MPDI2

Wykład 10

Matlab c.d.

- **Wykresy**
- **Obsługa plików danych**
- **Obliczenia symboliczne (Symbolic Toolbox)**

WYKRESY 2-wymiarowe

1 sposób: Funkcja plot

Wymaga utworzenia dwóch wektorów o tej samej liczbie elementów. Wykres łączy odcinkami punkty o parach współrzędnych z obu wektorów

```
x=0:10                  %generowanie wektora – krok 1
                        % wart_pocz:wart_koncowa
y=[5.1  1.1  -2  -3  4.2  5.5  4.3  3.1  4.5  5.9  4.9]
z=[0 2 3 3 5 4 3 4 5 4 9]                %trzeci wektor
title('wykres')         %dodanie tytułu
plot(x,y)               %rysowanie wykresu y(x)
%lub 2 przebiegi  y(x) i z(x)
plot(x, y, 'r', x , z ,'k')   %r – red    k- black
```

Przykłady wykresów funkcji

```
x=0:90                  %generowanie wektora co 1
y=sind(x)               %wektor   y
plot(x,y) ,grid         %wykres z siatką
```

```
x=0:pi/50:6*pi
y=cos(2*x)./sqrt(x+1)
plot(x,y)
```

Uwaga: zapis kropkowy

tablicowe dzielenie(mnożenie, potęgowanie) wektorów

```
x = -9:1:9
z = x.^2
plot(x, z)
v = x.^3
hold on %zamrożenie okna wykresu
plot(x, v) %dorysowanie drugiej krzywej
```

2 sposób: Funkcja fplot

Wykorzystanie fplot nie wymaga wcześniejszego przygotowania wektorów x oraz f(x)

Wykres funkcji podanej jako argument tekstowy – działa, lecz komunikat, że jest nowszy sposób rysowania wykresu tzw. funkcji anonimowej – poznamy…

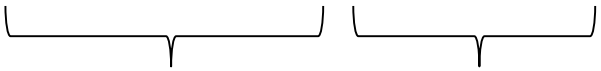
Jedna krzywa (z argumentem tekstowym):

fplot('sin(x*x)/x',[0 4*pi])

punkt dzielenia przez 0 nie jest rysowany - ostrzeżenie

dwie krzywe:

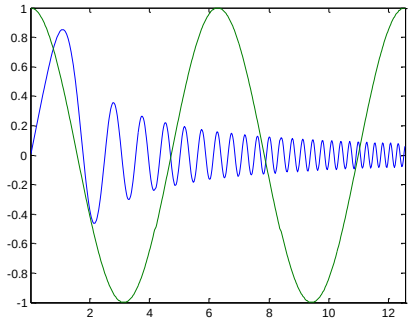
fplot('[sin(x*x)/x , cos(x)]',[0.01 4*pi])



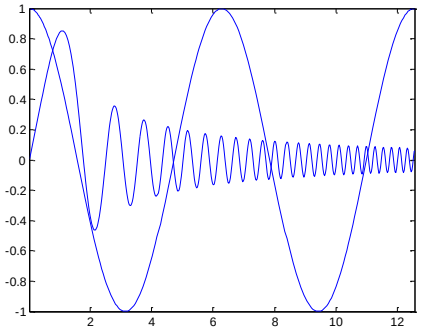
Uwaga: zamiast x można użyć innej, dowolnej nazwy zmiennej niezależnej

