



Politechnika Białostocka  
Wydział Elektryczny  
Katedra Elektrotechniki Teoretycznej i Metrologii

Instrukcja  
do pracowni specjalistycznej z przedmiotu  
**Technologie Informacyjne**

## **MATLAB CZ. 3 TWORZENIE WYKRESÓW FUNKCJI**

Numer ćwiczenia

**TI12**

Autor:  
dr inż. Jarosław Forenc

Białystok 2016

# Spis treści

<b>1. Opis stanowiska .....</b>	<b>3</b>
1.1. Stosowana aparatura .....	3
1.2. Oprogramowanie.....	3
<b>2. Wstęp teoretyczny .....</b>	<b>3</b>
2.1. Tworzenie i usuwanie okien graficznych.....	3
2.2. Grafika dwuwymiarowa (2D) .....	5
2.3. Grafika trójwymiarowa (3D).....	14
<b>3. Przebieg ćwiczenia.....</b>	<b>19</b>
<b>4. Literatura.....</b>	<b>20</b>
<b>5. Zagadnienia na zaliczenie.....</b>	<b>21</b>
<b>6. Wymagania BHP .....</b>	<b>21</b>

---

**Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.**

© Wydział Elektryczny, Politechnika Białostocka, 2016 (wersja 1.1)

Wszelkie prawa zastrzeżone. Żadna część tej publikacji nie może być kopiowana i odtwarzana w jakiegokolwiek formie i przy użyciu jakichkolwiek środków bez zgody posiadacza praw autorskich.

# 1. Opis stanowiska

## 1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows (XP/Vista/7).

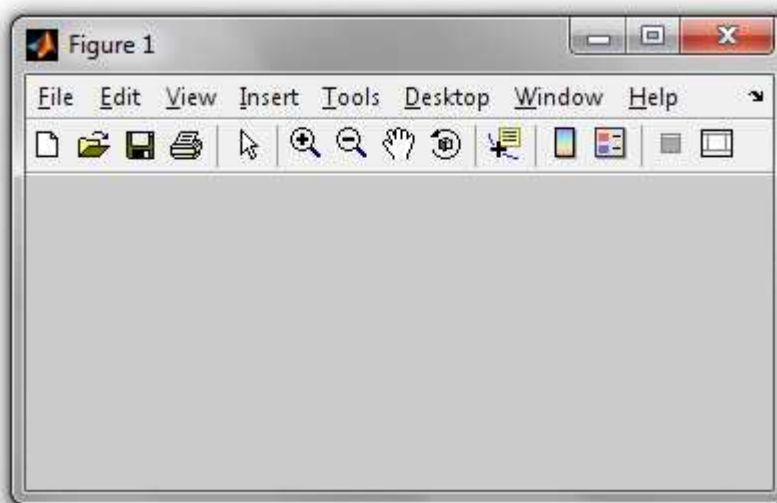
## 1.2. Oprogramowanie

Na komputerach zainstalowane jest środowisko Matlab R2007b (Version 7.5.0.342), classroom license.

# 2. Wstęp teoretyczny

## 2.1. Tworzenie i usuwanie okien graficznych

Grafika w Matlabie wyświetlana jest w **oknach graficznych**. Jednocześnie może być otwartych kilka okien graficznych, ale tylko jedno z nich jest oknem **aktywnym** (czyli takim, w którym wyświetlane są wyniki działania funkcji graficznych). Jeśli żadne okno graficzne nie jest otwarte, to funkcje graficzne automatycznie tworzą nowe okno. Każde okno graficzne ma unikalny numer wyświetlany w jego nagłówku, np. **Figure 1**, **Figure 2**.



Rys. 1. Okno graficzne

Do obsługi okien graficznych stosowane są poniższe funkcje.

<b>figure</b>	tworzy nowe okno graficzne
<b>figure (n)</b>	tworzy nowe okno graficzne o numerze <b>n</b> (jeśli okno o tym numerze już istnieje, to staje się aktywnym)
<b>close</b>	zamyka aktywne okno graficzne
<b>close (n)</b>	zamyka okno graficzne o numerze <b>n</b>
<b>close all</b>	zamyka wszystkie okna graficzne
<b>clf</b>	czyści zawartość aktywnego okna graficznego

Wyświetlenie w tym samym oknie nowego wykresu powoduje usunięcie poprzedniego. Ustawienie to można zmienić poleceniem **hold**.

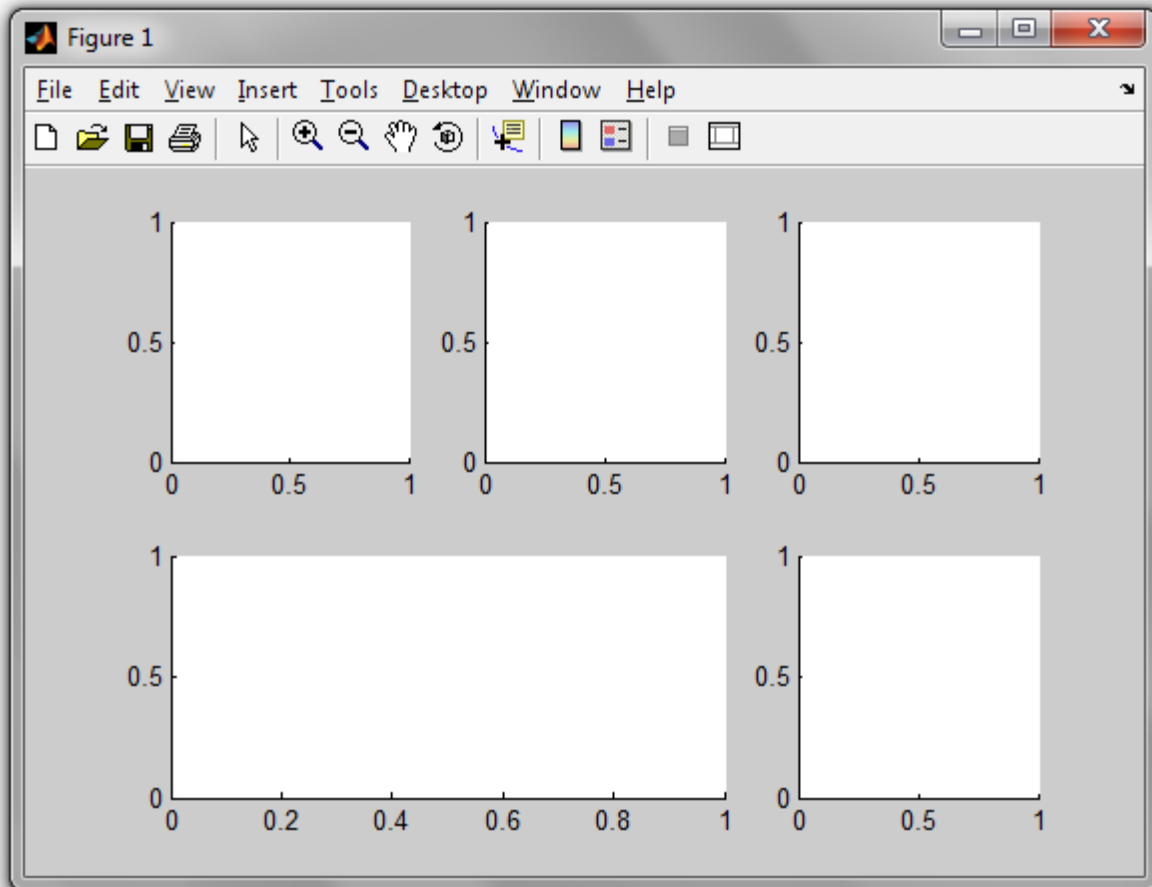
<b>hold on/off</b>	włącza/wyłącza tryb zachowania zawartości okna graficznego
--------------------	--

Jedno okno graficzne można podzielić funkcją **subplot** na kilka części i odwoływać się oddzielnie do każdej z nich.

