Mathematica - podstawy

Artur Kalinowski

Semestr letni 2011/2012

Spis treści

1. Program Mathematica
2. Podstawowe operacje
3. Wykresy i macierze
4. Równania algebraiczne
5. Równania różniczkowe

Program Mathematica

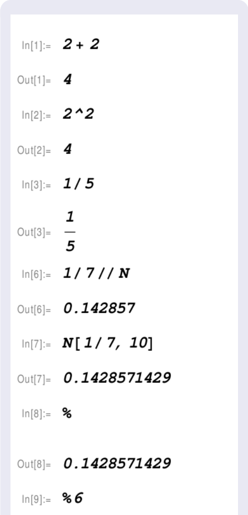
- Mathematica to jeden z najbardziej popularnych programów do wykonywania obliczeń symbolicznych i numerycznych
- Inne podobne programy to komercyjny MAPLE i darmowa MAXIMA
- Program uruchamiamy z Menu, lub z linii poleceń:

```
[akalinow@hpAK ~]$ mathematica
```

- W oknie powitalnym tworzymy nowy "Notebook", lub otwieramy już istniejący dokument
- Z menu "Help → Virtual book" otwieramy okno z dokumentacją

Podstawowe operacje matematyczne

- Wyrażenia obliczamy (ang. *evaluate*) wciskając **Shift+Enter** po wpisaniu wyrażenia
- Obliczanie wyrażenia anulujemy kombinacją **Alt+.**
- Poprzednie wyrażenie przywołujemy kombinacją **Ctrl+L**
- Przybliżoną wartość numeryczną uzyskujemy dodając **//N** na końcu, lub używając funkcji **N[wyrażenie, precyzja]**
- Wynik poprzedniego obliczenia przywołujemy używając **%**
- Wynik poprzedniego obliczenia o numerze X przywołujemy używając **%X**



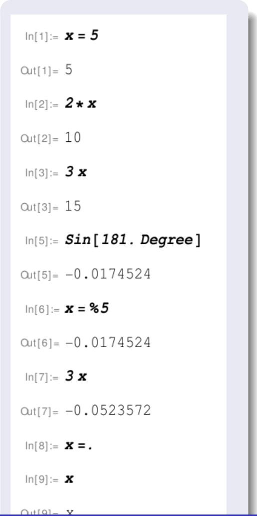
Podstawowe funkcje matematyczne

- Mnożenie zapisujemy jako **x*y**, lub **x y**, ale xy to już nazwa!
- Nazwy funkcji i stałych zaczynają się od wielkiej litery, np. **Sin[x]**
- Argument funkcji jest podawany w nawiasach kwadratowych **[]**
- Dając kropkę na końcu argumentu, jako wynik otrzymamy przybliżenie numeryczne
- Funkcje trygonometryczne domyślnie wymagają argumentu w radianach
- Argument w stopniach przekazujemy jako **X Degree**

```
In[10]:= Sqrt[3]
Out[10]= √3
In[11]:= Exp[4]
Out[11]= e^4
In[14]:= Sin[2.0]
Out[14]= 0.909297
In[15]:= Sin[2 Degree]
Out[15]= Sin[2 °]
In[16]:= Pi
Out[16]= π
In[17]:= Infinity
Out[17]= ∞
In[18]:= Sqrt[-1]
```

Własne definicje

- Zmienne definiujemy używając znaku = (zaleca się używania małych liter w nazwach własnych zmiennych)
- Możemy użyć zmiennych by przechowywać wartości liczbowe obliczeń
- Kiedy już nie potrzebujemy zmiennej lub funkcji należy je usunąć używając **nazwa=.** lub **Clear[nazwa]**