Sposoby rysowania wielu krzywych

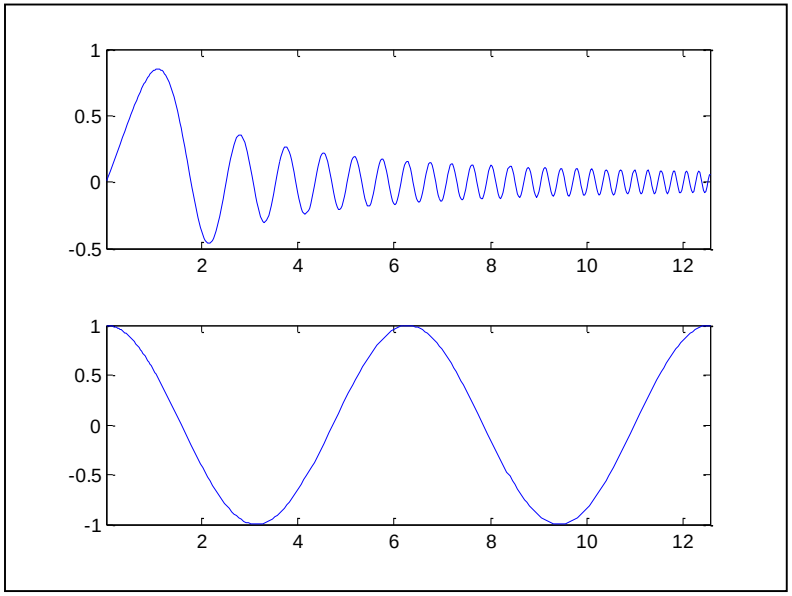
```
fplot('[sin(x*x)/x  cos(x)]',[0.01 4*pi])
```



```
fplot('sin(x*x)/x',[0.01 4*pi])
hold on
fplot('cos(x)',[0.01 4*pi])
```



```
subplot(1,2,1)
fplot('sin(x*x)/x',[0.01 4*pi])
subplot(1,2,2)
fplot('cos(x)',[0.01 4*pi])
```



Stosowanie funkcji fplot jak w poprzednich przykładach w nowych wersjach *Matlaba* jest metodą zdeprecjonowaną (choć nadal działa)

Zalecany jest poniższy sposób jak w przykładach poniżej:

```
fplot(@(x)sin(x), [0 4*pi])
```

dla jednej krzywej

```
fplot(@(x)[sin(x) , cos(x)], [0 4*pi])
```

dla wielu krzywych

lub z wykorzystaniem zapisu symbolicznego:

```
syms x
fplot([sin(x) ,  cos(x) , 0*x], [0 4*pi])
```

o czym dowiemy się za chwilę…

Inne funkcje wykorzystywane dla wykresów

axis([xmin, xmax, ymin, ymax])	- zakresy osi
grid	- siatka
xlabel(tekst)	- etykieta osi x
ylabel(tekst)	- etykieta osi y
title(tekst)	- tytuł wykresu

Wykresy 3D

Krzywe 3D

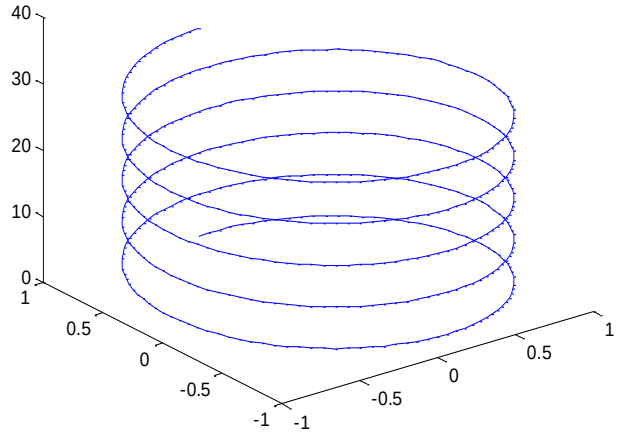
Wykorzystujemy funkcję **plot3(y,z,x)**

Przykładowo dla krzywej (**helisy**) danej równaniami:

$$y=\sin(x) \qquad z=\cos(x)$$

Tworzymy m-plik:

```
x = 0:pi/50:10*pi;
plot3(sin(x), cos(x), x);
```



Powierzchnie 3D

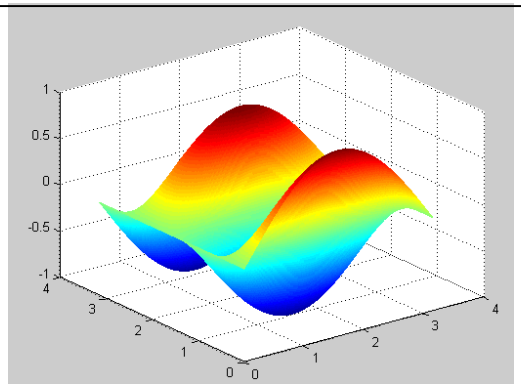
Korzystamy z funkcji **mesh**(x,y,z)

