



# Wykład 4

## Grafika 3D



Lista obecności 😊

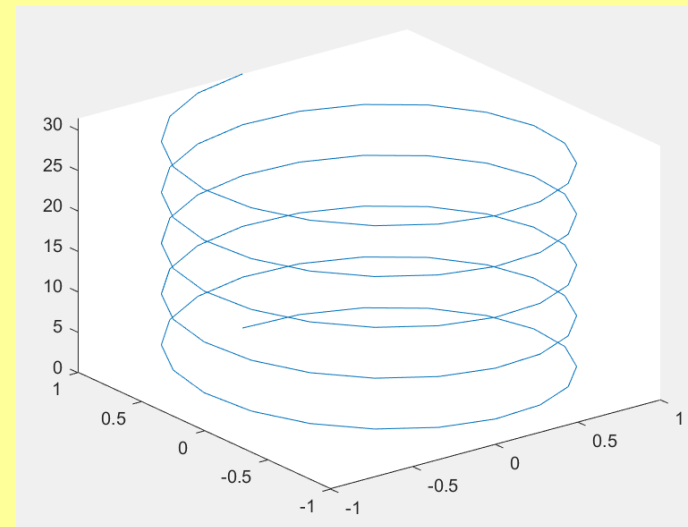
## Wykresy funkcji

Większość poleceń związanych z obsługą okna graficznego, używanych przy budowaniu wykresów dwuwymiarowych, stosuje się także w przypadku wykresów trójwymiarowych, np. `figure`, `subplot`, `axis`, `grid`, itp.

Również wiele funkcji 2D ma swoje odpowiedniki dla wykresów 3D, np. polecenie **plot3** działa analogicznie do **plot** (pojawia się tylko dodatkowa zmienna  $z$ ). Wygląd linii (kolor, styl, znaczniki) opisywany jest tak samo jak w przypadku polecenia **plot**.

**plot3** ( $x, y, z, s$ ) - rysuje linię opisaną punktami o współrzędnych  $x, y, z$ ; parametr  $s$  decyduje o wyglądzie linii (kolor, styl, znaczniki)

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



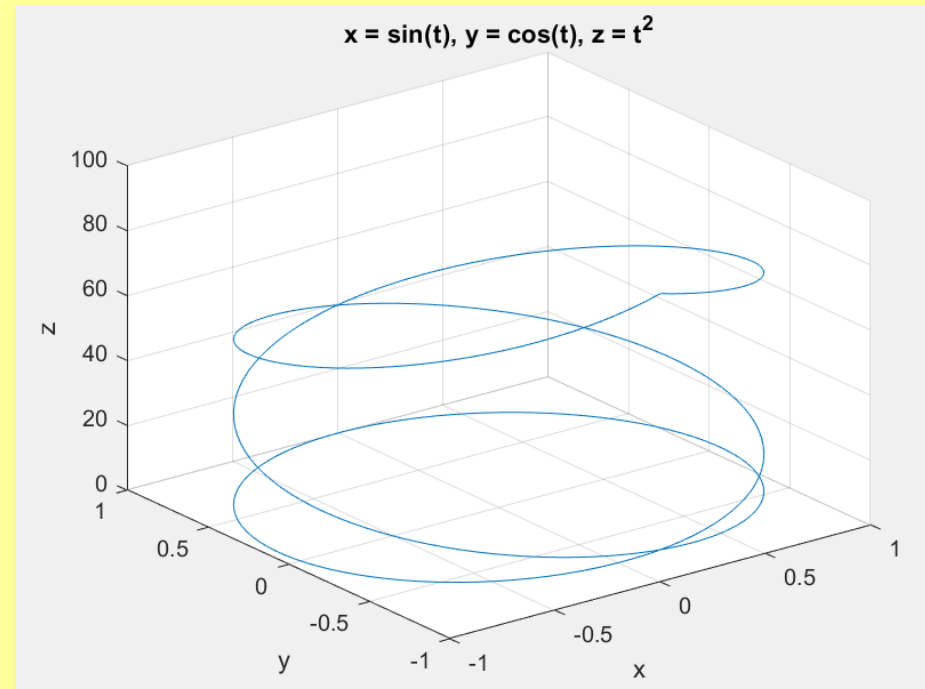
## Wykresy funkcji podanych wzorem

Matlab daje szerokie możliwości rysowania wykresów dwu- i trójwymiarowych funkcji definiowanych za pomocą wzorów, dzięki czemu użytkownik nie musi sam definiować niektórych parametrów (np. przedziału zmienności argumentu).


Oprócz poleceń **plot**, **plot3** oraz **surf** ( $x, y, z, c$ ), które służą do rysowania dwu- i trójwymiarowych wykresów wektorów  $x, y$  i  $z$ , Matlab umożliwia rysowanie wykresów funkcji podanych wzorem.

Instrukcja **ezplot** jest dostępna także w trójwymiarowej wersji **ezplot3**. Jest ona przeznaczona do rysowania wykresów krzywych wyrażonych parametrycznie w przestrzeni trójwymiarowej.

**ezplot3** (x,y, z, [tmin tmax]) dodaje tytuł wykresu i opis osi sporządza wykres parametrycznej krzywej przestrzennej o współrzędnych  $x(t)$ ,  $y(t)$  i  $z(t)$  w przedziale zmienności parametru określonym przez tmin i tmax; automatycznie dodaje tytuł wykresu i opis osi dodaje tytuł wykresu i opis osi; przedział zmienności parametru może być pominięty i wtedy przyjmowany jest przedział jest  $\langle 0; 2\pi \rangle$ ; funkcje  $x(t)$ ,  $y(t)$  i  $z(t)$  mogą być zdefiniowane wzorami za pomocą łańcuchów znakowych.



