Wykład 7

Operacje symboliczne

Pakiet Symbolic Math Toolbox umożliwia pracę na zmiennych symbolicznych i pozwala rozwiązywać problemy na drodze przekształceń analitycznych, bez odwoływania się do wartości liczbowych poszczególnych symboli. Przekształceniem może być uproszczenie złożonego wyrażenia algebraicznego, wyznaczenie wzoru na pierwiastek równania, obliczenie granicy, pochodnej czy całki zadanej funkcji.

Wyrażenia symboliczne w Matlabie stanowią zapis strukturalnej relacji pomiędzy symbolicznymi zmiennymi i stałymi liczbowymi. Przed przystąpieniem do obliczeń symbolicznych należy zdefiniować zmienne jako symboliczne. Wykorzystuje się w tym celu następujące polecenia:

sym	Tworzy zmienne, wyrażenia, funkcje i matryce symboliczne
syms	Tworzy zmienne i funkcje symboliczne
symfun	Tworzy funkcje symboliczne
str2sym	Oblicza wartość wyrażenia symbolicznego podanego w ciągu
fold	Składanie (fold) wektora przy pomocy funkcji
piecewise	Wyrażenie lub funkcja zdefiniowane warunkowo

Tworzenie i działania na liczbach symbolicznych

Do tworzenia liczb (stałych) symbolicznych wykorzystujemy polecenie **sym.**

Przykłady:

Działanie	Numerycznie	Symbolicznie
$\frac{1}{3}$	1/3 = 0.3333	sym(1/3) = 1/3
$\sqrt{2}$	sqrt(2) = 1.4142	a = sqrt(sym(2)) = 2^(1/2)
$\sin(\pi/4)$	sin(pi/4) = 0.7071	b = sin(sym(pi/4) = 2^(1/2)/2
$\frac{1}{2}+\frac{2}{3}$	1/2+ 2/3 = 1.1667	sym(1)/sym(2)+sym(2)/sym(3) = 7/6

Wynik działania symbolicznego można zamienić na liczbę przy pomocy polecenia **double,** np. double(a) = 1.4142, double(b) = 0.7071

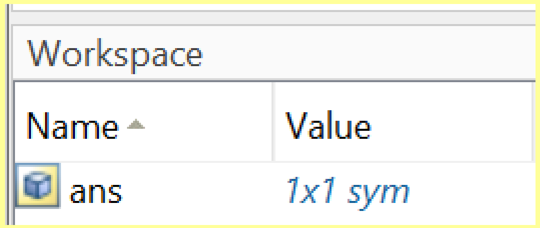
Tworzenie zmiennych i wyrażeń symbolicznych

Przed przystąpieniem do obliczeń symbolicznych należy zdefiniować zmienne jako symboliczne. Wykorzystuje się w tym celu polecenia **sym** lub **syms**:

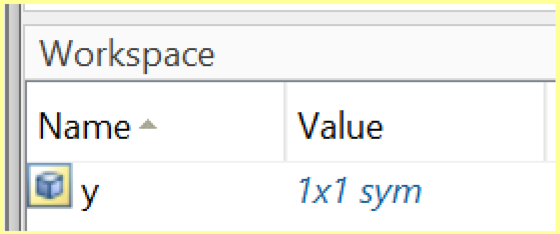
sym - tworzy numerowane zmienne symboliczne lub funkcje symboliczne Matlaba

syms – tworzy nowe zmienne symboliczne, także wiele zmiennych jednym pleceniem, np. polecenie **syms a b c d** definiuje a b c d jako zmienne symboliczne

Komenda **sym x** tworzy symboliczną zmienną x w przestrzeni roboczej Matlaba z wartością x przypisaną do zmiennej x



Komenda **y = sym('y')** tworzy symboliczną zmienną y o wartości y.