Dla powierzchni podanej równaniem:

$$z(x,y) = \cos 3x \sin y$$

Piszemy m-plik:

```
clear
x = [0:0.01:pi]'        %wektor x (kolumnowy!)
y = x'                  %wektor wierszowy y
z=cos(3*x) * sin(y)     %tablica z
mesh(x, y, z)
```



Przykładowe funkcje rysujące standardowe wykresy powierzchniowe 3D

peaks

cylider (*średnica*)

sphere (*precyzja*)

Obsługa plików

Zapis całej tablicy do pliku ASCII i odczyt z pliku

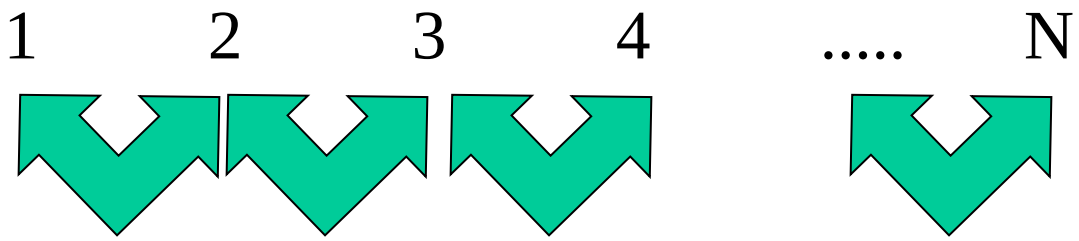
```
clear
c = [8 6 4 2;4 -1 4 5]
save ('mydata.dat', 'c','-ASCII') %zapis
clear                              %usunięcie zmiennych
load ('mydata.dat')                %odczyt
disp('Dane z pliku:');
mydata
```

Zapis danych do pliku typu *mat* i odczyt z pliku

```
a=rand(3);
b=6;
save ('plik.mat', 'a', 'b');
clear a
clear b
load ('plik.mat')
whos
disp(a)
disp(b)
```

Sortowanie bąbelkowe

Porównywanie kolejnych par elementów sąsiadujących i **zamiana miejscami** w przypadku niewłaściwej kolejności



N-1 porównań **Wykonujemy N przebiegów**

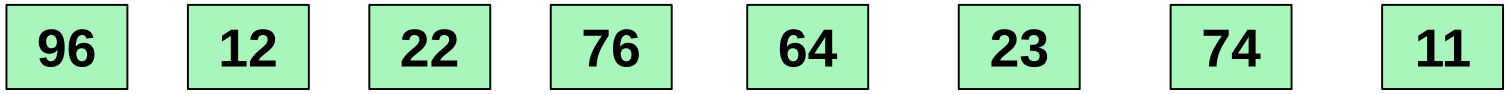
Sortowanie bąbelkowe skrócone

Przebiegów wykonujemy N-1

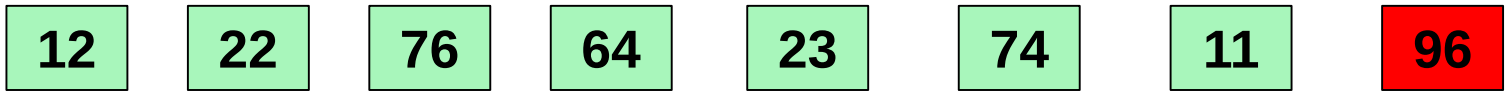
W każdym kolejnym przebiegu liczba analizowanych par jest zmniejszana o 1

Przykładowo

1 PRZEBIEG



2 PRZEBIEG



itd.