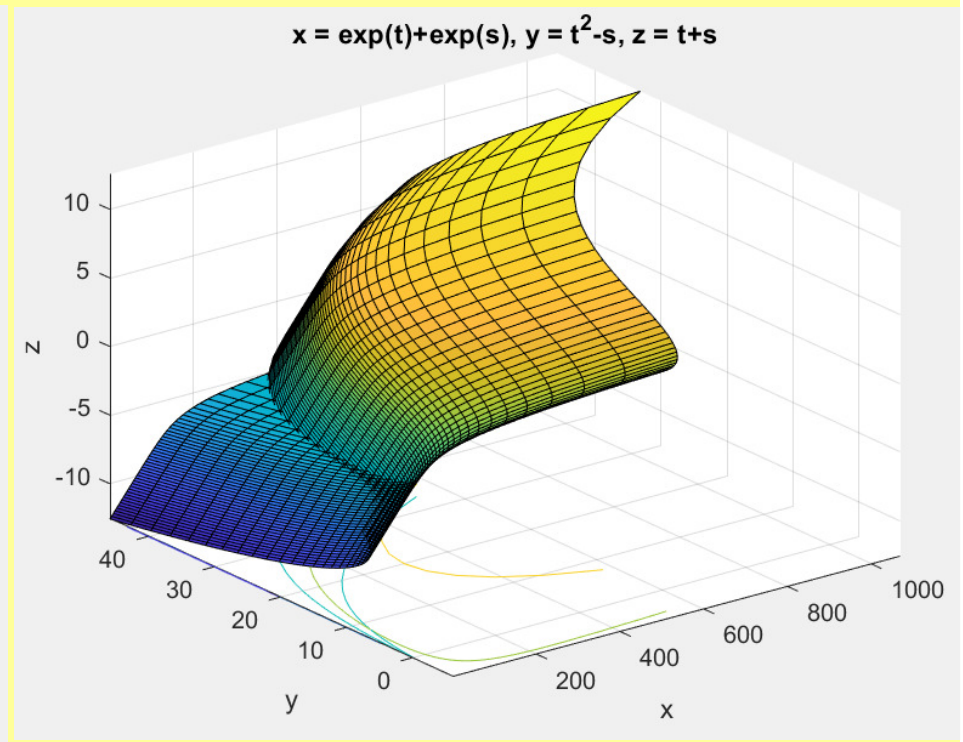
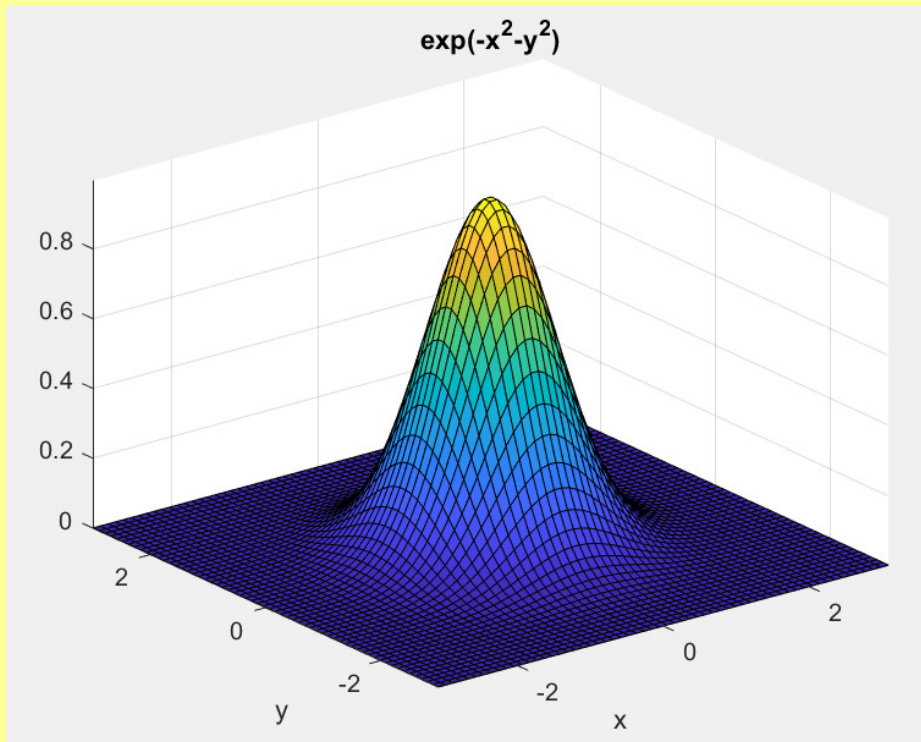
```
ezplot3('sin(t)', 'cos(t)', 't^2', [-3*pi 3*pi]);
```



**ezsurf ('funkcja', [xmin, xmax, ymin, ymax] lub ezsurf('funkcja') lub ezsurf (x, y, z, [smin, smax, tmin, tmax]) lub ezsurf (x,y,z)** sporządza trójwymiarowy wykres powierzchniowy funkcji dwóch zmiennych zdefiniowanej w łańcuchu znakowym 'funkcja' w podanym lub domyślnym przedziale zmienności zmiennych. Automatycznie dodaje tytuł wykresu i opis osi. Jeśli przedział zmienności zmiennych jest pominięty to przyjmowany jest przedział domyślny  $\langle -2\pi ; 2\pi \rangle$ . Powierzchnia może być również zdefiniowana poprzez współrzędne  $x(s, t)$ ,  $y(s, t)$ ,  $z(s, t)$  zależne od parametrów  $s$  i  $t$ . Funkcje  $x(t)$ ,  $y(t)$  i  $z(t)$  mogą być zdefiniowane wzorami za pomocą łańcuchów znakowych.


**ezsurfc ('funkcja', [xmin, xmax, ymin, ymax]) lub ezsurfc ('funkcja') lub ezsurfc ('funkcja') lub ezsurfc (x, y, z, [smin, smax, tmin, tmax]) lub ezsurf (x,y,z)** sporządza trójwymiarowy wykres powierzchniowy oraz dwuwymiarowy konturowy funkcji dwóch zmiennych zdefiniowanej w łańcuchu znakowym 'funkcja' lub określonej parametrycznie.

