code>NDSolve[równania, {y₁, y₂, ...}, {x, x_{min}, x_{max}}]</code>	numeryczne rozwiązanie układu równań różniczkowych ze względu na y_i , ze zmienną niezależną x z przedziału $[x_{\min}, x_{\max}]$.

Instrukcja `NDSolve` obliczone przybliżone rozwiązanie równania różniczkowego przedstawia przy pomocy funkcji aproksymacji (jako obiekt typu `InterpolatingFunction`). Znajdźmy liczbową wartość rozwiązania równania

$y'(x) = \sin(y(x))$, z warunkiem początkowym $y(0) = 1$, w punkcie $x = 2$. Zadanie to rozwiążemy w dwóch krokach. W pierwszym znajdziemy funkcję aproksymacji, natomiast w drugim, wartość tej funkcji w interesującym nas punkcie.

In[317]:=

```
roz1 = NDSolve [ {y' [x] == Sin [y [x] ], y [0] == 1 }, y, {x, 0, 4} ]
```

Out[317]=

```
{ {y → InterpolatingFunction[ +  Domain: {{0., 4.}} Output: scalar ] ] }
```

In[318]:=

```
y [2] /.roz1[[1]]
```

Out[318]=

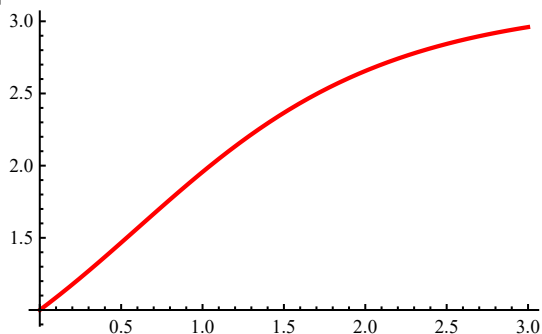
```
2.65591
```

Wykreślmy otrzymane rozwiązanie.

In[319]:=

```
Plot [ Evaluate [y [x] /.roz1 ], {x, 0, 3}, PlotStyle → {Red}, ImageSize → 250 ]
```

Out[319]=



Powyżej widzieliśmy, w jaki sposób odczytywać wartości liczbowe rozwiązania oraz jak rozwiązanie przedstawić graficznie.



<pre>y[x] /. rozwiązanie</pre>	<p>wykorzystanie otrzymanych wzorów na y, dla obliczenia wartości liczbowej y[x];</p>
<pre>Plot[Evaluate[y[x] /. rozwiązanie], {x, x_min, x_max}]</pre>	<p>sporządzenie wykresu rozwiązania równania różniczkowego.</p>

Podobnie postępujemy w przypadku układu równań różniczkowych.

In[320]:=

```
roz2 = NDSolve [ {y' [x] == -3 z [x], z' [x] == y [x] + 4, y [0] == 1, z [0] == 0 }, {y, z}, {x, 0, 10} ]
```

Out[320]=

```
{ {y → InterpolatingFunction[ +  Domain: {{0., 10.}} Output: scalar ] },  
  {z → InterpolatingFunction[ +  Domain: {{0., 10.}} Output: scalar ] ] }
```

Obliczmy wartość znalezionych funkcji w punkcie 1.

In[321]:=

`{y[5], z[5]} /. roz2[[1]]`

Out[321]=

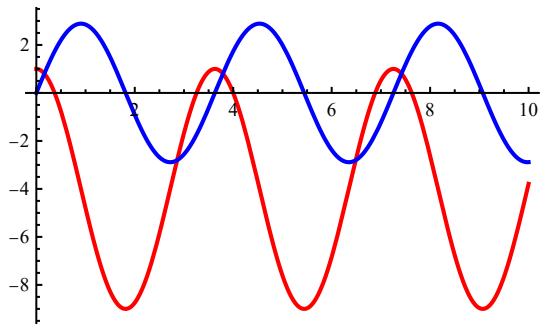
`{-7.60856, 1.99819}`

Wykreślmy także otrzymane rozwiązania (funkcję y wykreślmy kolorem czerwonym, a funkcję z kolorem niebieskim).