Własne definicje

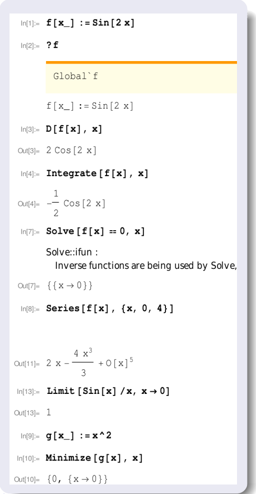
- Funkcje definiujemy używając składni **nazwa[x_]:= wzór**
- Uwaga: pamiętajmy by nie mieć zmiennych o nazwie używanej w definicji funkcji, np **x**
- Definicję funkcji własnych i wbudowanych możemy sprawdzić używając **?Nazwa**
- Kiedy już nie potrzebujemy zmiennej lub funkcji należy je usunąć używając **nazwa=.** lub **Clear[nazwa]**

```
f[x_] := x^2
?f

Global`f

f[x_] := x^2
f[16]
256
Clear[f]
?f

Global`f
```

Obliczenia symboliczne

`D`$[f,x]$	the (partial) derivative $\frac{\partial f}{\partial x}$
`Integrate`$[f,x]$	the indefinite integral $\int f\,dx$
`Sum`$[f,\{i,i_{min},i_{max}\}]$	the sum $\sum_{i=i_{min}}^{i_{max}} f$
`Solve`$[lhs{==}rhs,x]$	solution to an equation for x
`Series`$[f,\{x,x_0,order\}]$	a power series expansion of f about the point $x = x_0$
`Limit`$[f,x\text{->}x_0]$	the limit $\lim_{x\to x_0} f$
`Minimize`$[f,x]$	minimization of f with respect to x

Do elementów złożonego wyniku dostajemy się np. przez %x[[1]]



Obliczenia symboliczne

`Expand[expr]`
expands out products and positive integer powers in *expr*.

`Factor[poly]`
factors a polynomial over the integers.

`Simplify[expr]`
performs a sequence of algebraic and other transformations on *expr*, and returns the simplest form it finds.

`Simplify[expr, assum]`
does simplification using assumptions.

`FullSimplify[expr]`
tries a wide range of transformations on *expr* involving elementary and special functions, and returns the simplest form it finds.

In[15]:= **Expand[(a + b)^5]**

Out[15]= $a^5 + 5 a^4 b + 10 a^3 b^2 + 10 a^2 b^3 + 5 a b^4 + b^5$

In[16]:= **Factor[6 + 11 x + 6 x^2 + x^3]**

$(1 + x)(2 + x)(3 + x)$

In[18]:= **Sum**$\left[\frac{x\text{^}n}{n!}, \{n, 0, \text{Infinity}\}\right]$

Out[18]= e^x

In[38]:= **Simplify**$\left[\frac{1}{3(1+x)} - \frac{-1+2x}{6\left(1-x+x^2\right)} + \frac{2}{3\left(1+\frac{1}{3}(-1+2x)^2\right)}\right]$

Out[38]= $\frac{1}{1+x^3}$

Obliczenia numeryczne

`N[`*expr*`]`	numerical value of an expression
`NIntegrate[`f`, {`x`, `x_{min}`, `x_{max}`}]`	numerical approximation to $\int_{x_{min}}^{x_{max}} f\, dx$
`NSum[`f`, {`i`, `i_{min}`, Infinity}]`	numerical approximation to $\sum_{i_{min}}^{\infty} f$
`FindRoot[`lhs`==`rhs`, {`x`, `x_0`}]`	search for a numerical solution to an equation, starting with $x = x_0$
`NSolve[`lhs`==`rhs`, `x`]`	numerical approximations to all solutions of an equation
`FindMinimum[`f`, {`x`, `x_0`}]`	search for a minimum of f, starting with $x = x_0$
`NMinimize[`f`, `x`]`	attempt to find the global minimum of f

- Wartość liczbową wyniku otrzymujemy przez np. x/.%19

```
In[14]:= f[x_] := Sin[x] - Exp[x]

In[16]:= N[f[1]]
Out[16]= -1.87681

In[17]:= NIntegrate[f[x], {x, 0, 1}]
Out[17]= -1.25858

In[19]:= FindRoot[f[x] == -1.8, {x, 1}]
Out[19]= {1 → 0.963666}

In[27]:= f[x /. %19]
Out[27]= -1.8
```