## ezsurf i ezsurf



```
ezsurf('exp(-x^2-y^2)');
```

```
ezsurf('exp(t)+exp(s)', 't^2-s', 't+s');
```

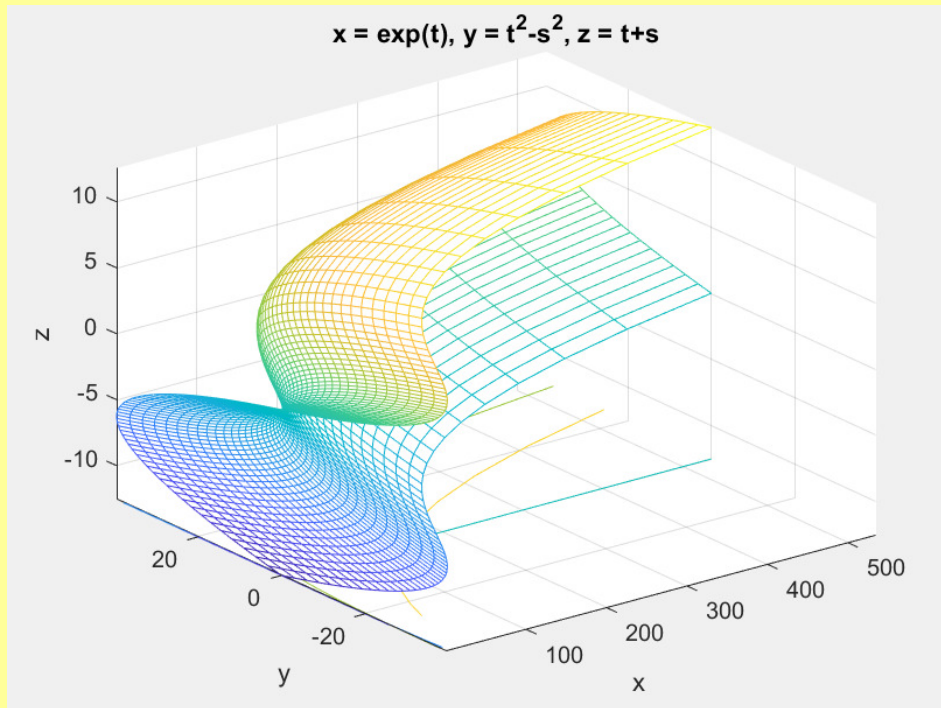
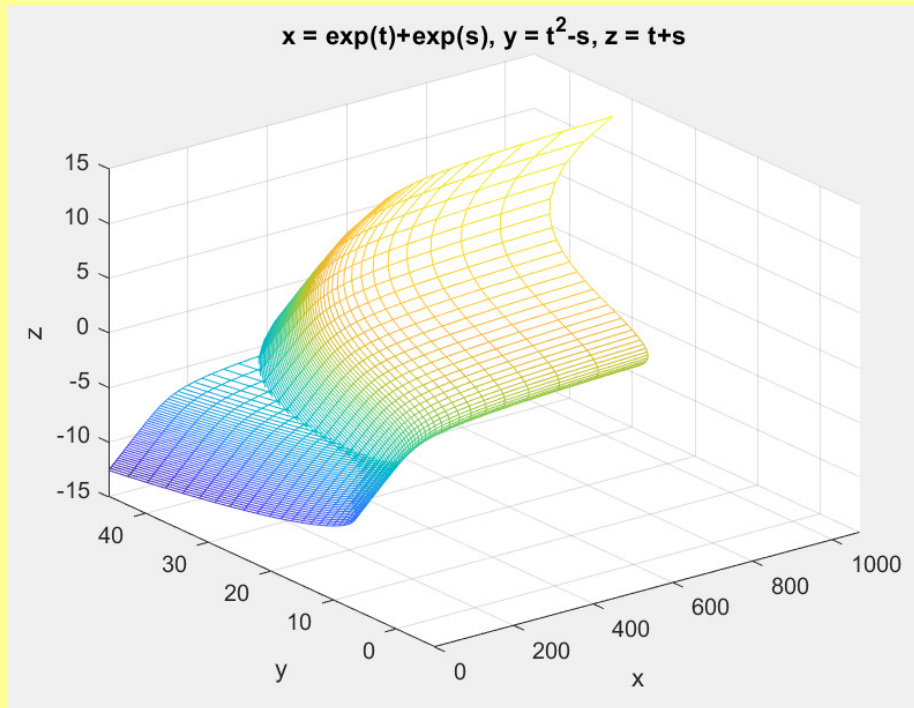


**ezmesh ('funkcja', [xmin, xmax, ymin, ymax]) lub ezmesh ('funkcja')** lub **ezmesh (x, y, z, [smin, smax, tmin, tmax])** lub **ezmesh (x,y,z)** sporządza trójwymiarowy wykres siatkowy funkcji dwóch zmiennych zdefiniowanej w łańcuchu znakowym 'funkcja' w podanym lub domyślnym przedziale zmienności zmiennych; automatycznie dodaje tytuł wykresu i opis osi; Jeśli przedział zmienności zmiennych jest pominięty to przyjmowany jest przedział domyślny  $\langle -2\pi ; 2\pi \rangle$ ; siatka może być również zdefiniowana poprzez współrzędne  $x(s, t)$ ,  $y(s, t)$ ,  $z(s, t)$  zależne od parametrów  $s$  i  $t$ ; funkcje  $x(t)$ ,  $y(t)$  i  $z(t)$  mogą być zdefiniowane wzorami za pomocą łańcuchów znakowych.

**ezmeshc ('funkcja', [xmin, xmax, ymin, ymax]) lub ezmeshc ('funkcja')** lub **ezmeshc (x, y, z, [smin, smax, tmin, tmax])** lub **ezmeshc (x,y,z)** sporządza trójwymiarowy wykres siatkowy oraz dwuwymiarowy konturowy funkcji dwóch zmiennych zdefiniowanej w łańcuchu znakowym 'funkcja' lub określonej parametrycznie.