In[322]:=

```
Plot[Evaluate[{y[x], z[x]} /. roz2], {x, 0, 10},  
PlotStyle -> {{Thickness[0.008], Red}, {Thickness[0.008], Blue}}, ImageSize -> 250]
```

Out[322]=

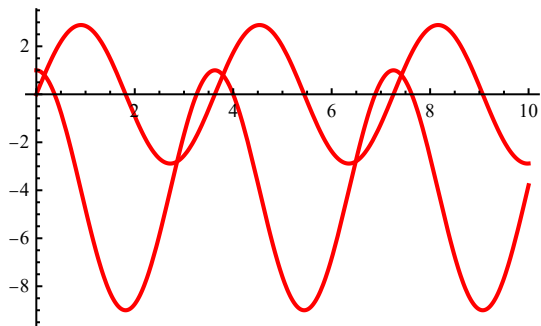


Zobaczmy efekt bez zastosowania instrukcji Evaluate.

In[323]:=

```
Plot[{y[x], z[x]} /. roz2, {x, 0, 10},  
PlotStyle -> {{Thickness[0.008], Red}, {Thickness[0.008], Blue}}, ImageSize -> 250]
```

Out[323]=

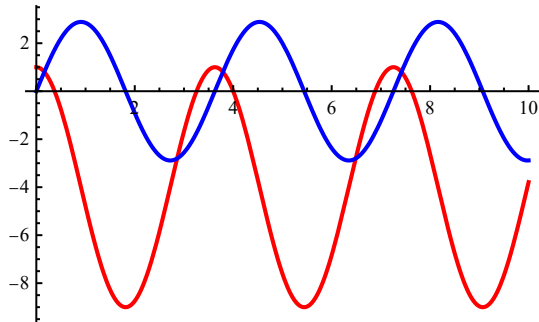


W tym przypadku *Mathematica* potraktowała wprowadzone wyrażenie jako jedną funkcję. Dlatego też zastosowała instrukcje tylko z pierwszego nawiasu. Żeby wyrażenie zostało potraktowane jako dwie funkcje, musimy wykonać osobne podstawienia za funkcje y oraz z .

In[324]:=

```
Plot[ {y[x] /. roz2, z[x] /. roz2}, {x, 0, 10},
      PlotStyle -> { {Thickness [0.008], Red}, {Thickness [0.008], Blue} }, ImageSize -> 250 ]
```

Out[324]=



6.3. Przykłady zastosowań równań różniczkowych zwyczajnych

In[325]:=

```
Clear["Global`*"]
```

6.3.1. Stygnięcie ciała

Rozpatrzmy zagadnienie, w którym ciało o temperaturze T_p zostaje umieszczone w otoczeniu, którego temperatura T_o jest stała i niższa od temperatury ciała. Zaczynając od tego momentu ciało stygnie. W przedziale $(t, t + \Delta t)$ dodatniemu przyrostowi czasu (Δt) odpowiada ujemny przyrost temperatury ciała (ΔT) . Stosunek tych dwóch przyrostów $(\frac{\Delta T}{\Delta t})$ jest średnią prędkością zmiany temperatury w rozważanym przedziale czasu, natomiast granica tego ilorazu, przy $\Delta t \rightarrow 0$, jest prędkością stygnięcia w chwili t . Zgodnie z prawem Newtona prędkość ta jest proporcjonalna do różnicy temperatury ciała i otoczenia:

$$\frac{dT}{d\tau} = -\lambda (T - T_o),$$

gdzie $\lambda > 0$. Jeśli równanie uzupełnimy warunkiem początkowym $T(0) = T_p$, to otrzymamy zagadnienie początkowe określające przebieg stygnięcia ciała.

Rozwiążmy powyższe zagadnienie początkowe.

In[326]:=

```
roz = DSolve[ {T'[t] == -lambda * (T[t] - To), T[0] == Tp}, T[t], t]
```

Out[326]=

```
{{T[t] -> e^{-t lambda} (-To + e^{t lambda} To + Tp)}}
```

Zdefiniujmy funkcję będącą rozwiązaniem omawianego zagadnienia.