<b>subplot (n, m, p)</b>	dzieli okno graficzne na mniejsze prostokątne okienka umieszczając je w <b>n</b> -wierszach i <b>m</b> -kolumnach, <b>p</b> - numer aktywnego okienka, w każdym okienku można umieścić odrębny wykres; każde okienko ma własne osie liczbowe; okna graficzne są numerowane od lewej do prawej, wierszami od góry do dołu; funkcja ta służy także do przełączania się pomiędzy wykresami w podzielonym oknie graficznym
--------------------------	--

Jeśli trzeci parametr funkcji **subplot** jest wektorem, to okienka o numerach stanowiących jego elementy łączone są w jedno, większe okno. Sytuację taką pokazuje poniższy przykład.

```
>> subplot(2,3,1)
>> subplot(2,3,2)
>> subplot(2,3,3)
>> subplot(2,3,[4 5])
>> subplot(2,3,6)
```



Rys. 2. Podział okna graficznego na mniejsze okienka

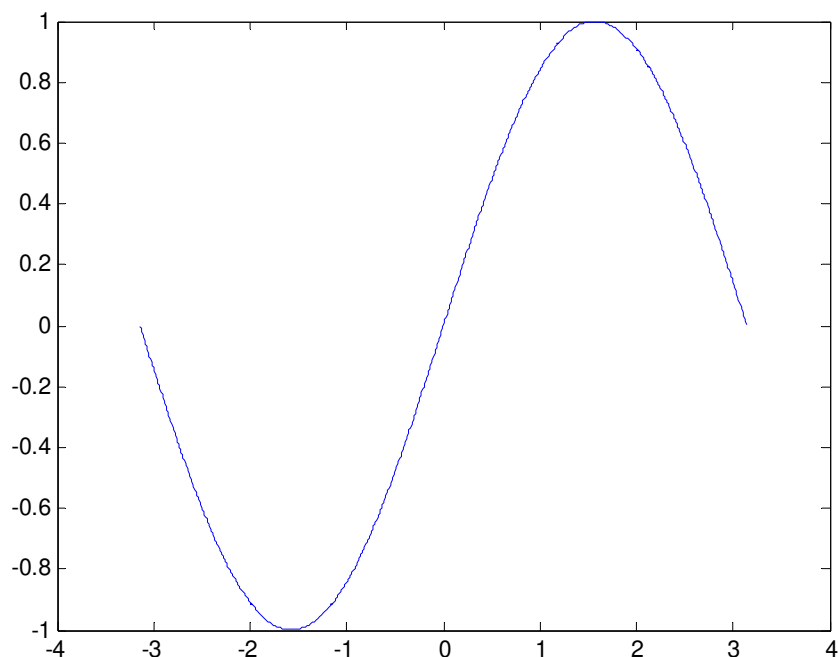
## 2.2. Grafika dwuwymiarowa (2D)

Do wyświetlania grafiki dwuwymiarowej (krzywych) służy funkcja **plot**. W zależności od liczby argumentów wywołanie funkcji może mieć różną postać.

<b>plot(x, y)</b>	rysuje wykres $y = f(x)$ , wektory <b>x</b> i <b>y</b> powinny mieć taką samą długość
<b>plot(y)</b>	rysuje wykres elementów wektora <b>y</b> , przyjmując za <b>x</b> kolejne liczby całkowite począwszy od 1
<b>plot(x, y, s)</b>	rysuje wykres $y = f(x)$ z określeniem sposobu rysowania linii

Poniższy przykład pokazuje sposób narysowania wykresu funkcji **sin(x)**.

```
>> x = -pi:0.01:pi;
>> y1 = sin(x);
>> plot(x, y1)
```



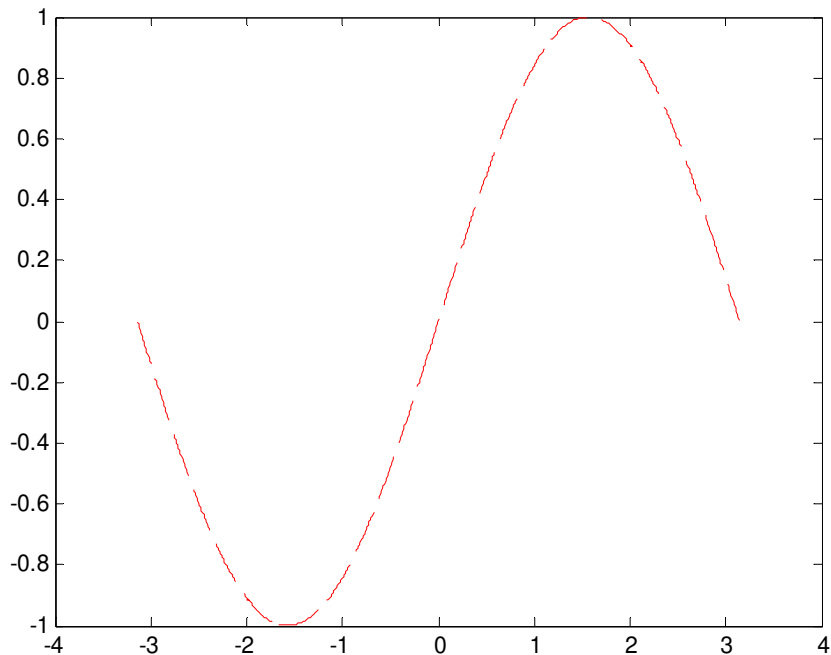
Rys. 3. Wykres funkcji  $\sin(x)$

Domyślnie wykres funkcji rysowany jest linią ciągłą w niebieskim kolorze. Do zmiany sposobu rysowania linii służy argument **s** funkcji **plot**. Jest to łańcuch znaków pozwalający określić: rodzaj linii, jej kolor oraz oznaczenie punktów. Możliwe wartości zestawiono w poniższej tabeli.

Znak	Rodzaj linii	Znak	Kolor linii	Znak	Oznaczenie punktów
-	ciągła (domyślna)	<b>y</b>	żółty	<b>+</b>	krzyżyk
--	kreskowana	<b>m</b>	karmazynowy	<b>*</b>	gwiazdka
:	kropkowana	<b>c</b>	turkusowy	<b>.</b>	kropka
-. .	kreska-kropka	<b>r</b>	czerwony	<b>o</b>	kółko
		<b>g</b>	zielony	<b>x</b>	iks
		<b>b</b>	niebieski	<b>s</b>	kwadrat
		<b>w</b>	biały	<b>d</b>	romb
		<b>k</b>	czarny	<b>p</b>	gwiazdka pięcioramienna
				<b>h</b>	gwiazdka sześcioramienna
				<b>v</b>	trójkąt skierowany do dołu
				<b>^</b>	trójkąt skierowany do góry
				<b>&lt;</b>	trójkąt skierowany w lewo
				<b>&gt;</b>	trójkąt skierowany w prawo

Narysowanie tego samego wykresu linią kreskowaną w czerwonym kolorze będzie miało postać:

```
>> x = -pi:0.01:pi;  
>> y1 = sin(x);  
>> plot(x,y1,'--r')
```