M-plik - sortowanie "bąbelkowe"

```
clc
clear
N=5;
G=rand(1,N)
%stadardowa funkcja sortująca  'sort'
G1=sort(G)
%sortowanie bąbelkowe
for k=1:N-1
   for m=1:N-k
      if G(m)>G(m+1)
         pom=G(m);
         G(m)=G(m+1);
         G(m+1)=pom;
      end
   end
   disp(G) %pokazuje kolejne wypływające "bąbelki"
end
%ostatecznie po posortowaniu
disp(G)
```

zamiana miejscami gdy elementy w niewłaściwej kolejności

Obliczenia symboliczne

Symbolic ToolBox to dodatkowy zestaw narzędziowy do Matlaba

Wymaga odrębnej licencji (wersja zakupiona dla studentów przez PRz posiada domyślnie pakiet Symbolic Toolbox)

W można go doinstalować do podstawowej wersji Matlaba (menu Home/Add-Ons/Get Add-Ons)

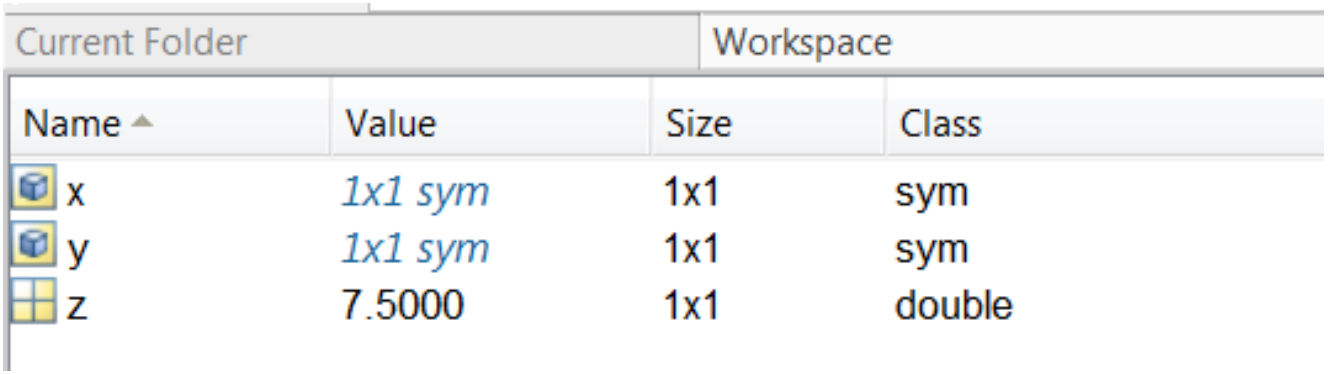
Obliczenia symboliczne wymagają zadeklarowania **zmiennych symbolicznych** – abstrakcyjnych zmiennych nie posiadających wartości.

Służy do tego polecenie:

syms *zmienna1 zmienna2 ...*

```
clc, clear
syms x y     %dwie zmienne  symboliczne
z=7.5        %a to zmienna typu liczbowego
```

Obserwacja zmiennych i ich typów w Workspace



Symboliczne operacje macierzowe

```
syms a b c d e f g h
A=[a b; c d]
```

A =

[a, b]
[c, d]

```
B=[e f ; g h]
```

B =

[e, f]
[g, h]

Ilustracja wzorów na iloczyny macierzy kwadratowych

```
il_m=A*B   %iloczyn macierzowy Cauchy'ego
il_t=A.*B  %iloczyn tablicowy Hadamarda
```

il_m =

[a*e+b*g, a*f+b*h]
[c*e+d*g, c*f+d*h]

il_t =

[a*e, b*f]
[c*g, d*h]

Symboliczne rozwiązywanie równań – funkcja solve()

Przykład:

```
syms x a
f=a - x^2
```

```
f =
 
a - x^2
```

```
r=solve(f, x)   %domyślnie f==0
```

```
r=
 
  a^(1/2)
 -a^(1/2)
```