Zadania

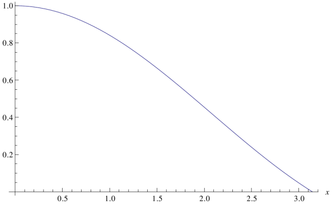
1. Wypisz wartość liczby Eulera z dokładnością do 100 cyfr
2. Wymnóż $(a+b+c)^7$
3. Zapisz w postaci iloczynowej $x^3+6x^2+11x+6$ i sprawdź wszystkie miejsca zerowe
4. Rozwiąż równanie ax^2+bx+c i sprawdź rozwiązania
5. Oblicz granice: $\lim_{n\to\infty} \sqrt[100]{n^{100}+n^{99}}-n$, $\lim_{n\to\infty} \frac{1^5+2^5+\ldots n^5}{n^6}$
6. Oblicz sumy: $S=1+2+\ldots+n$,
   $S=\frac{1}{1\cdot2\cdot3}+\frac{1}{2\cdot3\cdot4}+\ldots+\frac{1}{n\cdot(n+1)\cdot(n+2)}$
7. Oblicz pochodną wielomianu $w(x)=ax^5+(b+1)x^3+7x+1$
8. Oblicz $f^{(10)}(x)$ oraz $f^{(10)}(0)$ dla $f(x)=x^2\cdot\cos(2x)$
9. Oblicz symbolicznie następujące całki i sprawdź je licząc pochodne: $\int(x^2-2x+3)\exp(x)dx$, $\int\sqrt{x}(\log x)^2dx$

Wykresy funkcji

In[31]:= $f[x\_] := \dfrac{Sin[x]}{x}$

In[36]:= `Plot[f[x], {x, 0, Pi}, AxesLabel → Automatic]`

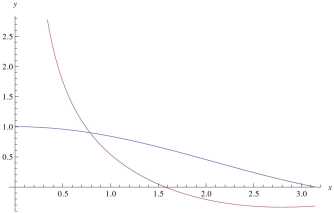
Out[36]=



In[34]:= $g[x\_] := \dfrac{Cos[x]}{x}$

In[37]:= `Plot[{f[x], g[x]}, {x, 0, Pi}, AxesLabel → {x, y}]`

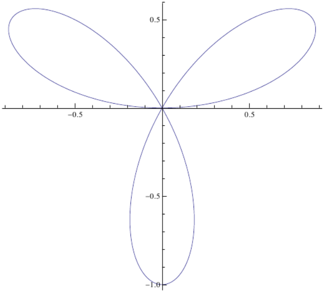
Out[37]=



Wykresy funkcji w układzie biegunowym

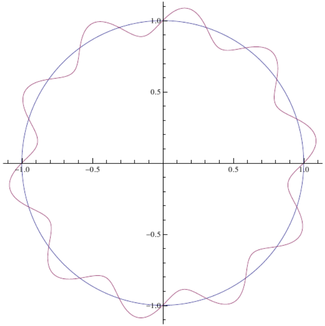
In[47]:= `PolarPlot[Sin[3 t], {t, 0, Pi}]`

Out[47]=



In[48]:= `PolarPlot[{1, 1 + 1/10 Sin[10 t]}, {t, 0, 2 Pi}]`

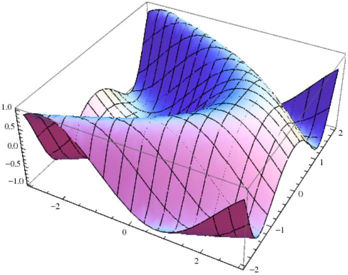
Out[48]=



Wykresy funkcji 2D

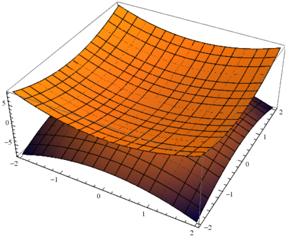
In[49]:=
```
Plot3D[Sin[x + y^2], {x, -3, 3}, {y, -2, 2}]
```

