



Wydział Elektryczny
Katedra Elektrotechniki Teoretycznej i Metrologii

Instrukcja do pracowni z przedmiotu Podstawy Informatyki

Kod przedmiotu: TS1C 100 003

Ćwiczenie pt.

ŚRODOWISKO MATLAB cz.4

Tworzenie wykresów funkcji

Numer ćwiczenia

PI EiT 14

Autor

Dr inż. Jarosław Forenc

Białystok 2012

1. OPIS STANOWISKA

1.1. Stosowana aparatura

- komputer klasy PC z systemem operacyjnym Microsoft Windows (XP/Vista/7).

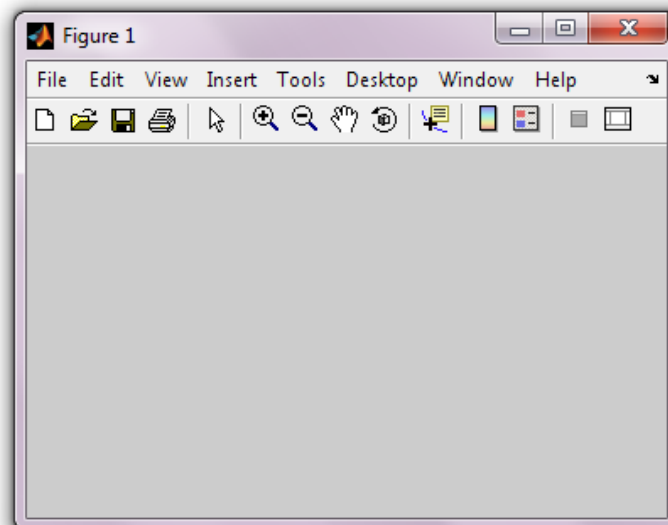
1.2. Oprogramowanie

- 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. Każde okno graficzne ma unikalny numer wyświetlany w jego nagłówku, np. **Figure 1**, **Figure 2**.



Do obsługi okien graficznych stosowane są poniższe funkcje.

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

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

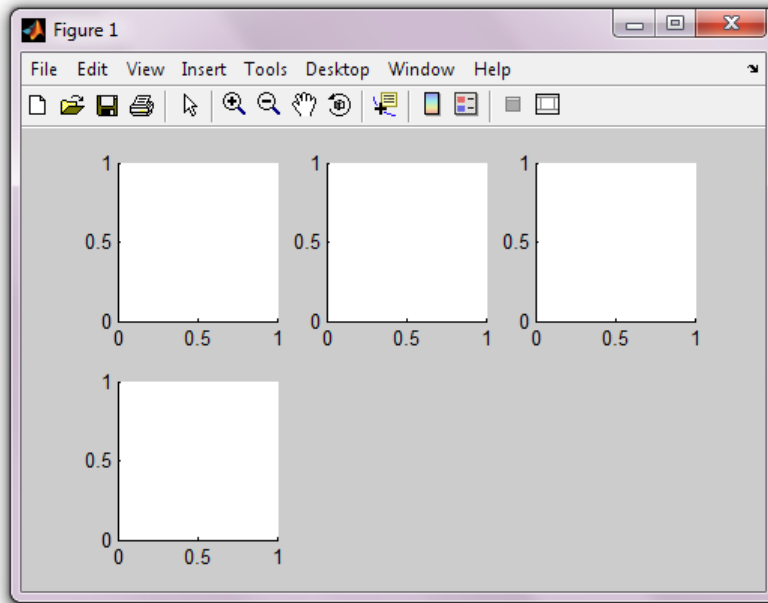
subplot (n, m, p)	dzieli okno graficzne na mniejsze prostokątne okienka umieszczając je w n -wierszach i m -kolumnach, p - numer aktywnego okienka, w każdym okienku można umieścić odrębny wykres; okna graficzne są numerowane od lewej do prawej, wierszami od góry do dołu; funkcja ta służy także do przełączania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

```
>> subplot (2, 3, 1)
```

```
>> subplot (2, 3, 2)
```

```
>> subplot (2, 3, 3)
```

```
>> subplot (2, 3, 4)
```



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

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

2.2. Grafika dwuwymiarowa (2D)

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

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

W jednym oknie 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, . . .)`

Łańcuch znaków (**s**) opisujący sposób rysowania linii może określać: rodzaj linii, jej kolor oraz oznaczenie punktów. Możliwe wartości zestawiono w poniższej tabeli.