Przykład: konstruowanie złożonych obiektów symbolicznych

Zdefiniujemy zmienne symboliczne **f, g** i skostruujemy z nich zmienne złożone **c** i **d**.

f = sym('alfa'),
g= sym('beta')

Niech: c=(sin(f))^2+(cos(g))^2
d = c*sin(f)+cos(g)/c

Otrzymamy:
c = cos(beta)^2 + sin(alfa)^2
d = cos(beta)/(cos(beta)^2 + sin(alfa)^2) + sin(alfa)*(cos(beta)^2 + sin(alfa)^2)

Funkcja odwrotna

Funkcję odwrotną funkcji f tworzymy komendą **finverse(f)**

Przykład:
Jeżeli f = exp(-x) , to g = **finverse(f) =** -log(x)

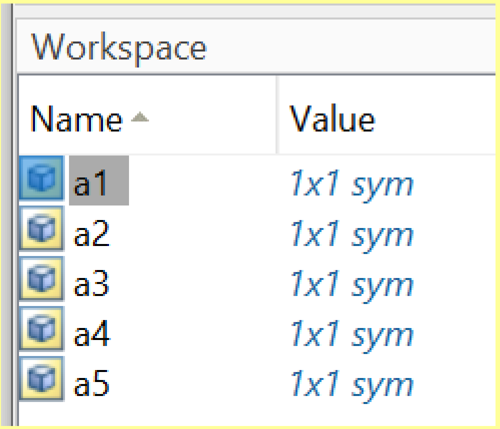
Tworzenie tablicy indeksowanych zmiennych symbolicznych A przy pomocy polecenia syms:

Polecenie A = sym('a', [1 20]) tworzy 20-elementową tablicę numerowanych zmiennych symbolicznych:

A = [a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14, a15, a16, a17, a18, a19, a20]

Tworzenie wielu symbolicznych zmiennych numerowanych:

Polecenie syms(sym('a', [1 5])) tworzy 5 numerowanych zmiennych symbolicznych w przestrzeni roboczej Matlaba:



Granice funkcji

Granicę funkcji ***f(x)*** przy ***x*** zbieżnym do h ($\lim_{x \to h} f(x)$) można policzyć przy pomocy polecenia ***limit(f, x, h)*** (wcześniej f, x, h należy zdefiniować jako zmienne symboliczne). Domyślnie (pominięcie argumentu ***h***) liczona jest granica przy ***x*** zbieżnym do zera.

Przykłady:

Działanie	Zapis	Wynik
$\lim_{x \to 0} \frac{\sin(x)}{x}$	limit(f,x,0)	1
$\lim_{x \to -\infty} \frac{1}{x}$	imit(f,x,-Inf)	0
$\lim_{x \to 0_+} \frac{1}{x}$	limit(f,x,0,'right')	Inf
$\lim_{x \to 0_-} \frac{1}{x}$	limit(f,x,0,’left')	-Inf

Różniczkowanie funkcji symbolicznych

Do obliczania pochodnych funkcji symbolicznych w Matlabie używana jest funkcja **diff**. Jej składnię przedstawiono poniżej:

diff(f) - pochodna funkcji ***f*** (funkcji jednej zmiennej) jednej zmiennej

diff(f,x) - pochodna funkcji ***f*** ze względu na zmienną ***x*** (pochodna cząstkowa)

Przykład: Jeśli $f=a\cdot x^2$, to diff(f,x)=$2a \cdot x$, a diff(f,a) = x^2

diff(f,x,n) – ***n***-ta pochodna funkcji ***f*** ze względu na zmienną ***x***

Przykład: Jeśli $f=a\cdot x^3$, to diff(f,x,2)=$6a \cdot x$, a diff(f,a,2) = 0

Ten sam efekt uzyskamy stosując polecenie **diff** n - krotnie:

diff(diff(f,x,2)= $6a \cdot x$

Df=diff(f,x,y) - pochodna mieszana funkcji ***f*** ze względu na zmienne ***x*** i ***y,*** czyli

$$\frac{\partial^2 f}{\partial xy} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right)$$

Przykład: Jeśli $f= x\cdot\sin(x \cdot y)$, to diff(f,x,y)=$2x \cdot\cos(x \cdot y)-x^2 \cdot y \cdot\sin(x \cdot y)$

Całkowanie funkcji symbolicznych

Do całkowania funkcji symbolicznych używamy polecenia **int**. Pozwala ona liczyć zarówno całki nieoznaczone, jak i oznaczone.

int(f) - całka nieznaczona z funkcji ***f*** (domyślnie po ***dx***)

int(f,x) - całka nieoznaczoną z funkcji ***f*** ze względu na wyróżnioną zmienną, w tym przypadku x

Przykład: Jeśli $f=a \cdot x^2$, to $int(f,x)=a \cdot x^3/3$, a $int(f,a) = a^2 \cdot x^2/2$

int(f,a,b) – całka oznaczona z funkcji ***f*** w granicach od **a** do **b**, przy czym a i b mogą być stałymi lub zmiennymi

Przykład: Jeśli $f=\cos(x)$, $a=-\pi/2$, $b=\pi/2$ to int(f,-pi/2, pi/2) =2

int(f,x,a,b) – całka oznaczona z funkcji ***f*** w granicach od **a** do **b**, ze względu na wyróżnioną zmienną.