Out[49]=



In[50]:=
```
Plot3D[{x^2 + y^2, -x^2 - y^2}, {x, -2, 2}, {y, -2, 2}, ColorFunction → "RustTones"]
```

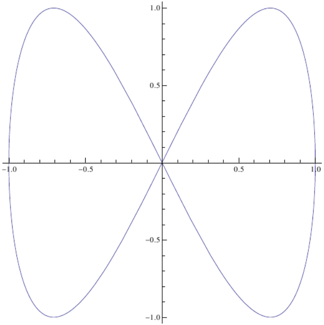
Out[50]=



Wykresy parametryczne

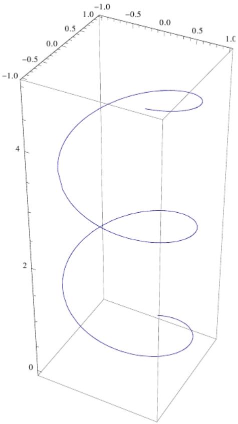
In[43]:= `ParametricPlot[{Sin[t], Sin[2 t]}, {t, 0, 2 Pi}]`

Out[43]=



In[46]:= `ParametricPlot3D[{Sin[t], Cos[t], t / 3}, {t, 0, 15}]`

Out[46]=



Wykresy: zadania

1. Narysuj wykres funkcji $\sin(x)$ dla $x \in 0, \pi$
2. Na jednym rysunku narysuj wykres funkcji $\cos(x)$ i e^x dla $x \in 0, \frac{\pi}{4}$
3. Narysuj wykres zadany równaniami: $x(t) = r \cdot (t - sin(t))$, $y(t) = r \cdot (1 - cos(t))$
4. Narysuj zbór punktów spełniających równianie $x^2 + y^2 = 9$
5. W układzie biegunowym narysuj krzywą zadaną równaniem: $r(\phi) = \frac{1+\epsilon}{1+\epsilon \cdot cos(\phi)}$, $0 < \epsilon < 1$

Wszystkie wykresy powinny mieć podpisane osie, i jeżeli trzeba opisane legendy

Wskazówka: sprawdź w dokumentacji hasła **PlotLegend** oraz **ContourPlot**

Wykresy: badanie funkcji

Mamy zadaną funkcję

$$f(x) = \frac{x^3}{1 - x^2},$$

której dziedziną jest zbiór $Df = \mathbb{R} \backslash \{-1, 1\}$. Wyznacz:

1. Miejsca zerowe
2. Granice funkcji dla punktów $x \to \infty$, $x \to -\infty$, $x \to 1^+$, $x \to 1^-$
3. Asymptotę ukośną $y = ax + b$, gdzie

$$a = \lim_{x \to \infty} \frac{f(x)}{x}$$

$$b = \lim_{x \to \infty} (f(x) - ax)$$

4. Punkty przegięcia ($f'' = 0$) oraz jej wartość w tych punktach

Stwórz wykres prezentujący funkcję oraz jej asymptotę. Dobierz skale osi, tak, aby ukazać istotny fragment wykresu. Odpowiednio nazwij osie oraz

Wykresy: badanie funkcji

Wskazówka: Sprawdź w dokumentacji hasło **Assumptions and Domains**

Macierze

+, *, ^, . . . — all automatically work element-wise

Dot (.) — products of matrices, automatically handling row and column vectors
Inverse — matrix inverse (use **LinearSolve** for linear systems)

Transpose — transpose (m^{T}, entered with Esc tr Esc)
ConjugateTranspose — conjugate transpose ($m^{\dagger}$, entered with Esc ct Esc)
Tr — trace
Det — determinant
KroneckerProduct — matrix direct product (outer product)

MatrixPower — powers of numeric or symbolic matrices
MatrixExp — matrix exponential

Eigenvalues, **Eigenvectors** — exact or approximate eigenvalues and eigenvectors
Eigensystem — eigenvalues and eigenvectors together
CharacteristicPolynomial — symbolic characteristic polynomial