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

Do przygotowania danych do wykresu przydatne może być zastosowanie funkcji **linspace**.

linspace (min, max, n)	generuje wektor n liczb rozłożonych równomiernie w przedziale od min do max
linspace (min, max)	generuje wektor 100 liczb rozłożonych równomiernie w przedziale od min do max

W poniższym przykładzie tworzony jest wektor x zawierający 7 równomiernie rozłożonych liczb z przedziału od $-\pi$ do π .

```
>> x = linspace(-pi, pi, 7)

x =

    3.1416    2.0944    1.0472         0    -1.0472    -2.0944    -3.1416
```

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

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

Zastosowanie przedstawionych powyżej funkcji do tworzenia wykresów zawierających grafikę dwuwymiarową oraz do ich opisywania pokazuje poniższy przykład. Ze względu na dużą liczbę funkcji oraz możliwość prostej modyfikacji wykresu, zaleca się umieszczenie wszystkich instrukcji w skrypcie.

```
x = -pi:0.01:pi;

y1 = sin(x);

y2 = cos(x);

y3 = sinh(x);

y4 = cosh(x);

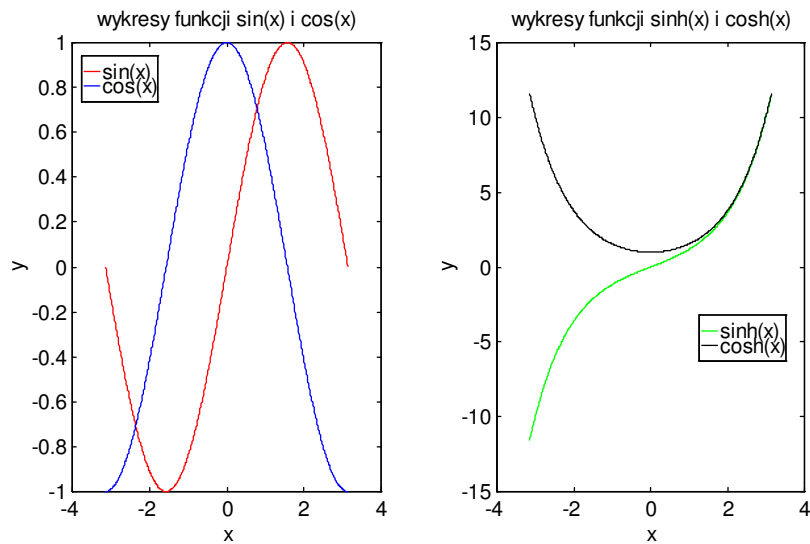
subplot(1, 2, 1);

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)');

```



Teksty wyświetlane funkcjami **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 **lit** 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** → α , **\beta**, **\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 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.

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

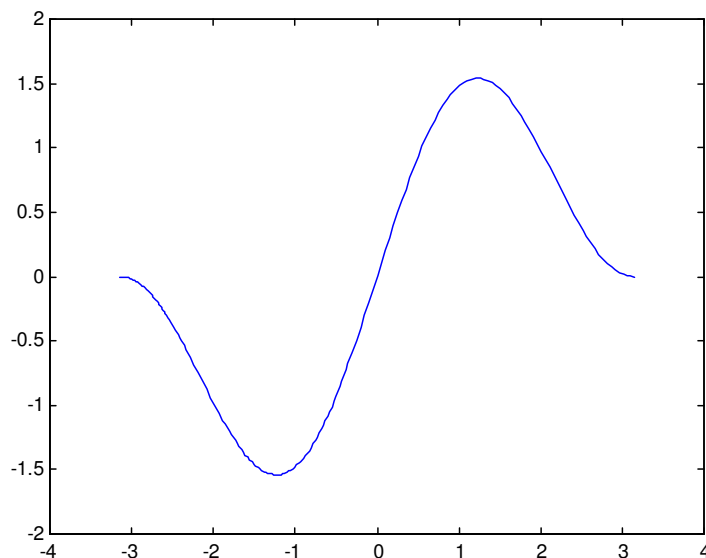
Wywołanie funkcji **fplot** jest następujące:

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

Pokazuje to poniższy przykład.

```
>> [x,y]=fplot ('2*sin(x)*cos(x/2)', [-pi pi]);
```

```
>> plot (x,y);
```



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

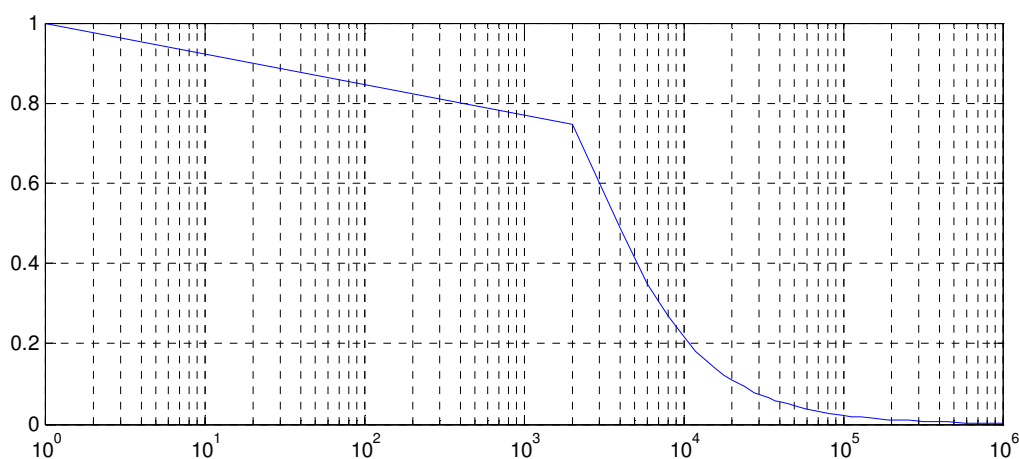
loglog(x, y, s)	rysuje wykres $y = f(x)$ przy zastosowaniu skali logarytmicznej na obu osiach, wektory x i y powinny mieć taką samą długość, s jest łańcuchem znaków opisującym wygląd linii
semilogx(x, y, s)	rysuje wykres $y = f(x)$ przy zastosowaniu skali logarytmicznej tylko na osi x
semilogy(x, y,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

logspace(min, max, n)	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
```

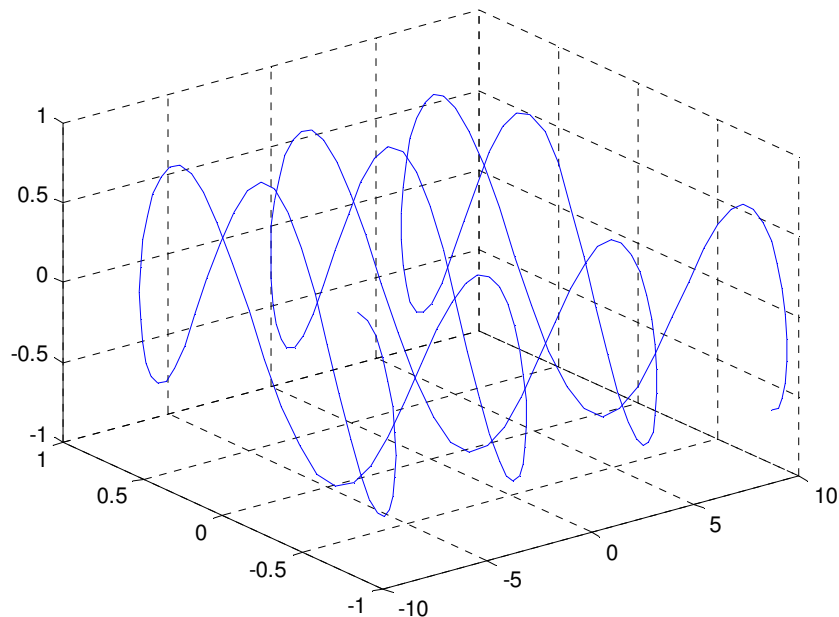


2.3. Grafika trójwymiarowa (3D)

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

<code>plot3(x,y,z,s)</code>	rysuje wykres funkcji trzech zmiennych; x, y, z - wektory o jednakowej długości określające współrzędne punktów, s - 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
```