In[327]:=

```
temp[t_, lambda_, To_, Tp_] = roz[[1, 1, 2]]
```

Out[327]=

```
e^{-t lambda} (-To + e^{t lambda} To + Tp)
```

W funkcji tej oprócz czasu, temperatury otoczenia oraz temperatury początkowej ciała występuje jeszcze stała proporcjonalności λ . Wyliczmy ją dla przypadku, gdy ciało o temperaturze 100°C umieszczono w środowisku, którego temperatura wynosi 20°C , przy założeniu, że po minucie temperatura ciała spadła o 10 stopni.

```
In[328]:=
w = FindRoot [ temp [ 60, λ, 20, 100 ] == 90, { λ, 0 } ]
```

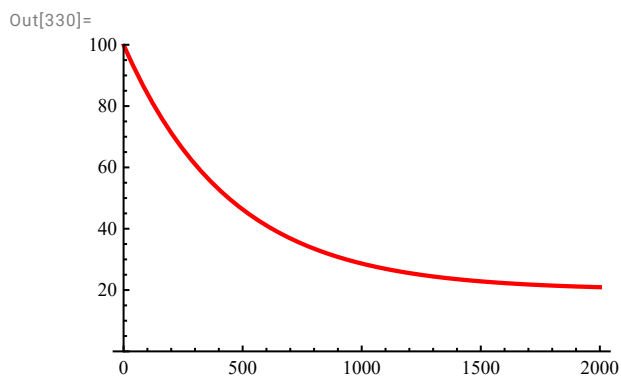
```
Out[328]=
{ λ → 0.00222552 }
```

```
In[329]:=
λ1 = w[[1, 2]]
```

```
Out[329]=
0.00222552
```

Wykreślmy teraz krzywą będącą rozwiązaniem omawianego zagadnienia.

```
In[330]:=
Plot [ temp [ t, λ1, 20, 100 ], { t, 0, 2000 }, PlotRange → { 0, 100 },
PlotStyle → { RGBColor [ 1, 0, 0 ] }, ImageSize → 250 ]
```



Widzimy, że temperatura ciała dąży do temperatury otoczenia. Sprawdźmy po jakim czasie ciało będzie miało temperaturę 30 °C.

```
In[331]:=
r1 = FindRoot [ temp [ t, λ1, 20, 100 ] == 30, { t, 500 } ]
```

```
Out[331]=
{ t → 934.361 }
```

Zamieńmy sekundy na minuty.

```
In[332]:=
r1[[1, 2]]
60
```

```
Out[332]=
15.5727
```

Sprawdźmy także, ile czasu potrzeba, aby ciało ostygło do 21 °C.

```
In[333]:=
r2 = FindRoot [ temp [ t, λ1, 20, 100 ] == 21, { t, 500 } ]
```

```
Out[333]=
{ t → 1968.99 }
```

In[334]:=

$$\frac{r2[[1, 2]]}{60}$$

Out[334]=

32.8165

6.3.2. Drgania mechaniczne tłumione

Na ciało o masie m działa siła sprężystości S , która jest proporcjonalna do wychylenia ciała z położenia równowagi, ale jej zwrot jest przeciwny do zwrotu tego wychylenia:

$$S = -s x,$$

gdzie s jest dodatnią stałą proporcjonalności. Na ciało działa również siła tłumiąca T proporcjonalna do prędkości ciała, ale skierowana przeciwnie do niej:

$$T = -k \frac{dx}{dt},$$

gdzie k jest stałą dodatnią. Wykorzystując prawo Newtona, możemy napisać:

$$m \frac{d^2 x}{dt^2} = T + S.$$

Wstawiając do ostatniego wzoru poprzednie zależności, otrzymamy równanie prostoliniowego ruchu ciała o masie m , poruszającego się pod działaniem sił S i T :

$$m \frac{d^2 x}{dt^2} + k \frac{dx}{dt} + s x = 0.$$

Otrzymaliśmy więc równanie różniczkowe drugiego rzędu jednorodne liniowe o stałych współczynnikach.