Suma szeregu

Komenda **symsum(S,n,a,b)** liczy sumę **n** wyrazów szeregu **S(n)** poczynając od wyrazu **a** kończąc na wyrazie **b**.

Przykłady:

$$Sum1 = \sum_{n=0}^{5} n^2$$

Sum1 = symsum(n^2,n,1,5) = 55

$$Sum2 = \sum_{n=1}^{\infty} \frac{1}{n^2}$$

Sum2 = symsum(1/n^2,n,1,Inf) = $\pi^2/6$

Szereg Taylora

Komenda **taylor(f,x,'ExpansionPoint',a,'Order',n)** rozwija funkcję **f** w szereg Taylora **n**-tego rzędu, względem **a**, według wzoru:

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

Przykład:

Jeżeli f = exp(x^2) to polecenie rozwinięcia f(x) w szereg Taylora 4-go rzędu względem x=1 ma postać:

t = taylor(f,x,'ExpansionPoint',1,'Order',4);

Wynik:

t = exp(1) + 2*exp(1)*(x - 1) + 3*exp(1)*(x - 1)^2 + (10*exp(1)*(x - 1)^3)/3

Upraszanie wyrażeń algebraicznych

Korzystając ze znanych tożsamości matematycznych można przekształcić dowolne wyrażenie arytmetyczne do różnych postaci. Matlab oferuje kilka funkcji, które to umożliwiają:

horner(f) - doprowadza do postaci Hornera

factor(f) - przedstawia w postaci czynnikowej

expand(f) - rozwija wyrażenia według znanych tożsamości,

np. cos(x+y) = cos(x)*cos(y)-sin(x)*sin(y)

collect(f) – „zbiera" współczynniki przy x o tym samym stopniu tym samym

simplify(f) - upraszcza funkcję (wielomian do postaci iloczynowej)

Przykład: Dany jest wielomian w=(x-1)*(x+1)*(x+2)

simplify(w)=(x-1)*(x+1)*(x+2)

factor(w) = [x - 1, x + 1, x + 2]

collect(w) = x^3 + 2*x^2 - x – 2

horner(w) =x*(x*(x + 2) - 1) – 2

expand(w)=x^3 + 2*x^2 - x – 2

Zastąpienie

Stosując polecenie **[r,sigma] = subexpr(expr)** można zastąpić powtarzający się fragment **expr** rozbudowanego wyrażenia **r**, co pozwala uzyskać prostszy, bardziej zwarty zapis.

Przykład: Rozwiążmy równanie kwadratowe postaci: $ax^2+bx+c=0$

```
syms a b c d x
rozwiazanie = solve(a*x^2 + b*x+ c == 0, x, 'MaxDegree', 2);
```

Otrzymujemy następujące wyniki:

```
rozwiazanie(1,1) = -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
rozwiazanie(2,1) = -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

Zastąpmy powtarzający się fragment wyrażenia **rozwiązanie** wyrażeniem **sigma**:

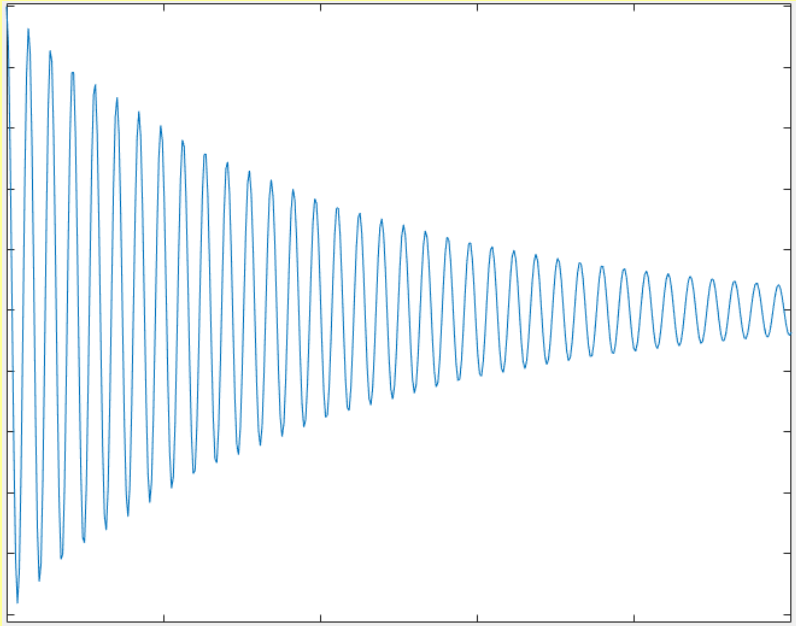
```
[r, sigma] = subexpr(rozwiazanie)
```

Ostatecznie otrzymujemy:

```
r =  -(b + sigma)/(2*a)
     -(b - sigma)/(2*a)          gdzie     sigma = (b^2 - 4*a*c)^(1/2)
```

Zastąpienie

Z kolei przy pomocy polecenia **subs(s,old,new)** można w wyrażeniu **s** zastąpić wszystkie wystąpienia **old** przez **new.**



Przykład 1.

subs(a + b, a, 4)

ans = b + 4

Przykład 2.

```
syms x m k b
s=dsolve('D2x+b/m*Dx+k/m*x=0','x(0)=10','Dx(0)=0');
s1=subs(s,b,0.1);
s2=subs(s1,m,2);
s3=subs(s2,k,10);
ezplot(s3,[0 100]);
```

Składanie funkcji symbolicznych

Polecenie **compose** służy do składania funkcji. Przykłady składni przedstawiono poniżej:

compose(f,g) = f(g(y))   gdzie f = f(x) and g = g(y)

compose(f,g,z) = f(g(z)) gdzie f = f(x), g = g(y)

Zdefiniujmy następujące funkcje:

$f = 1/(1 + x^2)$;

g = sin(y);

$h = x^t$;

p = exp(-y/u)

Składanie funkcji - przykłady:

compose(f,g) = f(g(y))   ans = 1/(1+sin(y)^2);

compose(f,g,t) = f(g(t)   ans = 1/(1+sin(t)^2);

compose(h,g,x,z) = sin(z)^t;

compose(h,p,x,y,z) = exp(-z/u)^t;

Równania postaci S(x) = 0

Do rozwiązywania równań stosuje się komendę **solve(S)**. Jeśli wyrażenie **S** jest funkcją zmiennej x, to powyższe polecenia szuka rozwiązań równania **S(x) = 0**.

Przykład:

Znajdźmy rozwiązanie równania $ax^2+bx+c = 0$:

```
syms x
S = a*x^2+b*x+c
X = solve(S)
Rozwiązanie   X =  -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
                   -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

Równania postaci S(x, y) = 0

W przypadku równania z dwiema zmiennymi **(S(x, y)=0)** należy dodatkowo wskazać względem której zmiennej równanie ma być rozwiązane. Załóżmy, że chcemy rozwiązać równanie postaci $\mathbf{x^2+y^2-3xy+1=0}$ **:**

Rozwiązanie względem **x**:

syms x y

X = solve(x^2+y^2-3*x*y+1, **x**)

Wynik: X = (3*y)/2 - (5*y^2 - 4)^(1/2)/2
(3*y)/2 + (5*y^2 - 4)^(1/2)/2

Rozwiązanie względem **y**:

syms x y

X = solve(x^2+y^2-3*x*y+1, **y**)

Wynik: Y = (3*x)/2 - (5*x^2 - 4)^(1/2)/2
(3*x)/2 + (5*x^2 - 4)^(1/2)/2

Rozwiązywanie układów równań

Polecenie **solve** służy także do rozwiązywania układów równań zapisanych w postaci wyrażeń symbolicznych:

r = solve (równaniel, równanie2, . . ,równanie n);

Przykład – układ równań liniowych:

3x+5y=2,

4x-7y=-5

[x,y] = solve(3*x+5*y==2, 4*x-7*y==-5)

Wynik: x = -11/41   y = 23/41

Rozwiązywanie układów równań

W przypadku bardziej skomplikowanego układu równań spodziewamy się, w ogólnym przypadku, większej liczby rozwiązań, rozmieszczonych w macierzy rozwiązań. Interesujące nas rozwiązanie wybieramy stosując odpowiednie komendy.