Rys. 4. Wykres funkcji  $\sin(x)$  ze zmianą sposobu rysowania linii

W jednym oknie graficznym może być umieszczonych wiele wykresów. W takim przypadku dane do wykonania tych wykresów podaje się w jednej funkcji **plot**:

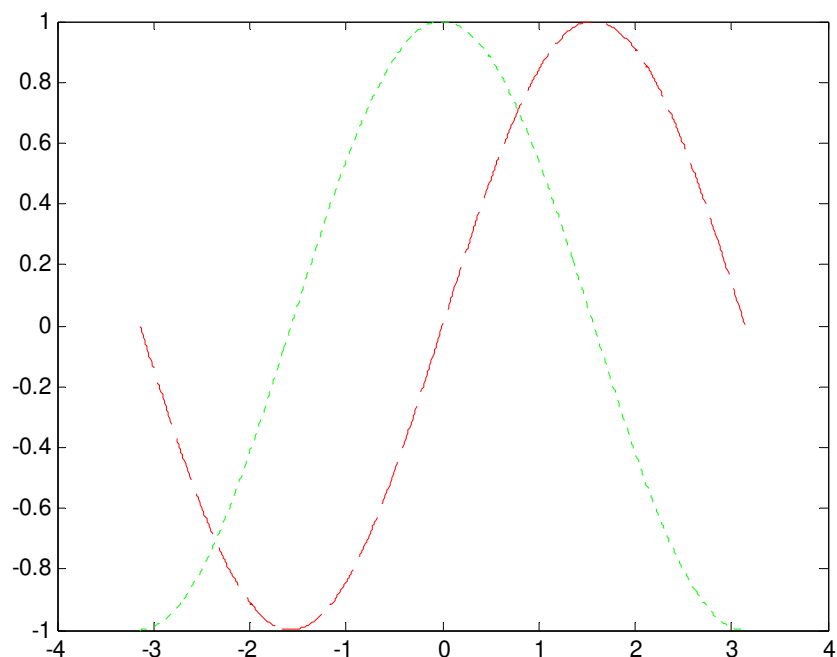
```
plot(x1,y1,x2,y2,...)
```

Dla każdego wykresu można podać oddzielnie sposób rysowania linii:

```
plot(x1,y1,s1,x2,y2,s2,...)
```

Poniższy przykład pokazuje sposób narysowania wykresów funkcji  $\sin(x)$  oraz  $\cos(x)$  w jednym oknie graficznym.

```
>> x = -pi:0.01:pi;  
>> y1 = sin(x);  
>> y2 = cos(x);  
>> plot(x,y1,'--r',x,y2,':g')
```



Rys. 5. Wykresy funkcji **sin(x)** oraz **cos(x)** w jednym oknie graficznym

Na wykresach można umieszczać dodatkowe teksty stanowiące tytuł wykresu, oznaczenia osi, czy też legendę. Odpowiednie funkcje zebrano w poniższej tabeli.

<b>xlabel (tekst)</b>	wyświetla <b>tekst</b> opisujący oś <b>x</b> aktywnego wykresu
<b>ylabel (tekst)</b>	wyświetla <b>tekst</b> opisujący oś <b>y</b> aktywnego wykresu
<b>title (tekst)</b>	wyświetla <b>tekst</b> jako tytuł aktywnego wykresu
<b>text (x, y, tekst)</b>	wyświetla łańcuch znaków <b>tekst</b> w miejscu określonym przez współrzędne <b>x</b> i <b>y</b> , przy czym współrzędne odnoszą się do wartości na aktywnym wykresie
<b>legend (s1, s2, ...)</b>	wyświetla legendę, <b>s1</b> - opis pierwszego wykresu, <b>s2</b> - opis drugiego wykresu, itd.
<b>grid on/off</b>	włącza/wyłącza wyświetlanie na wykresie pomocniczej siatki

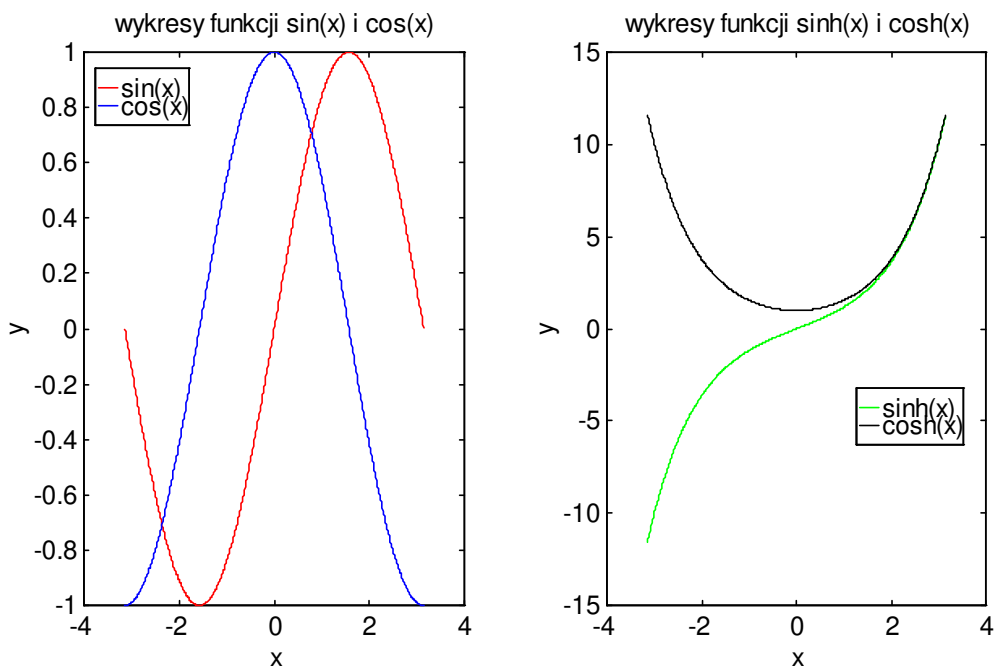
Zastosowanie przedstawionych powyżej funkcji do tworzenia wykresów zawierających grafikę dwuwymiarową oraz do ich opisywania pokazuje poniższy przykład.



```

>> x = -pi:0.01:pi;
>> y1 = sin(x);
>> y2 = cos(x);
>> y3 = sinh(x);
>> y4 = cosh(x);
>> subplot(1,2,1);
>> plot(x,y1,'-r',x,y2,':b');
>> xlabel('x');
>> ylabel('y');
>> legend('sin(x)', 'cos(x)');
>> title('wykresy funkcji sin(x) i cos(x)');
>> subplot(1,2,2);
>> plot(x,y3,'--g',x,y4,'-.k');
>> xlabel('x');
>> ylabel('y');
>> legend('sinh(x)', 'cosh(x)');
>> title('wykresy funkcji sinh(x) i cosh(x)');

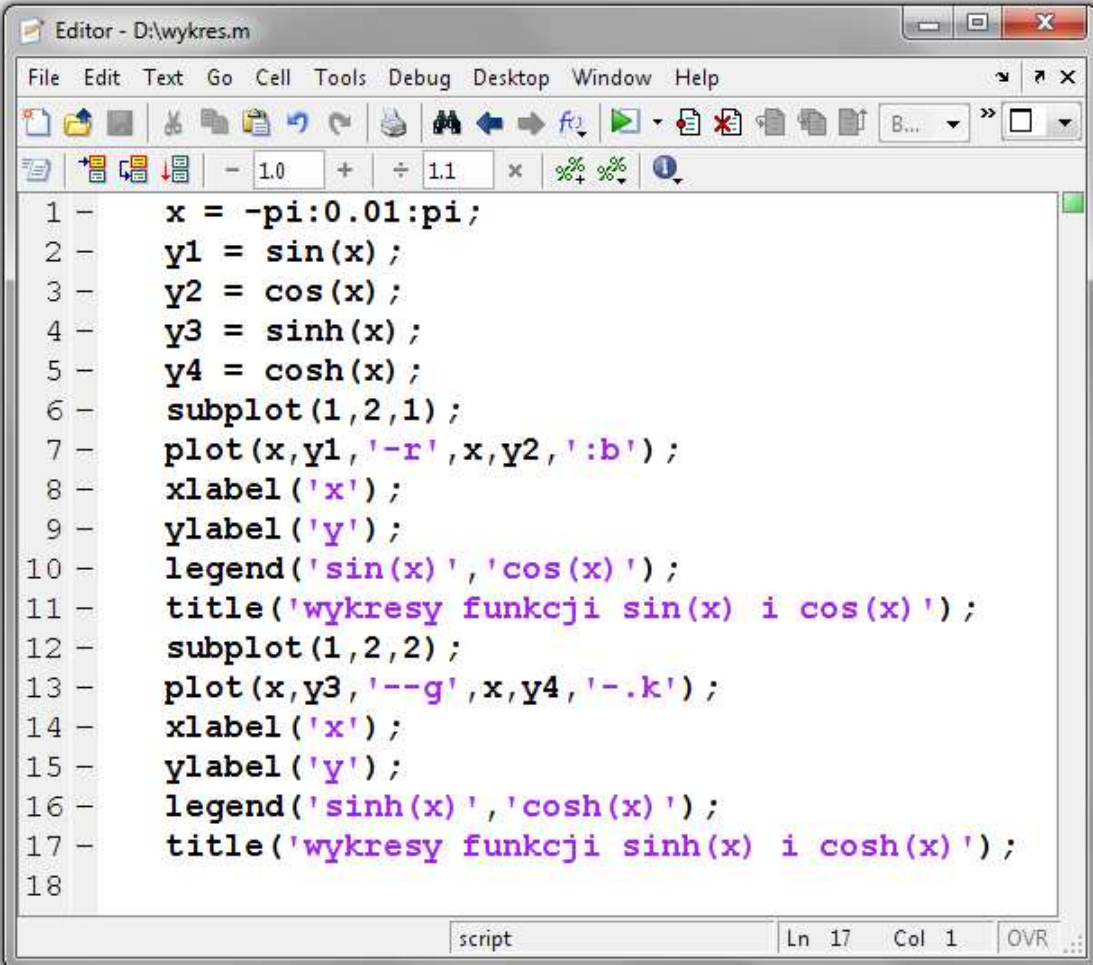
```