UWAGA: Algebraiczne mnożenie macierzy uzyskujemy przez **A.B**. Operacja **A*B** mnoży element po elemencie

```
In[9]:= {{Cos[x], Sin[x]}, {-Sin[x], Cos[x]}}
Out[9]= {{Cos[x], Sin[x]}, {-Sin[x], Cos[x]}}
In[14]:= %9 // MatrixForm
Out[14]//MatrixForm=
 ( Cos[x]   Sin[x] )
 ( -Sin[x]  Cos[x] )

In[15]:= A = %9
Out[15]= {{Cos[x], Sin[x]}, {-Sin[x], Cos[x]}}
In[16]:= B = Transpose[A]
Out[16]= {{Cos[x], -Sin[x]}, {Sin[x], Cos[x]}}
In[19]:= A * B
Out[19]= {{Cos[x]^2, -Sin[x]^2}, {-Sin[x]^2, Cos[x]^2}}
In[20]:= A.B
Out[20]= {{Cos[x]^2 + Sin[x]^2, 0}, {0, Cos[x]^2 + Sin[x]^2}}
In[22]:= Simplify[%20] // MatrixForm
Out[22]//MatrixForm=
 ( 1  0 )
 ( 0  1 )
In[24]:= Simplify[B.A] // MatrixForm
Out[24]//MatrixForm=
 ( 1  0 )
 ( 0  1 )
In[27]:= Simplify[Inverse[A] - B // MatrixForm]
Out[27]//MatrixForm=
 ( 0  0 )
 ( 0  0 )
In[30]:= Simplify[Det[A]]
Out[30]= 1
```

Macierze

Wektory i wartości własne macierzy: $\mathbb{A}$ - macierz, $\vec{x}$ - wektor. $\vec{x}$ jest wektorem własnym macierzy, a λ jest wartością własną macierzy $\mathbb{A}$, to zachodzi związek:

$$\mathbb{A} \cdot \vec{x} = \lambda \cdot \vec{x}$$

Eigensystem

`Eigensystem[m]`
gives a list {*values*, *vectors*} of the eigenvalues and eigenvectors of the square matrix *m*.

`Eigensystem[{m, a}]`
gives the generalized eigenvalues and eigenvectors of *m* with respect to *a*.

`Eigensystem[m, k]`
gives the eigenvalues and eigenvectors for the first *k* eigenvalues of *m*.

`Eigensystem[{m, a}, k]`

```
In[67]:= A = {{2, 0}, {0, 2}}
Out[67]= {{2, 0}, {0, 2}}
In[72]:= Eigensystem[A] // MatrixForm
Out[72]//MatrixForm= (   2        2
                      {0, 1}   {1, 0} )
In[75]:= %64[[1]]
Out[75]= {2, 2}
In[76]:= %64[[1, 1]]
Out[76]= 2
In[79]:= x = %64[[2, 1]]
Out[79]= {0, 1}
In[81]:= (A.x) .x / Norm[x]
Out[81]= 2
```

Macierze: obroty

- Skonstruuj macierz obrotu o kąt $\frac{\pi}{2}$ wokół osi **z**
- Skonstruuj macierz obrotu o kąt $\frac{\pi}{2}$ wokół osi **x**
- Znajdź współrzędne wektora $\vec{r} = (1, 0, 0)$ po obrocie najpierw wokół osi **x**, potem osi **z**
- Znajdź macierz odwrotną do macierzy obrotu będącego złożeniem obrotów wokół osi **z** potem **x** i zadziałaj nią na obrócony wektor $\vec{r}$
- Narysuj wektor po każdym z obrotów.