Można też używać tożsamości:

```
solve(f==0, x)
```

Inny przykład:

```
clc,clear
syms x
r = solve(sin(x)==cos(2*x))
```

```
r =

   pi/6
```

```
r2=double(r)
%funkcja double przelicza wartość symboliczną na liczbę dziesiętną
```

```
r2 =

   0.5236
```

Można też rozwiązać układ równań, przykładowo:

```
clc,clear
syms x y
f1=5*x+5*y==-2.1;   %1 równanie
f2=3.5*x+4*y==7;    %2 równanie
rozw=solve(f1, f2) ;
x=double(rozw.x)    %pierwszy element struktury
y=double(rozw.y)    %drugi element struktury
```

```
x =
  -17.3600
y =
   16.9400
```

Obliczenia liczbowe na wyrażeniach symbolicznych

Funkcja subs( ) - podstawienie wartości do wyrażenia symbolicznego

Przykład:

```
syms a b c x      % definicja 4 zmiennych symbolicznych
y = solve(a*x^2+b*x+c)  % rozwiązanie równania względem  x
a=3; b=4; c=1;   % Przypisanie wartości liczbowych a b c
w = double(subs(y))      % Podstawienie i przeliczenie na dziesiętne
```

```
y =
 
          -1/2*(b-(b^2-4*a*c)^(1/2))/a
          -1/2*(b+(b^2-4*a*c)^(1/2))/a
w =
          -1.0000
          -0.3333
```

Inny przykład działania funkcji subs

```
syms a b c d
M=[ a b; c d]  %macierz
a = 5          %teraz a jest typu liczbowego
M2= subs(M)
```

```
M =
 [a, b]
 [c, d]
 a =
     5
M2 =
   [5, b]
   [c, d]
```

Obliczenia granic ciągów i funkcji - funkcja limit( )

Do obliczania granic na podstawie wyrażenia symbolicznego służy funkcja **limit**.

Jej składnia może być następująca:

limit(F,*zmienna*,*b*)

wyznaczenie granicy dla wyrażenia symbolicznego F, względem wskazanej zmiennej,

granica dla *zmiennej*$\rightarrow$*b*,

Uwagi:

- zmienna jest opcjonalna, jeśli wyrażenie zawiera jedną zmienną.
- *b* opcjonalne, jego pominięcie oznacza granicę dla *zmienna*$\rightarrow$0.

Wyznaczenie granicy lewostronnej dla wyrażenia symbolicznego F, w punkcie *b*,

limit(F, *zmienna*, *b*, 'left')

Wyznaczenie granicy prawostronnej dla wyrażenia symbolicznego F, w punkcie *b*.

limit(F, *zmienna*, *b*, 'right')

Przykład

Obliczenie granicy ciągu:

$$\lim_{n\to\infty} \frac{1-3n}{1+n}$$

```
syms n
g=limit((1-3*n)/(1+n), inf)
```

```
g =
    -3
```

Uwaga: ***inf*** jest symbolem ∞ (nieskończoność)

Przykład:

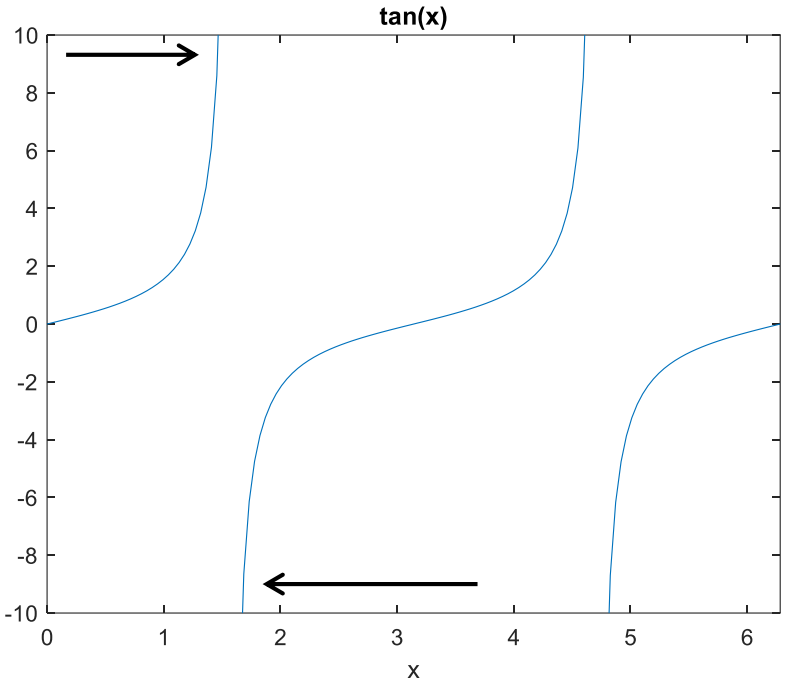
Obliczenie granicy w punkcie nieciągłości funkcji

```
syms x
g1=limit(tan(x),x, pi/2,'left')
```

```
g1 =
        Inf
```

```
g2=limit(tan(x),x, pi/2,'right')
```

```
g2 =
       -Inf
```




```
syms x
f=tan(x)
ezplot(f,[ 0 ,2*pi, -10, 10])
```

Obliczanie pochodnych funkcji - funkcja diff( )

Argumentami funkcji są:

- funkcja, której pochodna będzie liczona,
- zmienna, względem której pochodna jest liczona (opcjonalnie)
- rząd pochodnej (opcjonalnie)

diff (*funkcja*, *zmienna* , *N*)

opcjonalnie

Przykłady:

Obliczenie pochodnej funkcji: $f(x)=x^2$

```
syms x
df=diff(x^2)
```

```
df =
        2*x
```

Obliczenie pochodnej iloczynu funkcji: $f(x)=\sin x * x^2$

```
syms x
df=diff(sin(x)*x^2)
```

```
df =
        x^2*cos(x) + 2*x*sin(x)
```

matematyka ... f g' + f' g

Przykład:

Obliczenie pochodnej funkcji wielu zmiennych: **f(x, y, z)** $= xyz^x + \left(\frac{1}{xy}\right)^2$

Według każdej zmiennej (**pochodne cząstkowe**):

```
syms x y z
f=(x*y*z)^x+(1/(x*y))^2
dfx=diff(f)
dfx=diff(f,x) %dla sprawdzenia zmiennej domyślnej
dfy=diff(f,y)
dfz=diff(f,z)
```

```
dfx =
        (x*y*z)^x*(log(x*y*z)+1)-2/x^3/y^2
dfx =
        (x*y*z)^x*(log(x*y*z)+1)-2/x^3/y^2
dfy =
        x^2*z*(x*y*z)^(x - 1) - 2/(x^2*y^3)
dfz =
        x^2*y*(x*y*z)^(x - 1)
```

Przykład:

Obliczenie drugiej pochodnej $\frac{\partial^2 f}{dx^2}$

funkcji: $f(x, y, z) = xyz^x + \left(\frac{1}{xy}\right)^2$

```
syms  x y z
f=(x*y*z)^x+(1/(x*y))^2
u=diff(f,x,2)    %pochodna 2-go rzędu, po x
```

u=

```
(x^3*y^2*(x*y*z)^x + x^4*y^2*(x*y*z)^x + x^4*y^2*log(x*y*z)^2*(x*y*z)^x +
2*x^4*y^2*log(x*y*z)*(x*y*z)^x + 6)/(x^4*y^2)
```

Można też liczyć pochodną drugiego rzedu (i wyższych) poniższym sposobem:

```
u=diff(diff(f,x))
```

Całkowanie funkcji - funkcja int( )

Jej argumentem jest **funkcja symboliczna**, opcjonalnie także zmienna całkowania oraz **granice całkowania** (dla całek oznaczonych).