Rys. 6. Wykres dwuwymiarowy (2D) - funkcja `plot`

Ze względu na dużą liczbę zastosowanych poleceń oraz możliwość prostej modyfikacji wykresu wygodniej jest umieścić wszystkie instrukcje Matlaba w **skrypcie**. Skrypt jest to plik tekstowy o rozszerzeniu `.m` zawierający polecenia i instrukcje Matlaba. Skrypt można utworzyć w dowolnym edytorze zapisującym

niesformatowane pliki tekstowe. Środowisko Matlab zawiera własny edytorów m-plików wywoływany z menu **Desktop** → **Editor** (Rys. 7).



```
Editor - D:\wykres.m
File Edit Text Go Cell Tools Debug Desktop Window Help
1 - x = -pi:0.01:pi;
2 - y1 = sin(x);
3 - y2 = cos(x);
4 - y3 = sinh(x);
5 - y4 = cosh(x);
6 - subplot(1,2,1);
7 - plot(x,y1,'-r',x,y2,':b');
8 - xlabel('x');
9 - ylabel('y');
10 - legend('sin(x)', 'cos(x)');
11 - title('wykresy funkcji sin(x) i cos(x)');
12 - subplot(1,2,2);
13 - plot(x,y3,'--g',x,y4,'-.k');
14 - xlabel('x');
15 - ylabel('y');
16 - legend('sinh(x)', 'cosh(x)');
17 - title('wykresy funkcji sinh(x) i cosh(x)');
18
script Ln 17 Col 1 OVR
```

Rys. 7. Skrypt w edytorze m-plików

Skrypt uruchamia się podając jego nazwę (bez rozszerzenia) w wierszu poleceń Matlab. Jeśli powyższy skrypt będzie zapisany pod nazwą **wykres.m**, to jego wykonanie wymaga wpisania polecenia:

```
>> wykres
```

Wykonanie skryptu jest równoważne z wpisywaniem i zatwierdzaniem kolejnych jego poleceń w wierszu poleceń (*Command Window*).

Matlab wykona skrypt jeśli będzie on znajdował się w bieżącym katalogu lub w katalogu udostępnionym poleceniem **path**. Bieżący katalog (*Current Directory*) jest to katalog, w którym zapisywane są pliki tworzone podczas pracy. Nazwa

bieżącego katalogu wyświetlana jest w górnej części głównego okna programu Matlab (Rys. 8).



Rys. 8. Okno bieżącego katalogu

Teksty wyświetlane w oknach graficznych przy zastosowaniu funkcji **xlabel**, **ylabel**, **title**, **text** i **legend** mogą być odpowiednio formatowane:

- znak następujący po `_` zostanie wyświetlony jako **indeks dolny**,
- znak następujący po `^` zostanie wyświetlony jako **indeks górny**,
- znaki występujące po `\bf` zostaną wyświetlone **czcionką pogrubioną**,
- znaki występujące po `\it` zostaną wyświetlone **czcionką pochyloną**,
- znaki alfabetu łacińskiego (i inne znaki, których nie da się wprowadzić bezpośrednio z klawiatury) można wyświetlić podając ich nazwy poprzedzone znakiem ukośnika, np. `\alpha` →  $\alpha$ , `\beta` →  $\beta$ , `\infty` →  $\infty$ .

Jeśli indeks dolny lub górny ma obejmować więcej niż jeden znak, to wszystkie te znaki należy objąć nawiasami klamrowymi, np. `x=e^{\-\omega t}` →  $x=e^{-\omega t}$ .

Do przygotowania danych do wykresu można zastosować funkcję **linspace**.

<code><b>linspace</b>(min, max, n)</code>	generuje wektor <b>n</b> liczb rozłożonych równomiernie w przedziale od <b>min</b> do <b>max</b>
<code><b>linspace</b>(min, max)</code>	generuje wektor <b>100</b> liczb rozłożonych równomiernie w przedziale od <b>min</b> do <b>max</b>

W poniższym przykładzie tworzony jest wektor **x** zawierający 5 równomiernie rozłożonych liczb z przedziału od  $-\pi$  do  $\pi$ .

```
>> x = linspace(-pi, pi, 5)
x =
    -3.1416    -1.5708         0         1.5708     3.1416
```

Do narysowania wykresu dowolnej funkcji przydatne może być zastosowanie polecenia **fplot**, które przygotowuje wektory **x** i **y** dla funkcji **plot(x,y)**. **fplot** automatycznie dobiera liczbę argumentów funkcji z podanego zakresu.