Wskazówka 1: Sprawdź w dokumentacji hasła **RotationMatrix** i **VectorPlot3D**
Wskazówka 2: Przy rysowaniu wektora użyj parametru **VectorPoints -> 2**

Macierze: drgania

Znajdź i sprawdź wartości i wektory własne macierzy, następnie wyznacz wartości ω dla których wartości własne się zerują

$$\begin{pmatrix} -2\alpha+\omega^2 & \alpha \\ \alpha & -2\alpha+\omega^2 \end{pmatrix}$$

$$\begin{pmatrix} \frac{k}{m}-\omega^2 & -\frac{k}{m} & 0 \\ -\frac{k}{M} & \frac{2k}{M}-\omega^2 & -\frac{k}{M} \\ 0 & -\frac{k}{m} & \frac{k}{m}-\omega^2 \end{pmatrix}$$

Rozwiązywanie układów równań algebraicznych

`Solve[`*lhs*`==`*rhs*`,` *x*`]`	solve an equation, giving a list of rules for x
x`/.` *solution*	use the list of rules to get values for x
expr`/.` *solution*	use the list of rules to get values for an expression
`Solve[{`lhs_1`==`rhs_1`,` lhs_2`==`rhs_2`, ...}, {`x`,` y`, ...}]`	solve a set of simultaneous equations for x, y, ...

```
In[9]:=  r1[x_, y_, z_] := 2 x + 3 y - z - 1
In[10]:= r2[x_, y_, z_] := x - y + z - 2
In[11]:= r3[x_, y_, z_] := 3 x + 2 y - 5
In[12]:= Solve[{r1[x, y, z] == 0, r2[x, y, z] == 0, r3[x, y, z] == 0}, {x, y, z}]
Out[12]= {}
In[13]:= r1[x_, y_, z_] := 2 x + y + 3 z - 9
In[14]:= r2[x_, y_, z_] := x - 2 y + z + 2
In[15]:= r3[x_, y_, z_] := 3 x + 2 y + 2 z - 7
In[16]:= Solve[{r1[x, y, z] == 0, r2[x, y, z] == 0, r3[x, y, z] == 0}, {x, y, z}]
Out[16]= {{x → -1, y → 2, z → 3}}
In[27]:= r1[x /. %16[[1]], y /. %16[[1]], z /. %16[[1]]]
Out[27]= 0
In[28]:= r2[x /. %16[[1]], y /. %16[[1]], z /. %16[[1]]]
Out[28]= 0
In[29]:= r3[x /. %16[[1]], y /. %16[[1]], z /. %16[[1]]]
Out[29]= 0
In[30]:= r1[x /. %16[[1]], y /. %16[[1]], 2]
Out[30]= -3
In[32]:= r1[x, y, z] /. %16[[1]]
Out[32]= 0
```

Układy równań algebraicznych: zadania

Rozwiąż układy równań, i sprawdź rozwiązania:

$$\begin{cases} x - y + 2z = 1 \\ x - 2y - z = 2 \\ 3x - y + 5z = 3 \\ -2x + 2y + 3z = -4 \end{cases}$$

$$\begin{cases} x - y + 5z - u = 0 \\ x + y - 2z + 3u = 0 \\ 3x - y + 8z + u = 0 \\ x + 3y - 9z + 7u = 0 \end{cases}$$

Wskazówka: W drugim układzie załóż, że zmienne z i u mogą mieć dowolne wartości

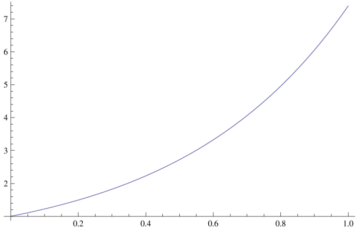
Rozwiązywanie równań różniczkowych pierwszego stopnia

`DSolve[`eqn, y, x`]`
solves a differential equation for the function y, with independent variable x.

`DSolve[{`eqn_1, eqn_2, …`}, {`y_1, y_2, …`}, `x`]`
solves a list of differential equations.

`DSolve[`eqn, y, `{`x_1, x_2, …`}]`
solves a partial differential equation.

```
In[1]:=  DSolve[y'[x] == y[x], y[x], x]
Out[1]=  {{y[x] → e^x C[1]}}
In[8]:=  DSolve[{y'[x] == a y[x], y[0] == 1}, y, x]
Out[8]=  {{y → Function[{x}, e^(a x)]}}
In[11]:= y'[x] - a y[x] /. %8[[1]]
Out[11]= 0
In[14]:= y[x] /. %8[[1]]
Out[14]= e^(a x)
In[15]:= a = 2
Out[15]= 2
In[16]:= Plot[y[x] /. %8[[1]], {x, 0, 1}]
Out[16]=
```