int *(funkcja, zmienna , a , b)*

Uwaga: przy pominiętej zmiennej - domyślnie zmienną symboliczną jest ***x***

Przykład:

Obliczenie całki nieoznaczonej funkcji f(a,x)=a+x

```
syms a x
c=int(a+x) % domyślna zmienna to x
```

```
c =
                    a*x+1/2*x^2
```

```
c=int(a+x, a)          % teraz zmienna to a
```

```
c =
                    1/2*a^2+a*x
```

Sprawdzenie:

```
s=diff(f)
```

```
s =
   a+x
```

Obliczenie całki oznaczonej: $\int_1^3 x^2 dx$

```
syms x
c=int(x^2, 1, 3)
```

```
c=
      26/3
```

Obliczenie całki oznaczonej funkcji *sin(x)* w przedziale (0, π)

```
syms x
c=int(sin(x), 0, pi)
```

```
c =
     2
```

Wykresy funkcji symbolicznych – funkcje ezplot i fplot

ezplot(f)

domyślny przedział: x (-2π, 2π)

ezplot(f, [xmin xmax])

ezplot(f, [xmin xmax ymin ymax])

lub

fplot(f)

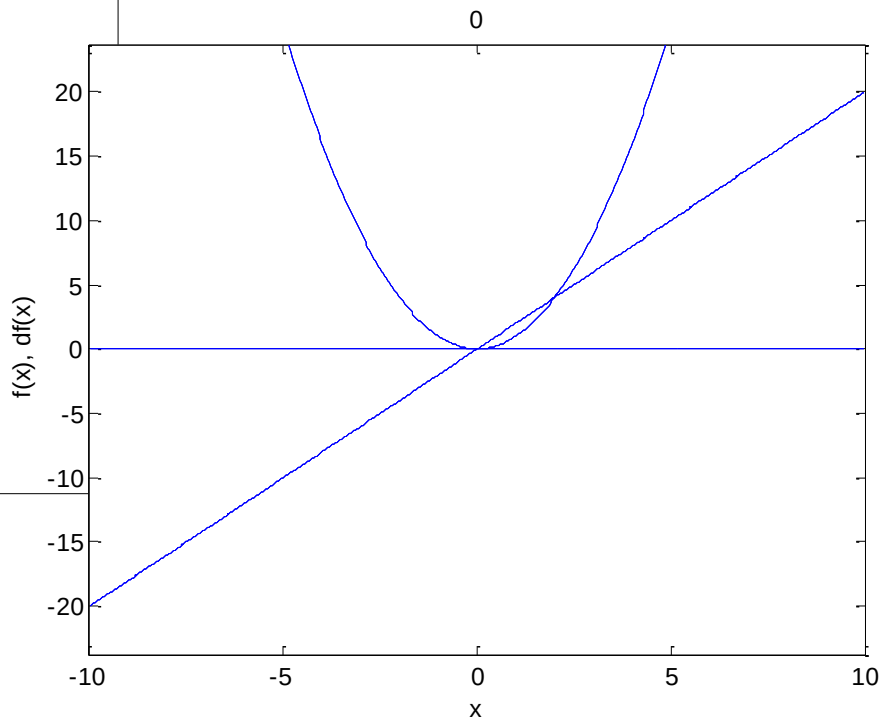
domyślny przedział: x (-5, 5)

fplot(f, [xmin xmax])

fplot([f1 f2], [xmin xmax])

Przykład

```
clc, clear
syms x
hold on   %wiele krzywych
f=x^2
ezplot(f,[-10 10])
df=diff(f)
ezplot(df,[-10 10])
f0=0*x
ezplot(f0,[-10 10]) %oś x
```



albo ...

```
clc,clear
syms x
f=x^2
df=diff(f)
f0=0*x
fplot([f df  f0],[-10 10])
```

Można też ustalić zakresy obu osi:

```
syms x

f=x^2

fplot(f)

axis([-5 5 0 15])
```