Napišemy funkcję, która będzie znajdować rozwiązanie równania (t jest symbolem zmiennej niezależnej, jaka zostanie wykorzystana w zapisie rozwiązania).

In[335]:=

```
drmania [m_, k_, s_, t_Symbol] := Module[{wynik},
  wynik = DSolve[m x''[t] + k x'[t] + s x[t] == 0, x[t], t];
  Return[wynik[[1, 1, 2]]]
```

Poszukajmy rozwiązania dla wybranych wartości stałych m , k i s .

In[336]:=

```
drmania [1, 2, 5, t]
```

Out[336]=

$$e^{-t} c_2 \text{Cos}[2 t] + e^{-t} c_1 \text{Sin}[2 t]$$

Uzyskane w rozwiązaniu stałe c_1 i c_2 możemy wyliczyć, podając położenie oraz prędkość ciała w chwili początkowej.

Napišemy drugą funkcję uwzględniającą warunki początkowe.

In[337]:=

```
drmania2 [m_, k_, s_, x0_, v0_, t_Symbol] := Module[{wynik},
  wynik = DSolve[{m x''[t] + k x'[t] + s x[t] == 0, x[0] == x0, x'[0] == v0}, x[t], t];
  Return[wynik[[1, 1, 2]]]
```

Znajdźmy równanie drogi ciała dla takich samych jak poprzednio stałych i przy założeniu, że w chwili początkowej

znajduje się w punkcie równowagi, a jego prędkość jest równa $1 \frac{m}{s}$.

In[338]:=

```
p1 = drgania2[1, 2, 5, 0, 1, t]
```

Out[338]=

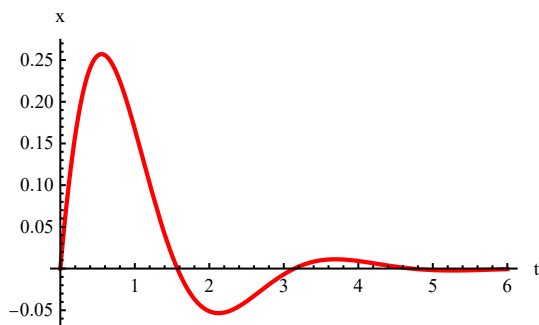
$$\frac{1}{2} e^{-t} \sin[2t]$$

Wykreślmy otrzymane rozwiązanie.

In[339]:=

```
Plot[p1, {t, 0, 6}, PlotStyle -> {RGBColor[1, 0, 0]},
      PlotRange -> All, AxesLabel -> {"t", "x"}, ImageSize -> 250]
```

Out[339]=



6.4. Wykorzystanie przekształcenia Laplace'a

In[340]:=

```
Clear["Global`*"]
```

Omówimy teraz wykorzystanie przekształcenia Laplace'a do rozwiązywania równań różniczkowych zwyczajnych. Stosując tę metodę, najpierw zamieniamy przy pomocy przekształcenia Laplace'a równanie różniczkowe na równanie algebraiczne, które jako niewiadomą zawiera transformatę Laplace'a szukanej funkcji. Rozwiązując następnie uzyskane równanie algebraiczne, znajdujemy transformatę Laplace'a szukanej funkcji. Na koniec przy pomocy odwrotnej transformaty Laplace'a znajdujemy szukaną funkcję, będącą rozwiązaniem wyjściowego równania różniczkowego.

6.4.1. Przykład 1

Na początku zdefiniujemy rozwiązywane równanie.

In[341]:=

```
rownanie = D[y[t], {t, 2}] + 2 * y[t] == Cos[t]
```

Out[341]=

$$2y[t] + y''[t] == \cos[t]$$

Zastosujmy przekształcenie Laplace'a do naszego równania.

In[342]:=

```
tl = LaplaceTransform[rownanie, t, s]
```

Out[342]=

$$2 \text{LaplaceTransform}[y[t], t, s] + s^2 \text{LaplaceTransform}[y[t], t, s] - s y[0] - y'[0] == \frac{s}{1 + s^2}$$

Rozwiążmy otrzymane równanie algebraiczne ze względu na transformatę Laplace'a funkcji y .