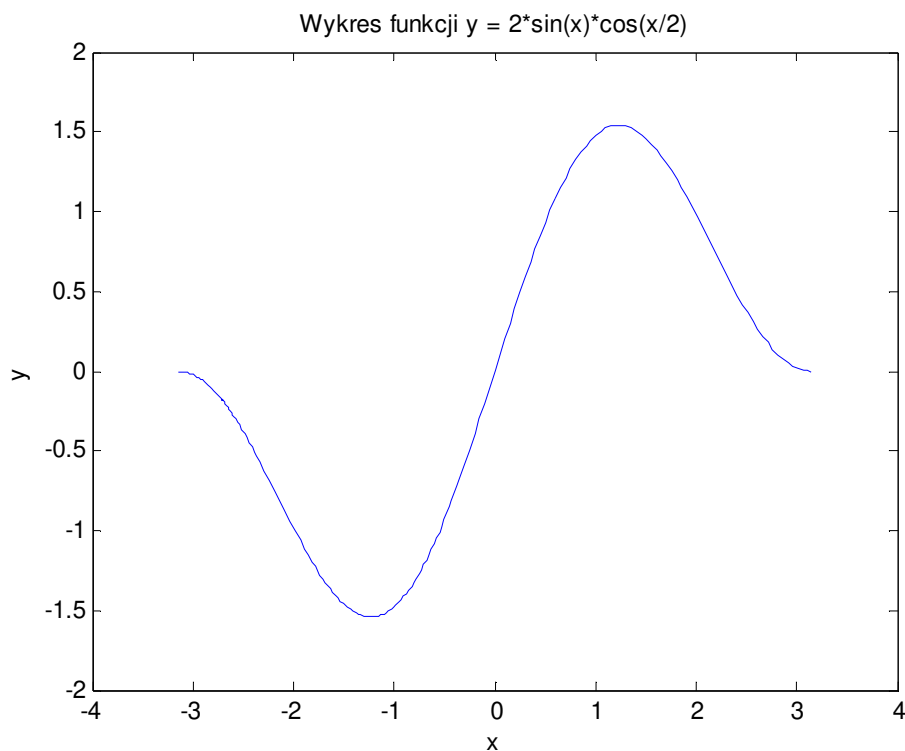
<b>fplot(fun, [x1 x2])</b>	przygotowuje wektory <b>x</b> i <b>y</b> do narysowania wykresu funkcji opisanej przez <b>fun</b> ; <b>fun</b> jest łańcuchem znaków zawierającym nazwę funkcji, zaś <b>x1</b> i <b>x2</b> to granice przedziału argumentów funkcji
----------------------------	---

Wywołanie funkcji **fplot** jest następujące:

```
>> [x,y] = fplot(fun, [x1 x2])
```

Pokazuje to poniższy przykład.

```
>> [x,y]=fplot('2*sin(x)*cos(x/2)', [-pi pi]);  
>> plot(x,y);  
>> xlabel('x');  
>> ylabel('y');  
>> title('Wykres funkcji y = 2*sin(x)*cos(x/2)')
```



Rys. 9. Wykres dwuwymiarowy - dane do wykresy przygotowane funkcją **fplot**

Do wykresów w **skali logarytmicznej** służą w Matlabie oddzielne funkcje.

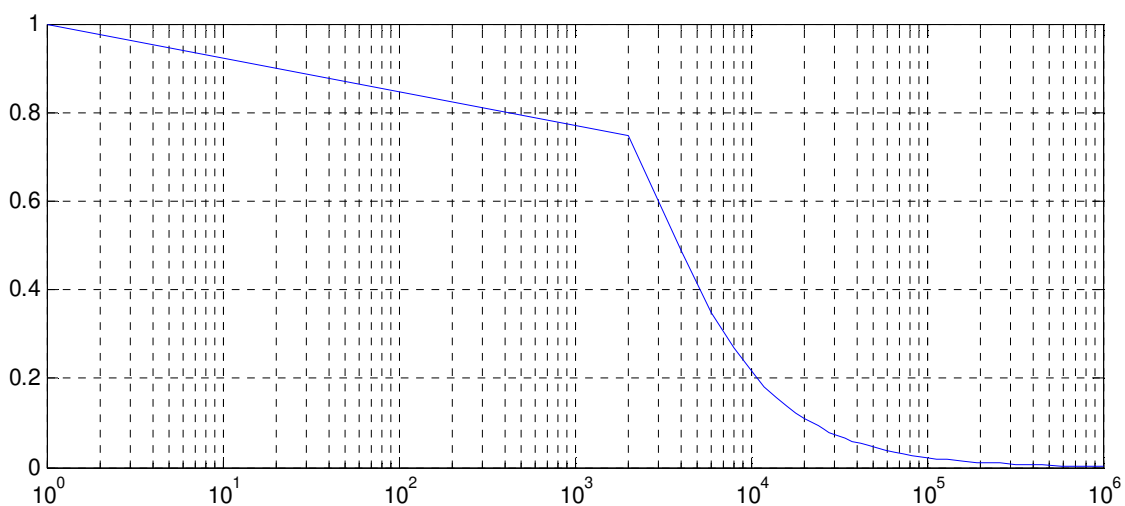
<b>loglog(x, y, s)</b>	rysuje wykres $y = f(x)$ przy zastosowaniu skali logarytmicznej na obu osiach, wektory $x$ i $y$ powinny mieć taką samą długość, $s$ jest łańcuchem znaków opisującym wygląd linii
<b>semilogx(x, y, s)</b>	rysuje wykres $y = f(x)$ przy zastosowaniu skali logarytmicznej tylko na osi $x$
<b>semilogy(x, y, s)</b>	rysuje wykres $y = f(x)$ przy zastosowaniu skali logarytmicznej tylko na osi $y$

Dane do wykresu w skali logarytmicznej mogą być utworzone przy zastosowaniu funkcji **logspace**.

<b>logspace(min, max, n)</b>	generuje wektor $n$ liczb rozłożonych równomiernie w przedziale od $min$ do $max$
------------------------------	---

W poniższym przykładzie zastosowano skalę logarytmiczną tylko na osi  $x$ .

```
>> [x,y] = fplot('1/sqrt(1+2e-7*x^2)', [1, 1e6]);
>> semilogx(x, y)
>> grid on
```



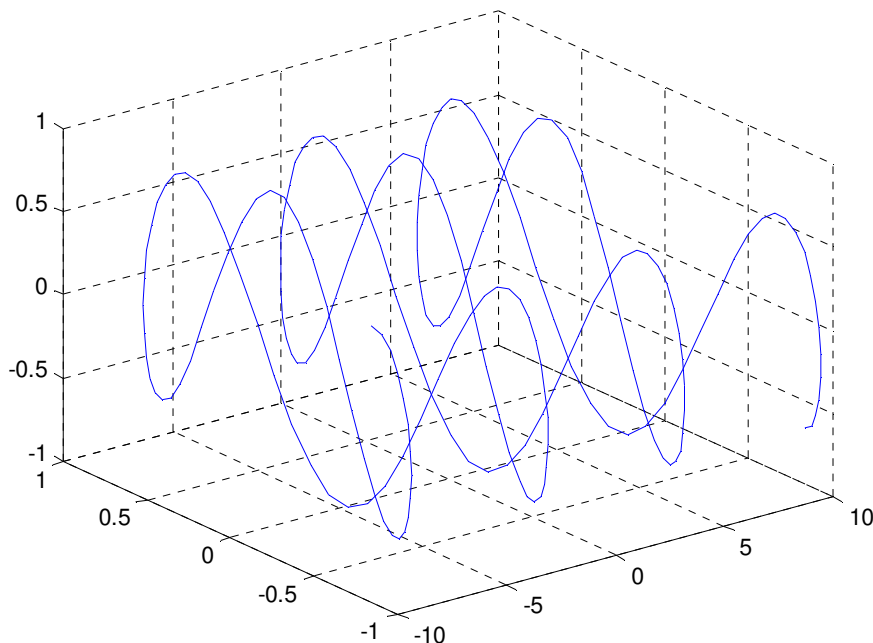
Rys. 10. Wykres dwuwymiarowy ze skalą logarytmiczną na osi  $x$

## 2.3. Grafika trójwymiarowa (3D)

Narysowanie wykresu funkcji trzech zmiennych (krzywej) umożliwia funkcja **plot3**.