## ezmesh i ezmeshc



```
ezmesh('exp(t)+exp(s)', 't^2-s', 't+s');
```

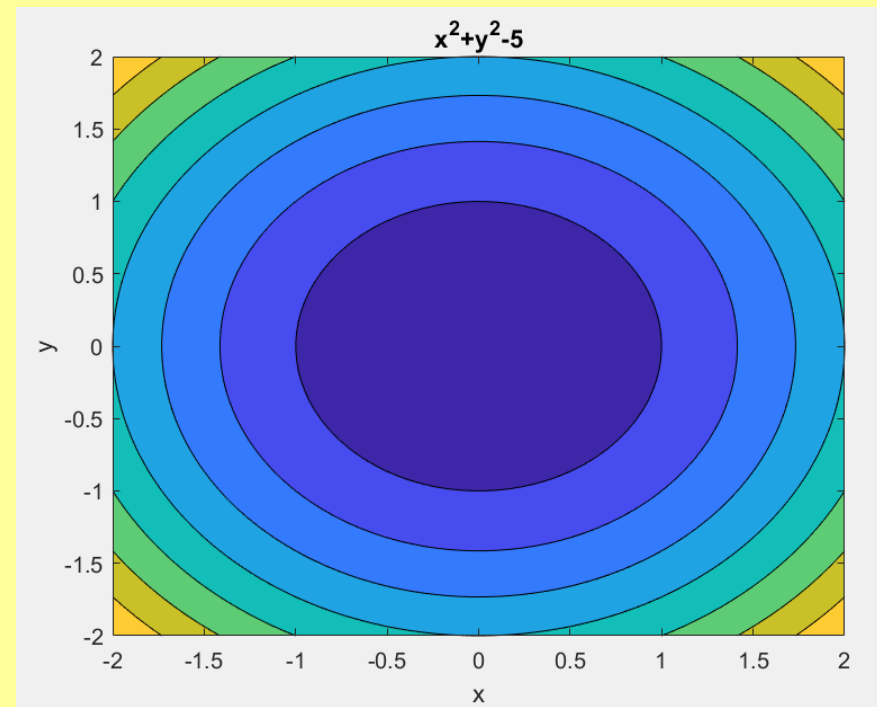
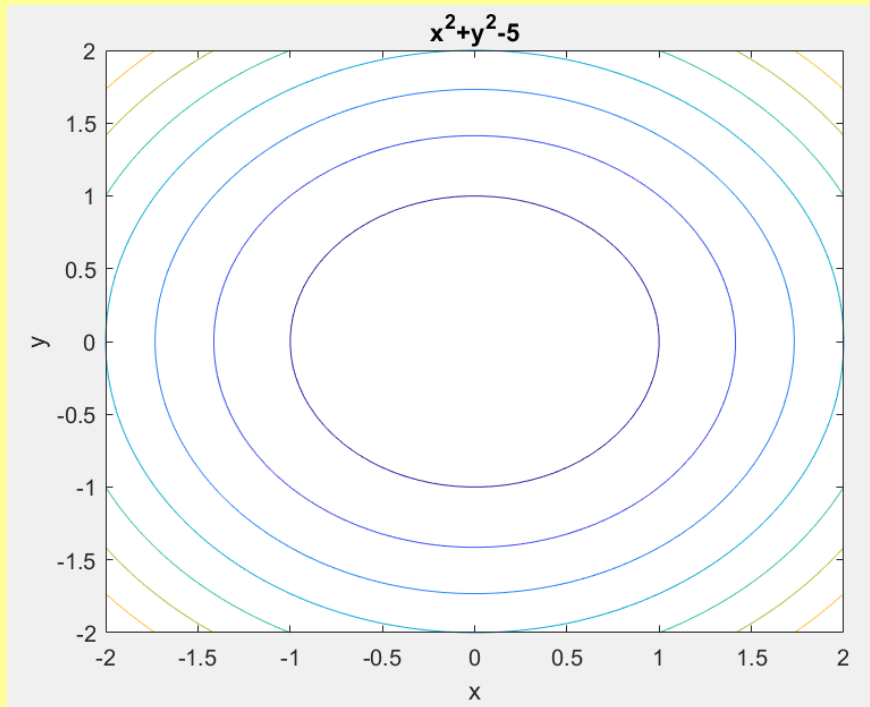
```
ezmeshc('exp(t)', 't^2-s^2', 't+s');
```



**ezcontour ('funkcja', [xmin, xmax, ymin, ymax])** lub **ezcontour ('funkcja')** sporządza wykres konturowy funkcji dwóch zmiennych zdefiniowanej w łańcuchu znakowym 'funkcja' w podanym przedziale zmienności zmiennych; automatycznie dodaje tytuł wykresu i opis osi; jeśli przedział zmienności zmiennych jest pominięty to jest przyjmowany przedział domyślny  $\langle -2\pi ; 2\pi \rangle$ .

**ezcontourf ('funkcja', [xmin, xmax, ymin, ymax])** lub **ezcontourf ('funkcja')** sporządza wypełniony wykres konturowy funkcji dwóch zmiennych zdefiniowanej w łańcuchu znakowym 'funkcja' w podanym lub domyślnym przedziale zmienności zmiennych; automatycznie dodaje tytuł wykresu i opis osi; jeśli przedział zmienności zmiennych jest pominięty to przyjmowany jest przedział domyślny  $\langle -2\pi ; 2\pi \rangle$ .

## ezcontour i ezcontourf



```
ezcontour('x^2+y^2-5', [-2, 2]);
```

```
ezcontourf('x^2+y^2-5', [-2, 2]);
```

## Polecenia mesh i surf

Rysowanie powierzchni jest realizowane za pomocą dwóch poleceń występujących w kilku odmianach: **mesh** oraz **surf**. Obydwa mają w ogólnym przypadku składnię:

**polecenie** (x, y, z, c) gdzie x i y to wektory zmiennych niezależnych, z wartość funkcji, c opcjonalnie macierz definiująca kolory rysowanej macierzy.

Instrukcja **mesh** ta jest przeznaczona do rysowania szkieletów (siatek) powierzchni. Podstawowa składnia ma postać: **mesh(x, y, z)**. Może też przyjąć prostszą formę **mesh(z)**, gdy wcześniej mamy określone macierze współrzędnych x i y utworzone za pomocą instrukcji **meshgrid(a:krok:b)** oraz funkcję **z = z(x,y)**.

Można także otrzymać połączenie wykresu szkieletowego i poziomicowego stosując instrukcję **meshc(x, y, z)** lub plasterkowego i szkieletowego gdy użyjemy polecenia **meshz(...)**. Dodatkowo do każdej instrukcji można wprowadzać pewne własności dotyczące powierzchni. Instrukcja przyjmuje na przykład postać: **mesh(...,c)**, gdzie c oznacza kolor rysowanej powierzchni lub też **mesh(...,'PropertyName', 'PropertyValue' )**.

Property Name - nazwa ogólna ustawianej własności, np.: **MarkerType**, **MarkerFaceColor** itp.