In[343]:=

```
ra = Solve [tl, LaplaceTransform [y [t], t, s] ]
```

Out[343]=

$$\left\{ \left\{ \text{LaplaceTransform} [y [t], t, s] \rightarrow \frac{s + s y [0] + s^3 y [0] + y' [0] + s^2 y' [0]}{(1 + s^2) (2 + s^2)} \right\} \right\}$$

Na koniec wystarczy znaleźć odwrotną transformatę Laplace'a.

In[344]:=

```
roz = InverseLaplaceTransform [ra[[1,1,2]], s, t]
```

Out[344]=

$$\frac{1}{2} \left(2 \text{Cos} [t] - 2 \text{Cos} [\sqrt{2} t] + 2 \text{Cos} [\sqrt{2} t] y [0] + \sqrt{2} \text{Sin} [\sqrt{2} t] y' [0] \right)$$

Sprawdźmy poprawność wyniku.

In[345]:=

```
D [roz, {t, 2}] + 2 * roz == Cos [t]
```

Out[345]=

$$2 \text{Cos} [t] - 2 \text{Cos} [\sqrt{2} t] + 2 \text{Cos} [\sqrt{2} t] y [0] + \sqrt{2} \text{Sin} [\sqrt{2} t] y' [0] + \frac{1}{2} \left(-2 \text{Cos} [t] + 4 \text{Cos} [\sqrt{2} t] - 4 \text{Cos} [\sqrt{2} t] y [0] - 2 \sqrt{2} \text{Sin} [\sqrt{2} t] y' [0] \right) == \text{Cos} [t]$$

In[346]:=

```
Simplify [%]
```

Out[346]=

```
True
```

6.4.2. Przykład 2

Korzystając z przekształcenia Laplace'a, możemy także rozwiązywać układy równań różniczkowych zwyczajnych. Rozwiążemy układ równań różniczkowych postaci:

$$\begin{aligned} x'(t) &= x(t) + y(t) \\ y'(t) &= x(t) + y(t) + t \end{aligned}$$

z warunkiem $x(0) = 0$ i $y(0) = 0$.

Wprowadzimy krótkie oznaczenia na transformaty szukanych funkcji.

In[347]:=

```
x /: LaplaceTransform [x [t], t, s] = ltx;
```

```
y /: LaplaceTransform [y [t], t, s] = lty;
```

Policzmy transformaty obu równań rozwiązywanego układu.

In[349]:=

```
r1 = LaplaceTransform [x' [t] == x [t] + y [t], t, s] /. {x [0] → 0, y [0] → 0}
```

Out[349]=

```
ltx s == ltx + lty
```

In[350]:=

```
r2 = LaplaceTransform [y' [t] == x [t] + y [t] + t, t, s] /. {x [0] → 0, y [0] → 0}
```

Out[350]=

```
lty s == ltx + lty +  $\frac{1}{s^2}$ 
```

W celu znalezienia transformat szukanych funkcji musimy rozwiązać układ równań.

In[351]:=

```
roz = Solve [ {r1, r2}, {ltx, lty} ]
```

Out[351]=

```
 $\left\{ \left\{ ltx \rightarrow \frac{1}{(-2 + s) s^3}, lty \rightarrow -\frac{1 - s}{(-2 + s) s^3} \right\} \right\}$ 
```

Wyznaczając transformaty odwrotne, otrzymamy szukane funkcje.

In[352]:=

```
xf = InverseLaplaceTransform [roz[[1,1,2]], s, t]
```

Out[352]=

```
 $\frac{1}{8} (-1 + e^{2t} - 2t - 2t^2)$ 
```

In[353]:=

```
yf = InverseLaplaceTransform [roz[[1,2,2]], s, t]
```

Out[353]=

```
 $\frac{1}{8} (-1 + e^{2t} - 2t + 2t^2)$ 
```

Sprawdźmy poprawność wyniku.

In[354]:=

```
{D [xf, t] == xf + yf, D [yf, t] == xf + yf + t} // Simplify
```

Out[354]=

```
{True, True}
```

In[355]:=

```
{xf, yf} /. {t → 0}
```

Out[355]=

```
{0, 0}
```