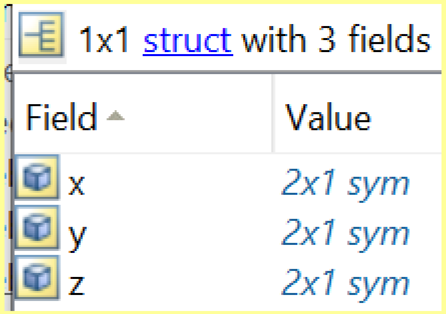
Przykład – rozwiązać układ równań: $x^2-y^2 = z,$

$$x + y = 1,$$

$$x^2-2y = 3z$$

S = solve(x^2-y^2 == z,x + y == 1,x^2-2*y == 3*z)

Dostajemy odpowiedź w postaci struktury S:



Aby wydobyć rozwiązanie stosujemy komendę:

M = [S.x, S.y, S.z]

M = [2 - 3^(1/2), 3^(1/2) - 1, 3 - 2*3^(1/2)]
[3^(1/2) + 2, - 3^(1/2) - 1, 2*3^(1/2) + 3]

lub jeszcze bardziej selektywnie: S.x = 2 - 3^(1/2)
3^(1/2) + 2

Rozwiązywanie równań różniczkowych zwyczajnych

Do rozwiązywania równań różniczkowych zwyczajnych służy komenda **dsolve**. Aby rozwiązać równanie różniczkowe postaci

$$\frac{dy}{dt} = f(x, y)$$

należy skorzystać z polecenia dsolve w jednej z dwu postaci:

y = dsolve('Dy = f(x,y)')   lub   eqn = diff(y,t) == f(x,y)

S=dsolve(eqn)

Przykład 1. Znaleźć rozwiązanie równania $\frac{dy}{dt} = a \cdot y$ bez warunków początkowych.

Sposób I

syms y a

y = dsolve('Dy = a*y')

SposóbII

syms y(t) a

eqn = diff(y,t) == a*y;

S = dsolve(eqn)

W obydwu przypadkach odpowiedź jest taka sama: y =C1*exp(a*t)

Przykład 2. Znaleźć rozwiązanie równania z poprzedniego przykładu z warunkiem początkowym y(t=0)=10

Sposób I

```
syms y a
y = dsolve('Dy =  a*y','y(0)=10')
```

SposóbII

```
syms y(t) a
eqn = diff(y,t) == a*y;
S = dsolve(eqn,y(0)==10)
```

W obydwu przypadkach odpowiedź jest taka sama: y =10*exp(a*t)

Przykład 3. Znaleźć rozwiązanie równania $\frac{d^2y}{dt^2} = a \cdot y$ bez warunków początkowych.

```
syms y(t) a
eqn = diff(y,t,2) == a*y;
ySol(t) = dsolve(eqn)
```

Odpowiedź: ySol(t) = C1*exp(-a^(1/2)*t) + C2*exp(a^(1/2)*t) gdzie C1, C2 - stałe

Przykład 4. Znaleźć rozwiązanie równania $\frac{d^2y}{dt^2} = a^2 \cdot y$ z warunkami początkowymi: y(0) = 1, y'(0)=2

```
syms y(t) a b
eqn = diff(y,t,2) == a^2*y;
Dy = diff(y,t);
cond = [y(0)==10, Dy(0)==1];
ySol(t) = dsolve(eqn,cond)
```

Odpowiedź: ySol(t) = (exp(a*t)*(10*a + 1))/(2*a) + (exp(-a*t)*(10*a - 1))/(2*a)

Układy równań różniczkowych

W pakiecie Symbolic Math Toolbox można rozwiązywać także układy równań różniczkowych zwyczajnych pierwszego rzędu z zadanymi warunkami początkowymi. Ponieważ równania wyższych rzędów dadzą się sprowadzić do układu równań pierwszego rzędu, można również rozwiązywać równania wyższych rzędów.

Układy równań różniczkowych rozwiązuje się przy pomoc polecenia dsolve o następującej składni:

r = dsolve(eql, eq2, ... eqn, in1, in2, ... inn) gdzie:

eqi, eq2, ... eqn - oznaczają kolejne równania różniczkowe rozwiązywanego układu, natomiast in1, in2, ...,, inn - oznaczają zadane warunki początkowe.