Rozwiązywanie równań różniczkowych drugiego stopnia

In[13]:=
```
DSolve[{x''[t] + ω^2 x[t] == 0, x[0] == x0, x'[0] == v0}, x, t]
```

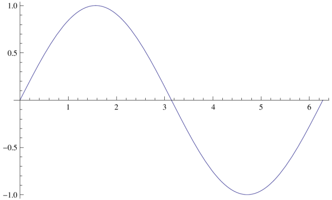
Out[13]= $\{\{x \to \text{Function}[\{t\}, \frac{1}{\omega}(x0\,\omega\,\text{Cos}[t\,\omega] + v0\,\text{Sin}[t\,\omega])]\}\}$

In[41]:=
```
{x0 = 0, v0 = 1, ω = 1}
```

Out[41]= {0, 1, 1}

In[43]:=
```
Plot[x[t] /. %13[[1]], {t, 0, 2 Pi}]
```

Out[43]=



In[44]:=
```
x[t] /. %13[[1]]
```

Out[44]= Sin[t]

In[27]:=
```
Clear[Evaluate[Context[] <> "*"]]
```

In[28]:=
```
DSolve[{x''[t] + 2 b x'[t] + ω^2 x[t] == 0, x[0] == x0, x'[0] == v0}, x, t]
```

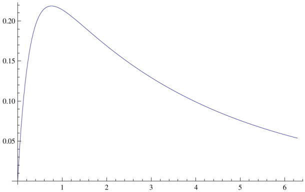
Out[28]= $\{\{x \to \text{Function}[\{t\}, \frac{1}{2\sqrt{b^2-\omega^2}}\left(-e^{t\left(-b-\sqrt{b^2-\omega^2}\right)} v0 + e^{t\left(-b+\sqrt{b^2-\omega^2}\right)} v0 - b\,e^{t\left(-b-\sqrt{b^2-\omega^2}\right)} x0 + b\,e^{t\left(-b+\sqrt{b^2-\omega^2}\right)} x0 + e^{t\left(-b-\sqrt{b^2-\omega^2}\right)} x0\sqrt{b^2-\omega^2} + e^{t\left(-b+\sqrt{b^2-\omega^2}\right)} x0\sqrt{b^2-\omega^2}\right)]\}\}$

In[39]:=
```
{x0 = 0, v0 = 1, ω = 1, b = 2}
```

Out[39]= {0, 1, 1, 2}

In[40]:=
```
Plot[x[t] /. %28[[1]], {t, 0, 2 Pi}]
```

Out[40]=



Rozwiązywanie układów równań różniczkowych

```
In[50]:= Clear[Evaluate[Context[] <> "*"]]

In[54]:= DSolve[{x''[t] + ω1^2 x[t] == 0, y''[t] + ω2^2 y[t] == 0,
           x[0] == x0, x'[0] == v1, y'[0] == v2, y[0] == y0}, {x, y}, t]
```

Out[54]= $\left\{\left\{x \to \text{Function}\left[\{t\}, \frac{\text{Sin}[t\,\omega_1]\,v_1 + x0\,\text{Cos}[t\,\omega_1]\,\omega_1}{\omega_1}\right], y \to \text{Function}\left[\{t\}, \frac{\text{Sin}[t\,\omega_2]\,v_2 + y0\,\text{Cos}[t\,\omega_2]\,\omega_2}{\omega_2}\right]\right\}\right\}$

```
In[58]:= {x0 = 0, y0 = 0, v1 = 1, v2 = 1, ω1 = 1, ω2 = 2}

Out[58]= {0, 0, 1, 1, 1, 2}

In[59]:= {x[t], y[t]} /. %54[[1]]
```

Out[59]= $\left\{\text{Sin}[t], \frac{1}{2}\,\text{Sin}[2\,t]\right\}$

```
In[60]:= ParametricPlot[{x[t], y[t]} /. %54[[1]], {t, 0, 2 Pi}]
```

Out[60]=