<b>plot3(x,y,z,s)</b>	rysuje wykres funkcji trzech zmiennych; <b>x, y, z</b> - wektory o jednakowej długości określające współrzędne punktów, <b>s</b> - łańcuch znaków opisujący wygląd linii
-----------------------	--

```
>> x = [-10:0.1:10];  
>> y = cos(x);  
>> z = sin(3*x) .* cos(y);  
>> plot3(x,y,z)  
>> grid on
```



Rys. 11. Wykres funkcji trzech zmiennych

Narysowanie **wykresu powierzchniowego** (czyli wykresu  $z = f(x,y)$ ) jest nieco bardziej skomplikowane. Pierwszym krokiem jest wygenerowanie specjalnej siatki na płaszczyźnie **XY** w węzłach której określone będą wartości funkcji w osi **z**. Służy do tego funkcja **meshgrid** o składni:

```
[x,y] = meshgrid(X,Y)
```

gdzie: **X, Y** - wektory zawierające wartości w punktach ograniczających płaszczyznę **XY**, **x, y** - macierze określające współrzędne, dla których będą liczone wartości **z**.

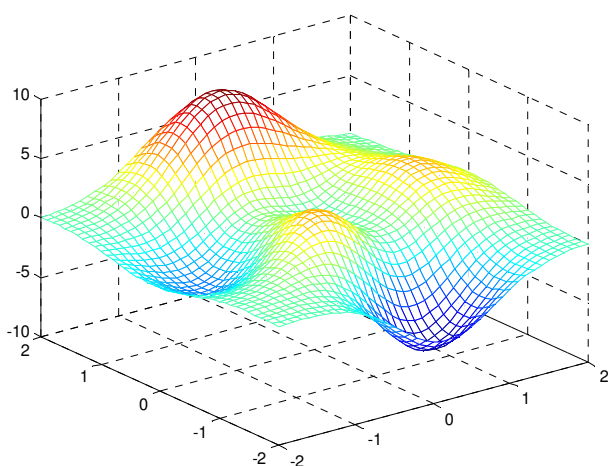
Po przygotowaniu siatki opisanej macierzami  $x$  i  $y$  należy obliczyć wartości macierzy  $z$ . Do narysowania wykresu powierzchniowego można wykorzystać jedną z przedstawionych poniżej funkcji. Powstały wykres składa się z czworokątów, których wierzchołki leżą w punktach o współrzędnych opisanych macierzami  $x$ ,  $y$ ,  $z$ .

<b>mesh (x, y, z, c)</b>	rysuje powierzchnię w postaci kolorowej siatki o polach wypełnionych kolorem tła, elementy macierzy $c$ określają kolory linii poszczególnych pól
<b>mesh (x, y, z)</b>	rysuje powierzchnię przy $c = z$
<b>mesh (z, c)</b>	rysuje wykres wartości elementów macierzy $z$ przyjmując na osiach $x$ i $y$ kolejne liczby całkowite począwszy od 1
<b>mesh (z)</b>	rysuje wykres wartości elementów macierzy $z$ przyjmując na osiach $x$ i $y$ kolejne liczby całkowite począwszy od 1 oraz przyjmując $c = z$
<b>meshc (x, y, z, c)</b>	działa jak <b>mesh</b> , ale dodatkowo umieszcza pod siatką wykres poziomicowy
<b>meshz (x, y, z, c)</b>	działa jak <b>mesh</b> , ale dodatkowo rysuje w dół od krawędzi wykresu dodatkowe linie określające płaszczyzny odniesienia
<b>surf (x, y, z, c)</b>	rysuje różnokolorową powierzchnię
<b>surf (x, y, z)</b>	rysuje różnokolorową powierzchnię przyjmując $c = z$
<b>surf (z, c)</b>	rysuje różnokolorową powierzchnię dla wartości elementów macierzy $z$ przyjmując na osiach $x$ i $y$ kolejne liczby całkowite począwszy od 1
<b>surf (z)</b>	rysuje różnokolorową powierzchnię dla wartości elementów macierzy $z$ przyjmując na osiach $x$ i $y$ kolejne liczby całkowite począwszy od 1 oraz przyjmując $c = z$
<b>surfc (x, y, z, c)</b>	działa jak <b>surf</b> , ale dodatkowo umieszcza pod różnokolorową powierzchnią wykres poziomicowy
<b>surf1 (x, y, z, c)</b>	rysuje powierzchnię z uwzględnieniem odbić światła
<b>contour (x, y, z)</b>	rysuje wykres konturowy (poziomicowy)

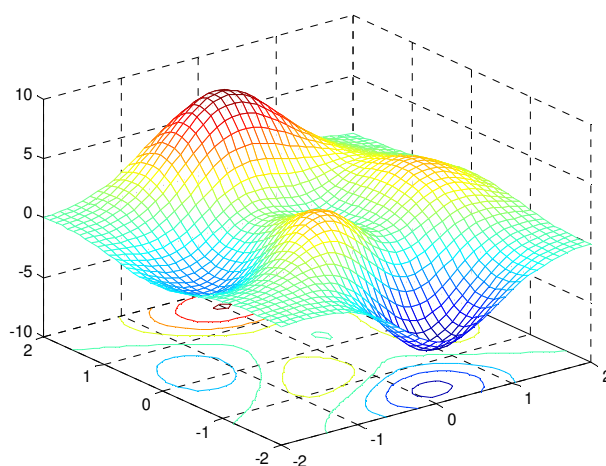
<b>contour(z)</b>	rysuje wykres konturowy (poziomicowy) wartości elementów macierzy <b>z</b> przyjmując na osiach <b>x</b> i <b>y</b> kolejne liczby całkowite począwszy od 1
<b>contour(z,n)</b>	rysuje wykres konturowy (poziomicowy) wartości elementów macierzy <b>z</b> stosując <b>n</b> linii poziomic
<b>contourf(z)</b>	działa jak <b>contour</b> , ale obszar pomiędzy poziomiami jest wypełniony kolorem zależnym od wartości w macierzy <b>z</b>

Poniżej przedstawiono przykłady różnych typów wykresów powierzchniowych.

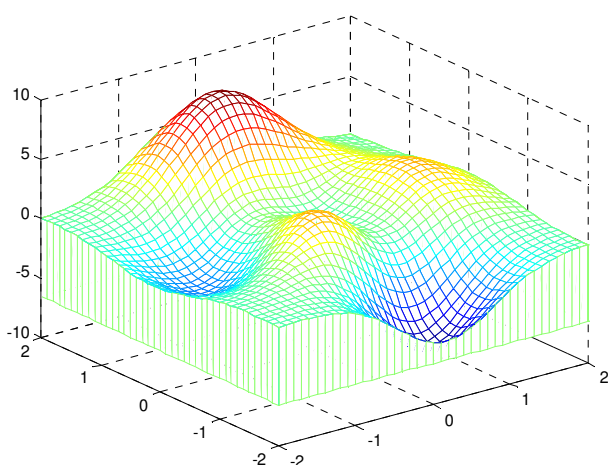
```
>> [x,y] = meshgrid(-2:0.1:2,-2:0.1:2);
>> z = peaks(x,y);
```