Property Value - wartość liczbowa lub oznaczenie symboliczne konkretnej własności

## Polecenia mesh i surf

**mesh (z), mesh(z, y, z), mesh(z, y, z, c)** rysuje siatkę wartości macierzy z, przyjmując za zmienne niezależne wektor  $x_{1 \times m}$  oraz  $y_{1 \times n}$  oraz kolorując powierzchnię wartościami macierzy c; jeżeli pominię się parametr x oraz y, przyjmowane są odpowiednio wektory 1:m oraz 1 :n

**meshc(x, y, z, c)** tak jak **mesh**, ale rysowane są również poziomice pod wykresem

**meshz (x, y, z, c)** tak jak **mesh**, ale rysowane są również linie określające boczne płaszczyzny wykresu

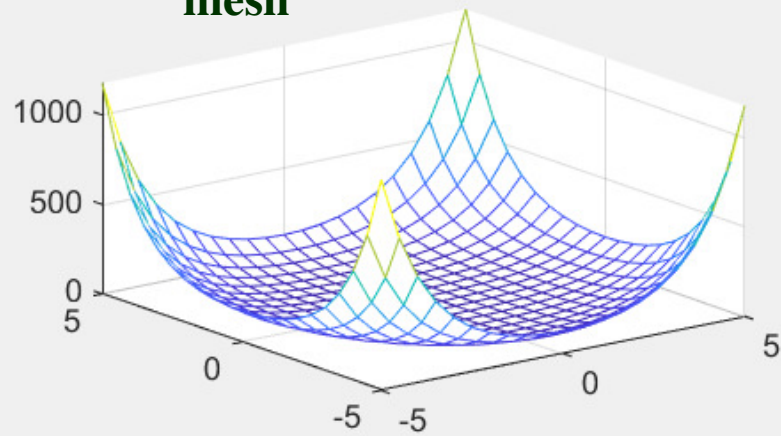
**surf (z), surf(x, y, z), surf(x, y, z, c)** analogicznie do mesh, z tym że tym razem rysowana jest kolorowa powierzchnia „dachowa”

**surfc(x,y, z, c)** tak jak meshc

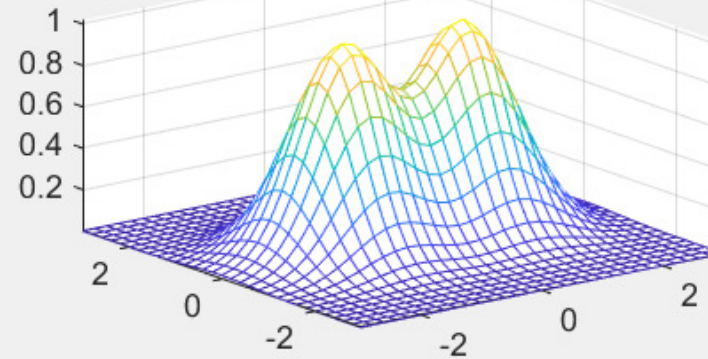
**waterfall (x, y, z, c)** tak jak meshz