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

Po przygotowaniu siatki opisanej macierzami \mathbf{x} i \mathbf{y} należy obliczyć wartości macierzy \mathbf{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 $\mathbf{x}, \mathbf{y}, \mathbf{z}$.

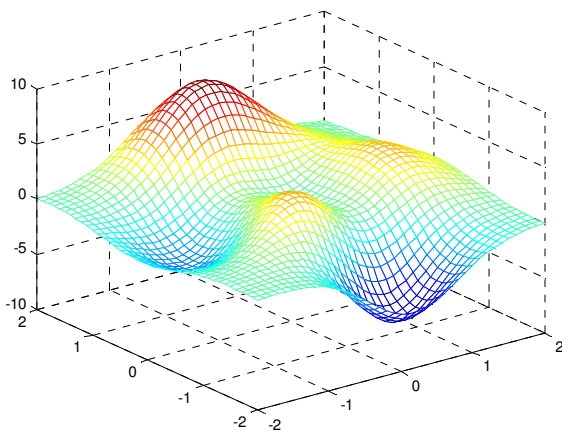
mesh (x, y, z, c)	rysuje powierzchnię w postaci kolorowej siatki o polach wypełnionych kolorem tła, elementy macierzy c określają kolory linii poszczególnych pól
mesh (x, y, z)	rysuje powierzchnię przy c = z
mesh (z, c)	rysuje wykres wartości elementów macierzy z przyjmując na osiach x i y kolejne liczby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

<code>surf1(x,y,z,c)</code>	rysuje powierzchnię z uwzględnieniem odbić światła
-----------------------------	----------------------------------------------------

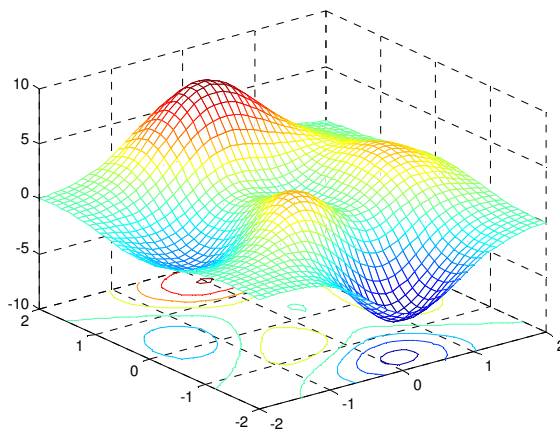
Poniżej przedstawiono przykłady różnych typów wykresów powierzchniowych.