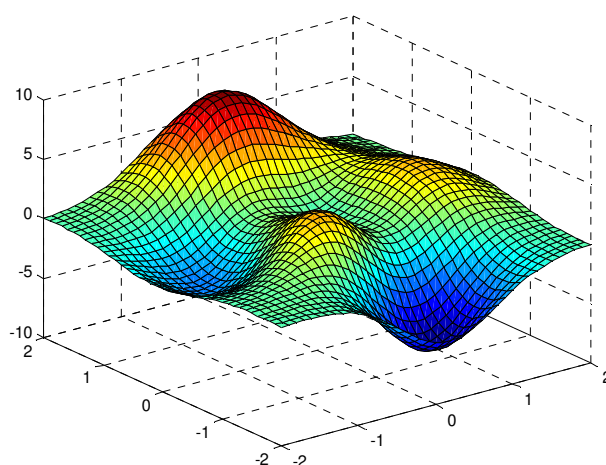
```
>> mesh(x,y,z)
```



```
>> meshc(x,y,z)
```

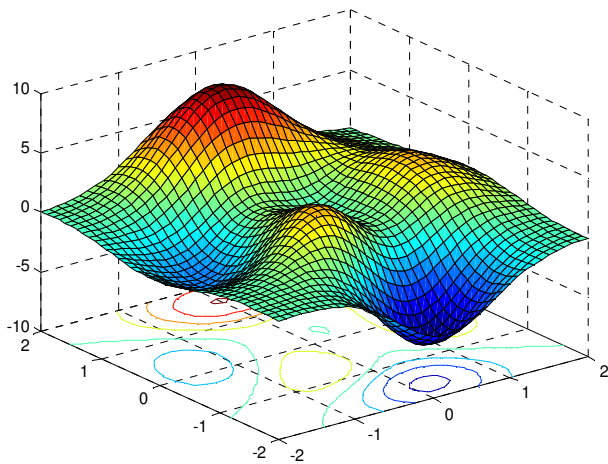


```
<< meshz(x,y,z)
```

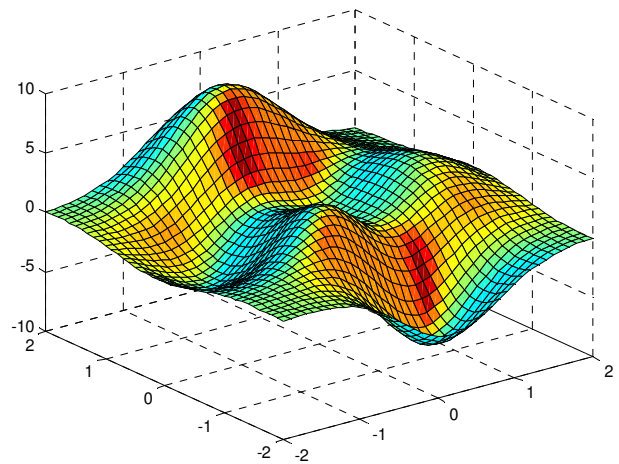


```
<< surf(x,y,z)
```



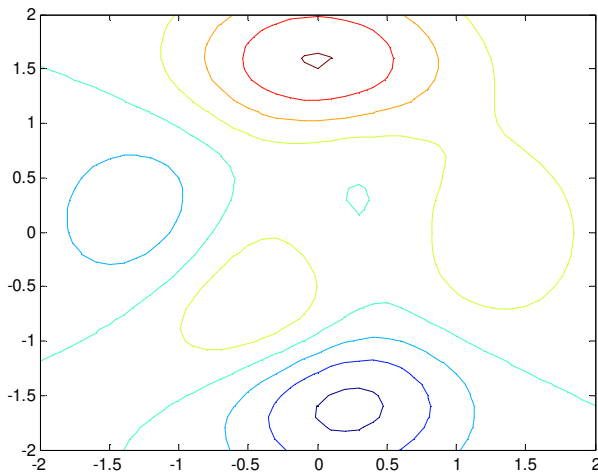


`<< surfc(x,y,z)`

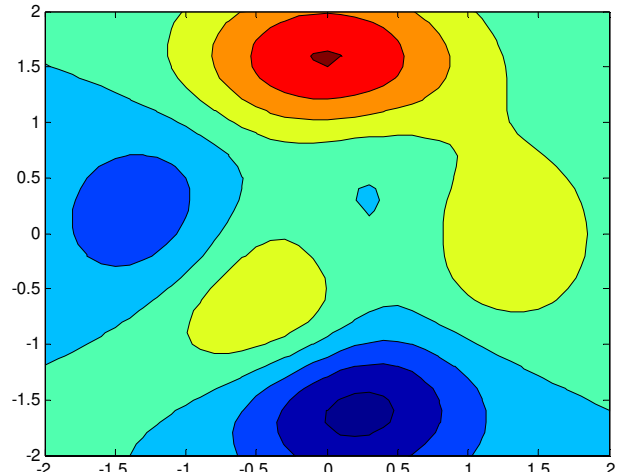


`<< surf1(x,y,z)`

Rys. 12. Różne typy wykresów powierzchniowych



`<< contour(x,y,z)`

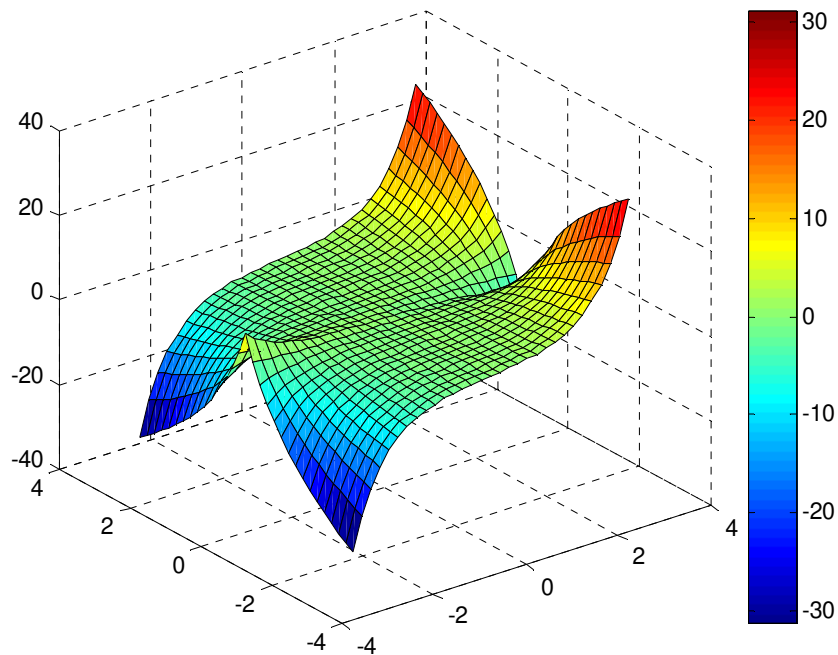


`<< contourf(x,y,z)`

Rys. 13. Różne typy wykresów konturowych

Do wykresu można dodać skalę kolorów poprzez wywołanie polecenia **colorbar**.