Przykład. Rozwiązać następujący układ równań różniczkowych:

$$\frac{df}{dt} = 3f + 4g \qquad \frac{dg}{dt} = -4f + 3g$$

z warunkami początkowymi f(0) = 0 i g(0) = 1

S = dsolve('Df = 3*f+4*g', 'Dg = -4*f+3*g', 'f(0) = 0, g(0) = 1’)

Wynik: S.f = sin(4*t)*exp(3*t); S.g = cos(4*t)*exp(3*t)

Transformaty

Rodzaj transformaty	Wzór	Komenda
Transformata Fourier'a:	$f(x) \rightarrow F(w) = \int\limits_{-\infty}^{+\infty} f(x) \cdot \exp(-iwx) \cdot dx$	**F=fourier(f,x,w)**
Odwrotna transformata Fourier'a	$f(x) = \frac{1}{2\pi} \int\limits_{-\infty}^{+\infty} F(w) \cdot \exp(iwx) \cdot dw$	**f=ifourier(F,w,x)**
Transformata Laplace'a	$L(s) = \int\limits_{0}^{\infty} F(t) \cdot \exp(-st) \cdot dt$	**L=laplace(F,t,s)**
Odwrotna transformata Laplace'a	$F(t) = \int\limits_{c-io}^{c+io} L(s) \cdot \exp(st) \cdot ds$	**F=ilaplace(L,s,t)**

Jako pierwszą podajemy funkcję na której będziemy dokonywać transformaty (bądź odwrotnej transformaty), jako drugą podajemy zmienną tej funkcji, a jako trzecią podajemy zmienną funkcji które będzie rozwiązaniem.

Przykłady

I. Transformata Fouriera funkcji sinus

```
syms x w
f= sin(x);
F =  fourier(f,x,w)             Wynik: F = -pi*(dirac(w - 1) - dirac(w + 1))*1i
```

II. Odwrotna transformata Fouriera funkcji cosinus

```
syms w x
F = cos(w)
f = ifourier(F,w,x)             Wynik: f = 1/2*dirac(x+1)+1/2*dirac(x-1)
```

III. Transformata Laplace'a funkcji $\sin^2 t$

```
syms t s
F = (sin(t))^2
L = laplace(F,t,s)              Wynik:   L = 2/s(s^2+4)
```

IV. Odwrotna transformata Laplace'a funkcji $2s/(s^2-4)$

```
syms s t
L = 2*s/(s^2-4)
F = ilaplace(L,s,t)
F = 2*cosh(2*t)
```

Funkcja pretty

Funkcja **pretty** pozwala zapisać wyrażenie w takiej postaci graficznej, do jakiej przywykliśmy na lekcjach matematyki.

Przykładowo, zastosowanie funkcji pretty do poniższego wyrażenia

pretty((exp(a*t)*(10*a + 1))/(2*a) + (exp(-a*t)*(10*a - 1))/(2*a))

Sprowadza je do postaci:

```
exp(a t) (10 a + 1)   exp(-a t) (10 a - 1)
------------------- + --------------------
        2 a                   2 a
```

Nowe wersje Matlaba oferują znacznie nowocześniejszy sposób prezentowania wyników obliczeń z wykorzystaniem *Live Script*. Pokażemy możliwości tego narzędzia na przykładzie analizy funkcji kwadratowej postaci ax^2+bx+c.

Otwórzmy nowe okno Live Script i wpiszmy kilka pierwszych linii:

```
syms a b c
syms f(x)
f(x)=a*x^2+b*x+c
rozw_analityczne=solve(f)
```

Po uruchomieniu zostają obliczone i przedstawione w postaci graficznej pierwiastki równania – można kopiować i wklejać:

$$\begin{pmatrix} -\dfrac{b+\sqrt{b^2-4\,a\,c}}{2\,a} \\ -\dfrac{b-\sqrt{b^2-4\,a\,c}}{2\,a} \end{pmatrix}$$

Wprowadźmy drugą funkcję f2(x)=2 i znajdźmy jej punkty przecięcia z parabolą (kontynuujemy skrypt):

```
syms f2
f2(x)=2
rozw_analityczne2=solve(f==f2)
```

$$\begin{pmatrix} -\dfrac{b+\sqrt{b^2+8\,a-4\,a\,c}}{2\,a} \\ -\dfrac{b-\sqrt{b^2+8\,a-4\,a\,c}}{2\,a} \end{pmatrix}$$

Jeśli wprowadzimy konkretne wartości liczbowe dla a, b, c (F=subs(f, [a b c], [1 2 3])), to konkretne wartości numeryczne można uzyskać przy pomocy funkcji vpasolve:

```
r_numeryczne=vpasolve(F==f2)
```

Wynik: r_numeryczne = $\begin{pmatrix} -1.0 \\ -1.0 \end{pmatrix}$

DZIĘKUJĘ ZA UWAGĘ