## Polecenia mesh i surf

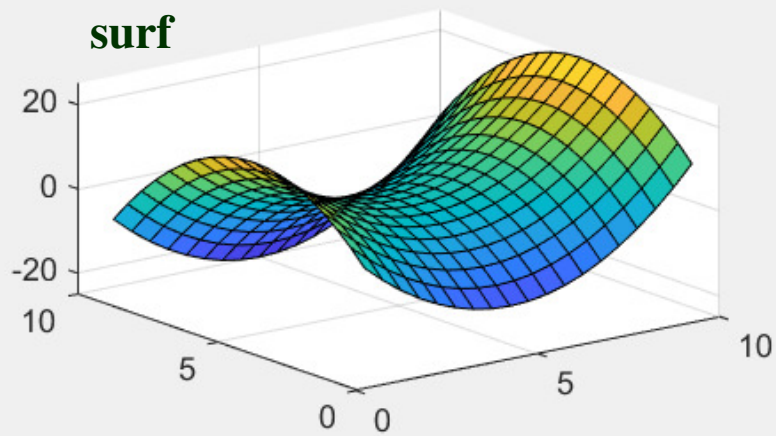
**mesh**



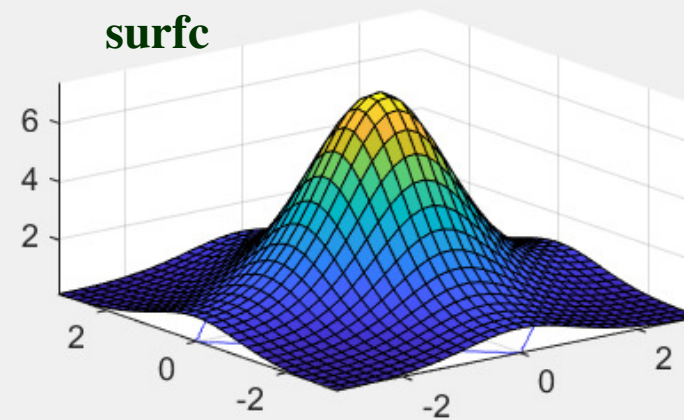
**meshc**



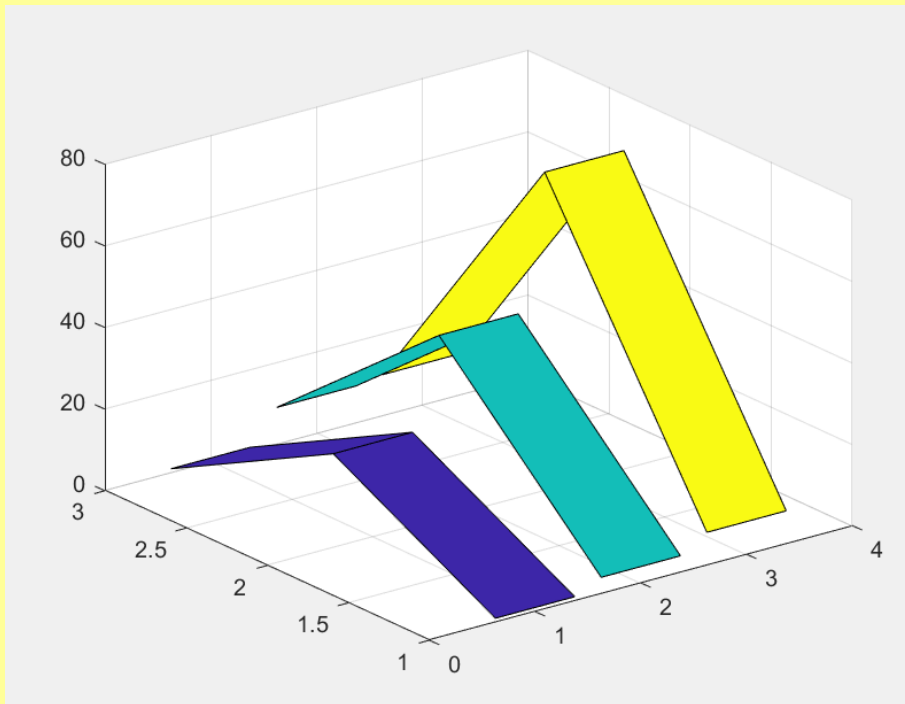
**surf**



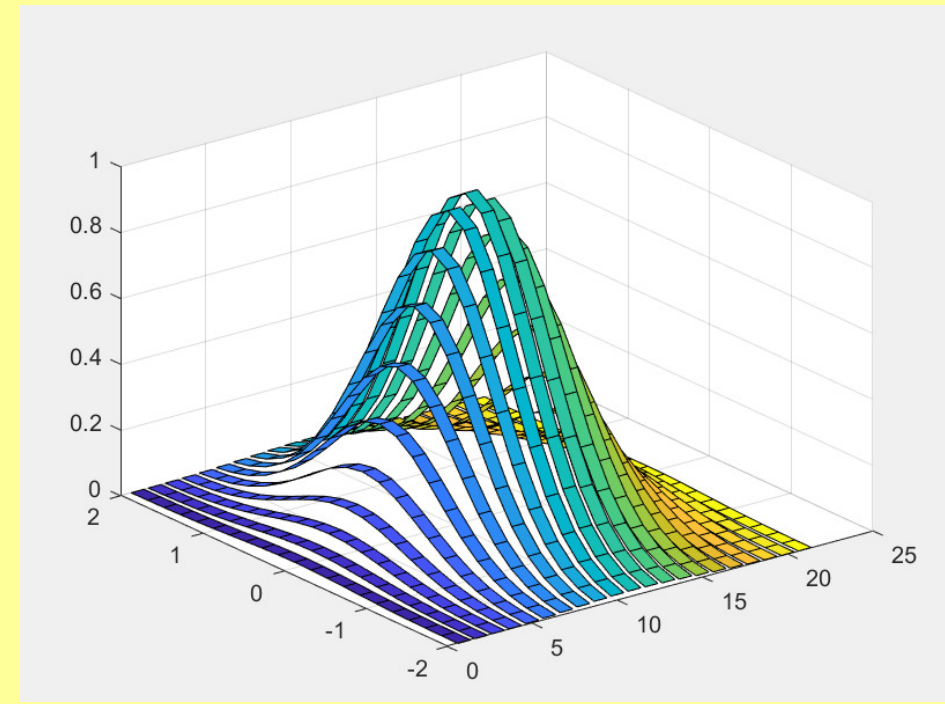
**surfc**



## Wykresy wstęgowe (ribbon)



`ribbon(y);`



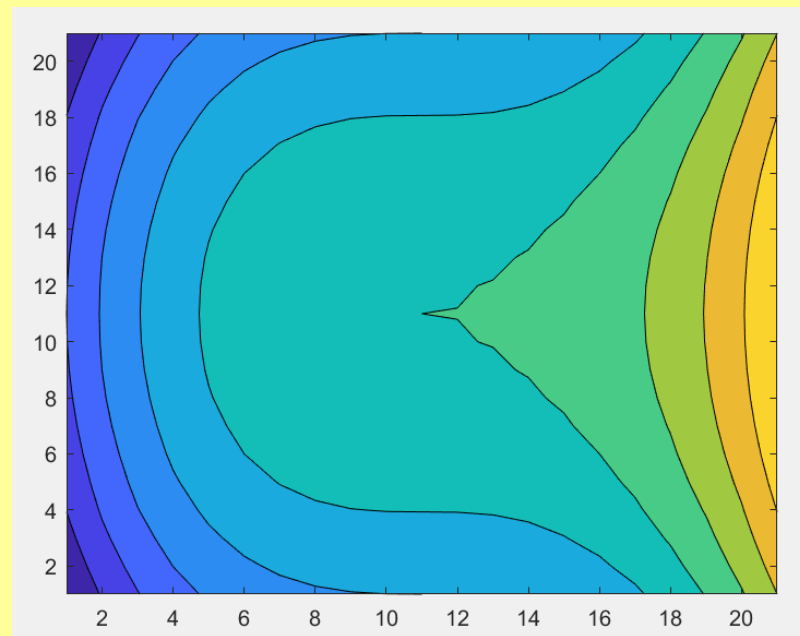
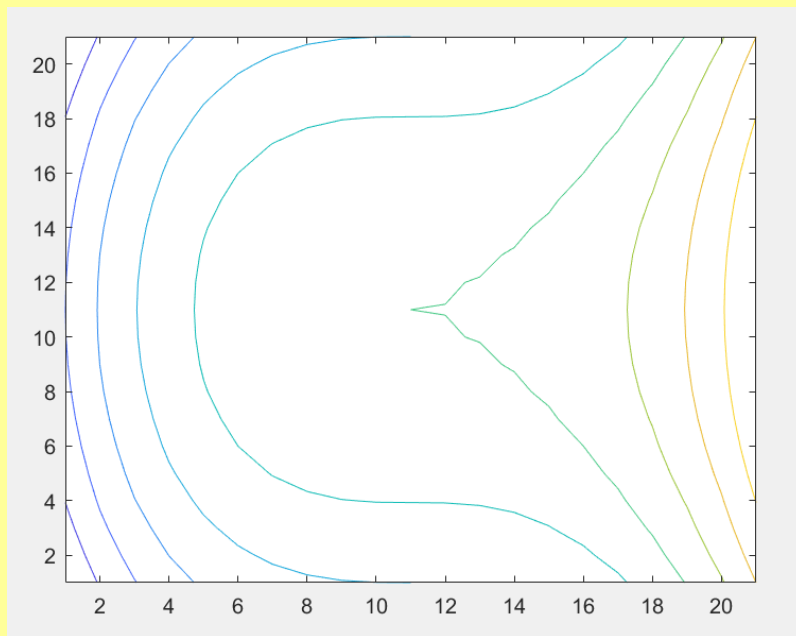
`ribbon(y,z);`

## Wykresy typu „contour”

Wykresy typu „contour” stanowią rodzinę **wykresów warstwicznych (poziomicowych)**. Mogą być rysowane wspólnie z innymi wykresami powierzchniowymi. Aby narysować wykres typu contour należy:

- zdefiniować za pomocą polecenia **meshgrid** siatkę współrzędnych (x, y),
- zdefiniować powierzchnię **z** jako funkcję zmiennych x i y,
- użyć polecenia **contour(z)** - składnia podstawowa.

```
[x,y]=meshgrid(-2:.2:2); z=x.^3-y.^2;  
contour(z);
```