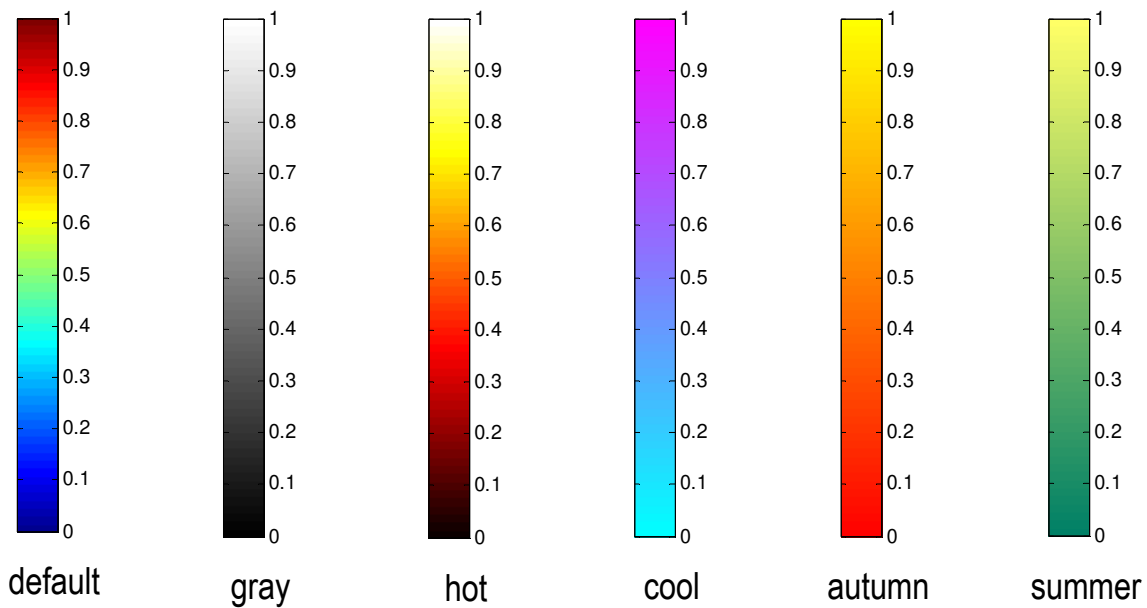
```
>> [x,y] = meshgrid(-3:0.2:3,-3:0.2:3);
>> z = sinh(x) .* cosh(x/2) .* log(abs(y)+0.75);
>> surf(x,y,z);
>> colorbar
```



Rys. 14. Wykres powierzchniowy ze skalą kolorów

Kolory na wykresie można zmodyfikować poprzez zmianę **mapy kolorów**. Mapa kolorów jest macierzą trójkolumnową, której elementami są liczby z zakresu 0-1. Kolumny odpowiadają intensywności każdego z trzech podstawowych kolorów RGB: czerwonego, zielonego i niebieskiego. Aktualną mapę kolorów zmienia polecenie **colormap(m)**, gdzie **m** jest nową mapą. W Matlabie zdefiniowane są następujące mapy:

- **gray** - mapa odcieni szarości,
- **hot** - kolory ciepłe (od czarnego przez czerwony, pomarańczowy, żółty do białego),
- **cool** - kolory zimne (od turkusowego do karmazynowego),
- **autumn** - kolory od czerwonego przez pomarańczowy do żółtego,
- **summer** - odcienie żółtego i zielonego.



Rys. 15. Wybrane mapy kolorów

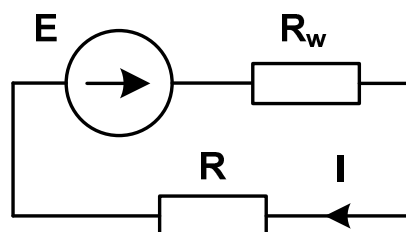
Domyślną mapę kolorów przywraca polecenie:

```
>> colormap('default')
```

### 3. Przebieg ćwiczenia

Wykonaj podane poniżej zadania.

1. Odbiornik  $R$  jest zasilany ze źródła napięcia stałego  $E$  o rezystancji wewnętrznej  $R_w$ . Narysuj zależność mocy  $P$  wydzielanej na odbiorniku od wartości jego rezystancji  $R$ . Na podstawie wykresu określ dla jakiej wartości rezystancji  $R$  następuje dopasowanie odbiornika do źródła. Przyjmij:  $R_w = 100 \Omega$ ,  $E = 10 \text{ V}$ ,  $R \in \langle 0, 250 \rangle$ . Opisz osie wykresu, dodaj legendę i tytuł, włącz wyświetlanie pomocniczej siatki.



2. Napisz skrypt wyświetlający na jednym wykresie przebiegi trzech funkcji:

- **cos(x)** - kolor czerwony, linia kropkowana,
- **cos<sup>2</sup>(x)** - kolor niebieski, linia ciągła,
- **cos<sup>3</sup>(x)** - kolor zielony, linia kreskowana,

w przedziale  $\langle 0, 2\pi \rangle$ . Opisz osie wykresu, dodaj legendę i tytuł, włącz wyświetlanie pomocniczej siatki.

3. Napisz skrypt wyświetlający w jednym, podzielonym, oknie graficznym dwa wykresy funkcji  $y = \cos(10 \cdot e^x)$  w przedziale  $\langle -2, 2 \rangle$ . Wykresy umieść jeden pod drugim. Do przygotowania danych do wykonania pierwszego wykresu zastosuj funkcję **fplot**, natomiast do drugiego - **100** równomiernie rozłożonych punktów w podanym przedziale. Oba wykresy narysuj funkcją **plot**. Opisz osie wykresów, dodaj legendę i tytuły.

fplot, plot
100 pkt, plot

Opisz różnice pomiędzy wykresami. Podaj z czego one wynikają (sprawdź ile punktów wygenerowała funkcja **fplot**, dodaj do wykresów wyświetlanie punktów)?

4. Napisz skrypt wyświetlający w jednym, podzielonym, oknie graficznym cztery wykresy funkcji (dwa w wierszu i dwa w kolumnie)  $z = \sin(x) \cdot \cos(y)$  dla  $x, y \in \langle -\pi, 2\pi \rangle$  z krokiem **0,3**. Do narysowania wykresów zastosuj funkcje: **mesh**, **meshc**, **surf**, **surfc**. Dodaj opisy do wykresów.

## 4. Literatura

- [1] Mrozek B., Mrozek Z.: MATLAB i Simulink. Poradnik użytkownika. Wydanie III. Helion, Gliwice, 2012.
- [2] Stachurski M. Treichel W.: Matlab dla studentów. Ćwiczenia, zadania, rozwiązania. Witkom, Warszawa, 2009.

- [3] Pratap R.: MATLAB dla naukowców i inżynierów. Wydanie 2. Wydawnictwo Naukowe PWN, Warszawa, 2015.
- [4] Brzóska J., Dorobczyński L.: Matlab: środowisko obliczeń naukowo-technicznych. „Mikom”, Wydawnictwo Naukowe PWN, Warszawa, 2008.
- [5] Kamińska A., Pańczyk B.: Ćwiczenia z Matlab. Przykłady i zadania. Wydawnictwo MIKOM, Warszawa, 2002.
- [6] Sobierajski M., Łabuzek M.: Programowanie w Matlabie dla elektryków. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005.
- [7] Dyka E., Markiewicz P., Sikora R.: Modelowanie w elektrotechnice z wykorzystaniem środowiska MATLAB. Wydawnictwa Politechniki Łódzkiej, Łódź, 2006.
- [8] Sradomski W.: Matlab. Praktyczny podręcznik modelowania. Helion, Gliwice, 2015.
- [9] Czajka M.: MATLAB. Ćwiczenia. Helion, Gliwice, 2005.

## **5. Zagadnienia na zaliczenie**

1. Omów sposób tworzenia w Matlabie wykresów funkcji dwu zmiennych.
2. Omów sposób tworzenia w Matlabie wykresów funkcji trzech zmiennych.
3. Omów sposób tworzenia w Matlabie wykresów powierzchniowych.

## **6. Wymagania BHP**

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciwpożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.

- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.