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

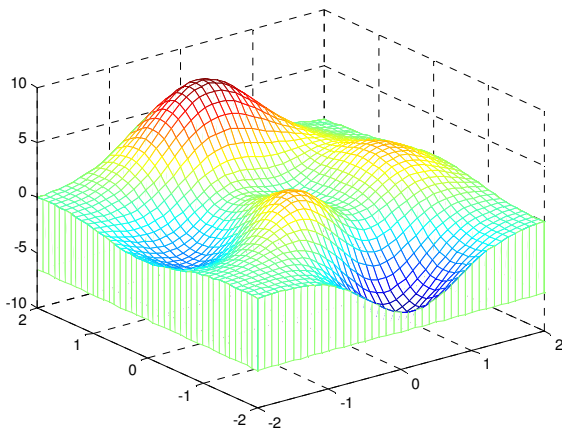
```
>> z = peaks(x,y);
```



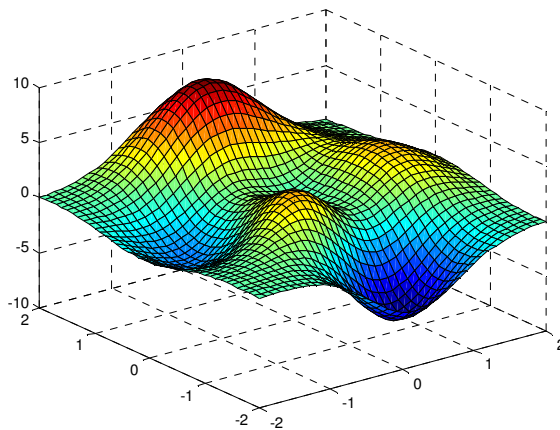
```
>> mesh(x,y,z)
```



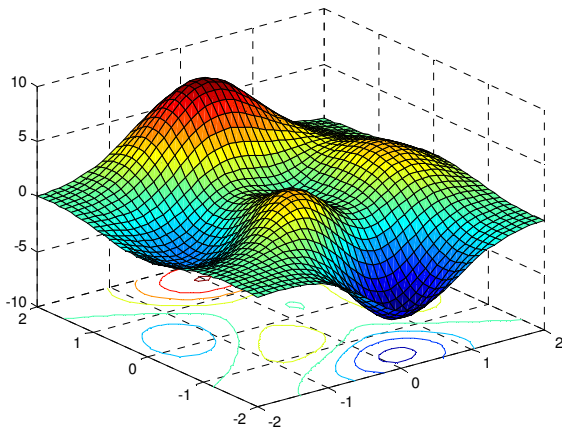
```
>> meshc(x,y,z)
```



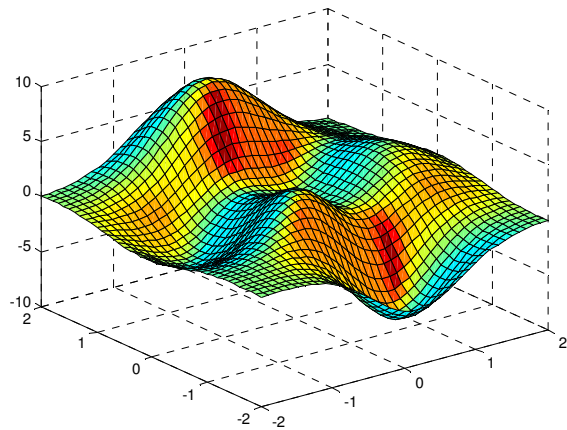
```
<< meshz(x,y,z)
```



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



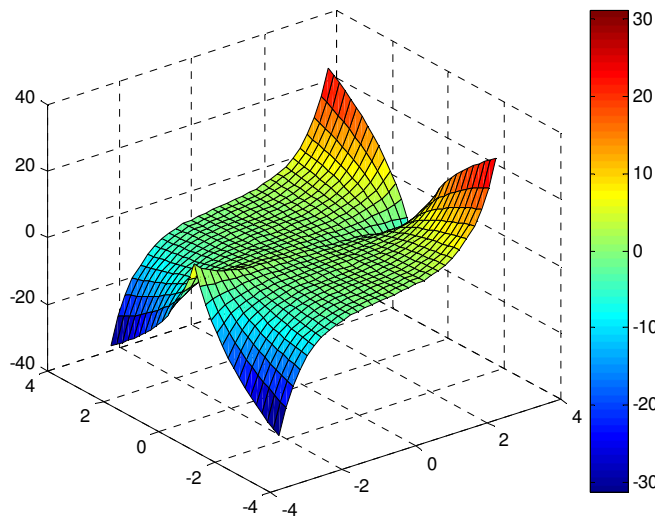
`<< surfc(x,y,z)`



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

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

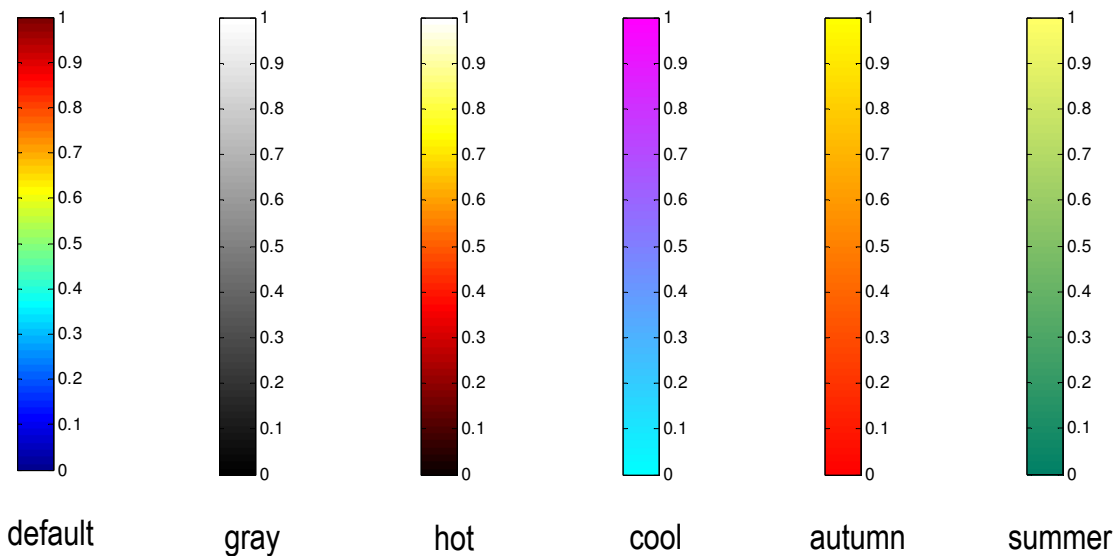
```
>> [x,y] = meshgrid(-3:0.2:3,-3:0.2:3);
>> z = sinh(x) .* cosh(x/2) .* log(abs(y)+0.75);
>> surf(x,y,z);
>> colorbar
```



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.



Domyślną mapę kolorów przywraca polecenie:

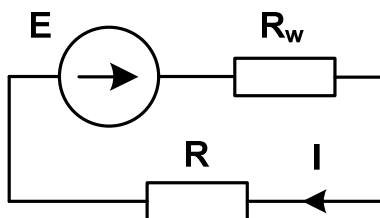
```
>> colormap('default')
```

3. PRZEBIEG ĆWICZENIA

Wykonaj podane poniżej zadania.

Zadanie 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.



Zadanie 2

Napisz skrypt wyświetlający na jednym wykresie przebiegi trzech funkcji:

- $\cos(x)$ - kolor czerwony, linia kropkowana,
- $\cos^2(x)$ - kolor niebieski, linia ciągła,
- $\cos^3(x)$ - kolor zielony, linia kreskowana,

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

Zadanie 3

Napisz skrypt wyświetlający w jednym, podzielonym, oknie graficznym dwa wykres 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)?

Zadanie 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, 2010.
- [2] Stachurski M. Treichel W.: Matlab dla studentów. Ćwiczenia, zadania, rozwiązania. Witkom, Warszawa, 2009.
- [3] Pratap R.: MATLAB 7 dla naukowców i inżynierów. Wydawnictwo Naukowe PWN, Warszawa, 2010.
- [4] Brzoska 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] 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.