## Wykresy typu „contour”

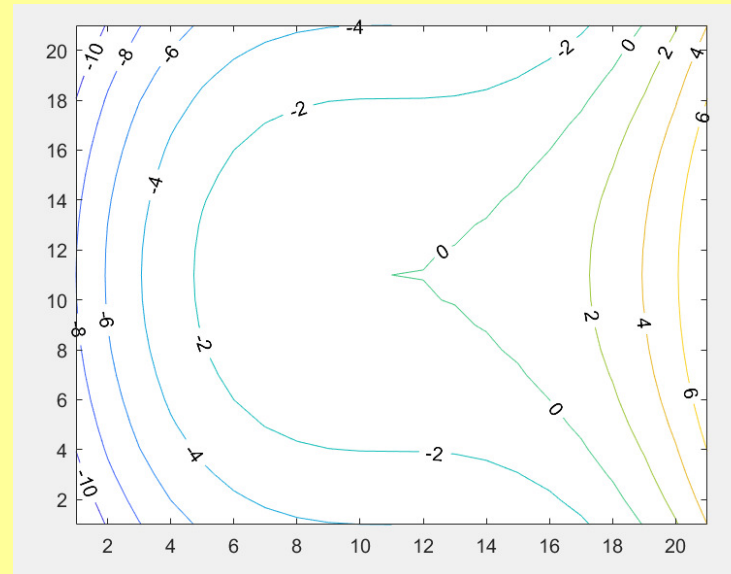
Zmienna **z** jest interpretowana jako wysokość związana ze zmiennymi **x** i **y**. Liczba poziomów jest wybierana automatycznie i jest związana z minimum i maximum wartości **z**. Aby samodzielnie określać liczbę poziomów należy użyć polecenia **contour(z, n)**.

Chcąc samodzielnie zdecydować o rodzaju linii, jaką ma być rysowany wykres, należy użyć polecenia **contour(...,LineStyle)**, gdzie zamiast parametru **LineStyle** wstawiamy symbol odpowiedniej linii. Aby otrzymać wykres typu **contour** z wyspecyfikowaną w każdym kolorze liczbą oznaczającą wysokość należy:

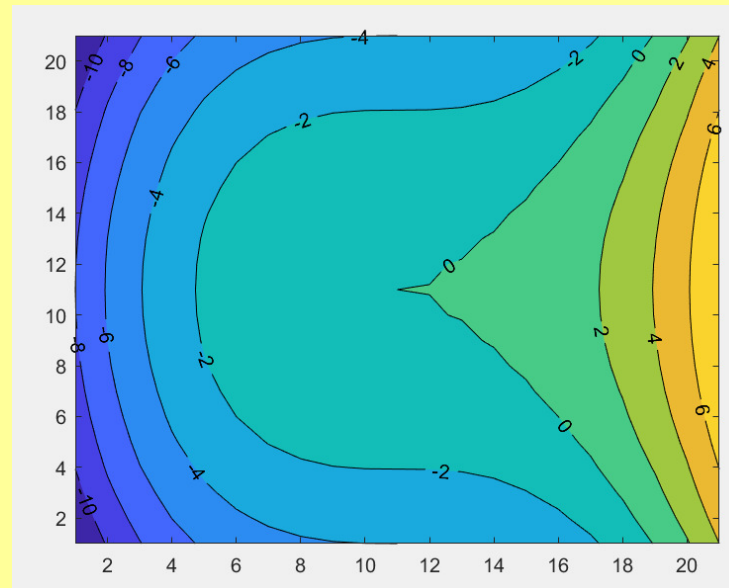
- utworzyć obiekt z **uchwytem** wykorzystując polecenie `[c, h]=contour(...)`,
- wywołać wykres konturowy poleceniem `clabel(c,h)`

**Uchwyt (*handle*)**, to - najprościej ujmując - etykieta lub numer przydzielony obiektowi, który został zdefiniowany zgodnie z obowiązującymi go zasadami. Każdy obiekt graficzny można bowiem od razu wizualizować lub tylko wygenerować jego własności, np. współrzędne, i zapamiętać pod pewną nazwą. Dopóki obiekt funkcjonuje w postaci uchwytu w przestrzeni roboczej, można go użyć przy okazji wywołania innej funkcji. Można utworzyć także uchwyt do mapy kolorów, czyli do elementu określającego cechy obiektu.

```
[c,h]=contour(z);  
clabel(c,h);
```



```
[c,h]=contourf(z);  
clabel(c,h);
```



## Wykres Coneplot

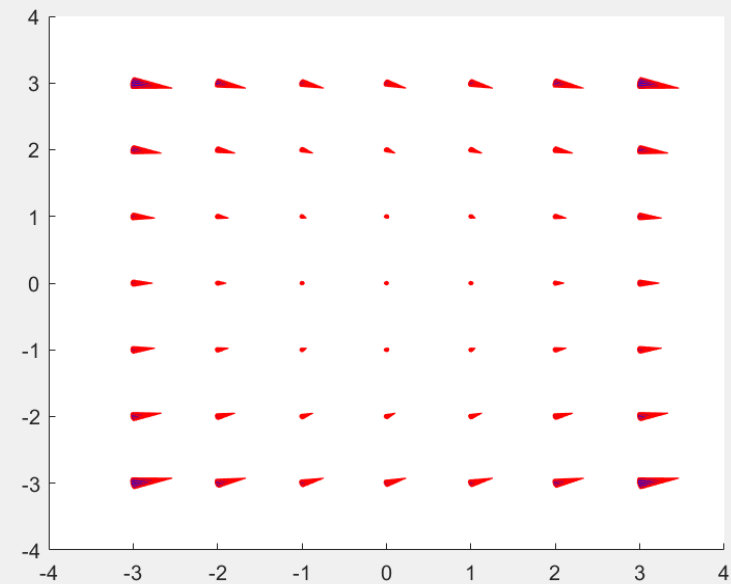
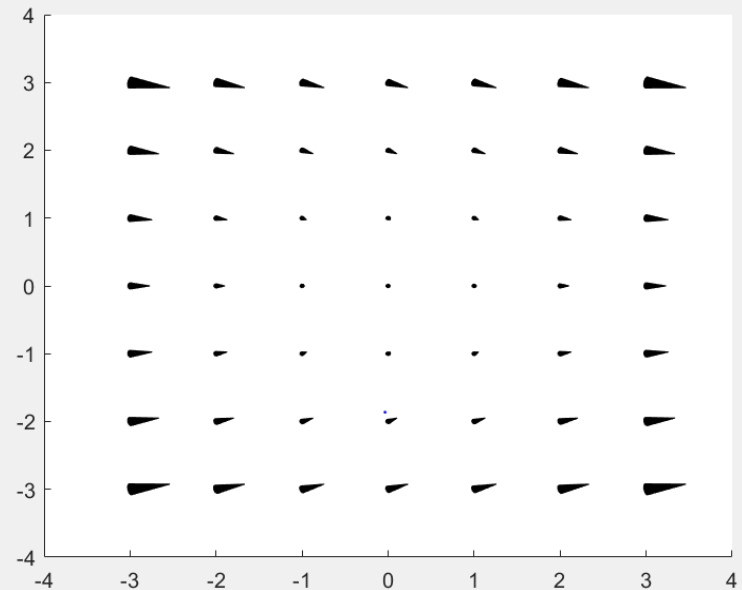
Wykres stożkowy służy do prezentowania i obrazowania pól wektorowych. Stożki są skierowane w kierunku zgodnym z kierunkiem wektorów prędkości i mają długość proporcjonalną do wartości wektorów prędkości. Aby zobrazować pole wektorowe za pomocą instrukcji o składni **coneplot(x,y,z,u,v,w,cx,cy,cz)**, należy:


- zdefiniować współrzędne x, y, z, które określają zakresy wyświetlania pola wektorowego, co można zrobić za pomocą polecenia **[x,y,z]=meshgrid(-2:2:2)**, jeżeli wszystkie współrzędne mają taki sam zakres lub **[x,y,z]=meshgrid(-3:0.2:2;-5:5:5; -3:2:3)** w przypadku różnych zakresów dla każdej ze współrzędnych
- zdefiniować składowe pola wektorowego np.:  $u=x.^2$   $v=-y$ ,  $w=2*z$
- określić punkty zaczepienia stożków w określonej przestrzeni **[cx,cy,cz] = meshgrid(-3:5:2,-5:1:5,-3:1:3)**.

Aby dokonać zmiany parametrów wykresu należy utworzyć uchwyt i za jego pośrednictwem odwołać się do wykresu:

```
hcones=coneplot(x,y,z,u,v,w,cx,cy,cz);
```

```
set(hcones,'Facecolor','blue','EdgeColor','red'); %wprowadzenie  
czerwonego dla stożków i  
niebieskiego dla punktów  
zaczepień
```





Do podstawowego polecenia można dodać kilka parametrów. Jeżeli do polecenia podstawowego dodamy parametr **s**: **coneplot(x,y,z,u,y,w,cx,cy,cz,s)** - wówczas mamy możliwość nieautomatycznego skalowania stożków. Automatyczne skalowanie następuje wtedy, gdy wartość parametru **s** jest ustawiona na 1 lub wcale nie jest zapisana.

Aby zmienić kolor stożków należy zdefiniować tablicę kolorów **c**, która musi mieć takie same rozmiary jak tablice **u**, **y**, **w**. **coneplot(x,y,z,u,y,w,cx,cy,cz,c)**.

Aby dodatkowo zaznaczyć zwrot strzałkami należy dodać parametr **quiver**.  
**coneplot(x,y,z,u,y,w,cx,cy,cz, 'quiver')**

Sposób wyświetlania jest określony parametrem **method**, który przyjmuje jedną z wartości: **linear**, **cubic**, **nearest**. **coneplot(x,y,z,u,y,w,cx,cy,cz, 'method')**.

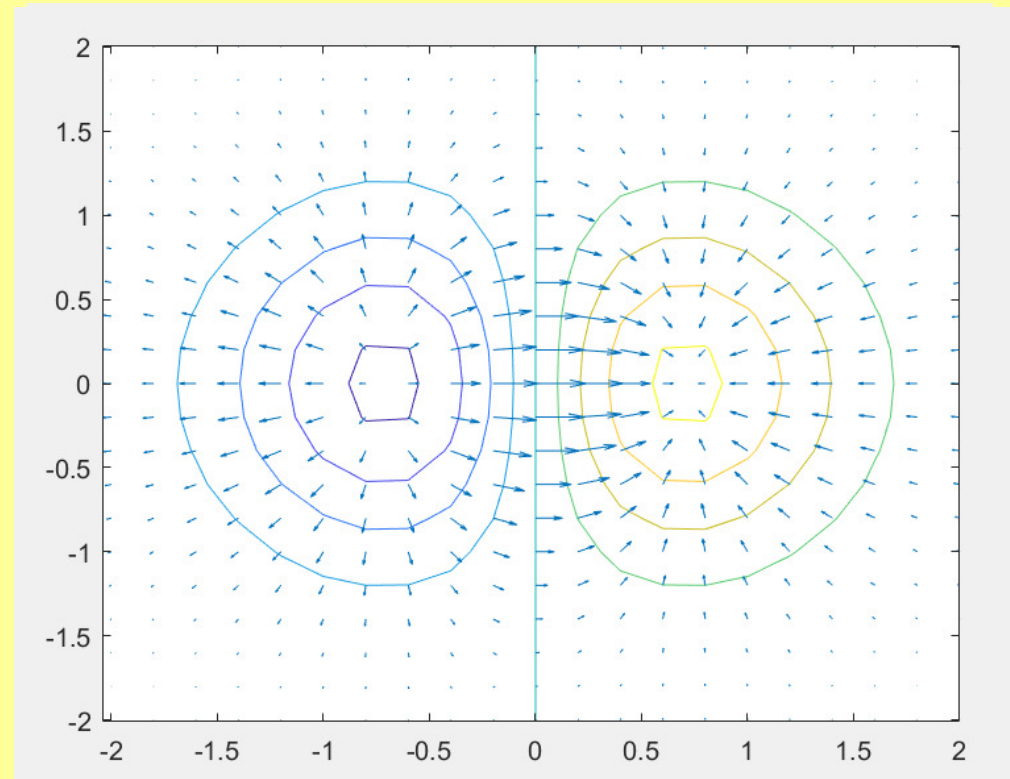
## Pole gradientu

```
[x,y]=meshgrid(-2:0.2:2);  
z=x.*exp(-x.^2-y.^2);  
contour(x,y,z);
```

## Pojawiają się poziomicice

```
[dx,dy]=gradient(z,0.2,0.2);  
hold on;  
quiver(x,y,dx,dy);  
%dx, dy - składowe wektorów
```

## Pojawiają się wektory

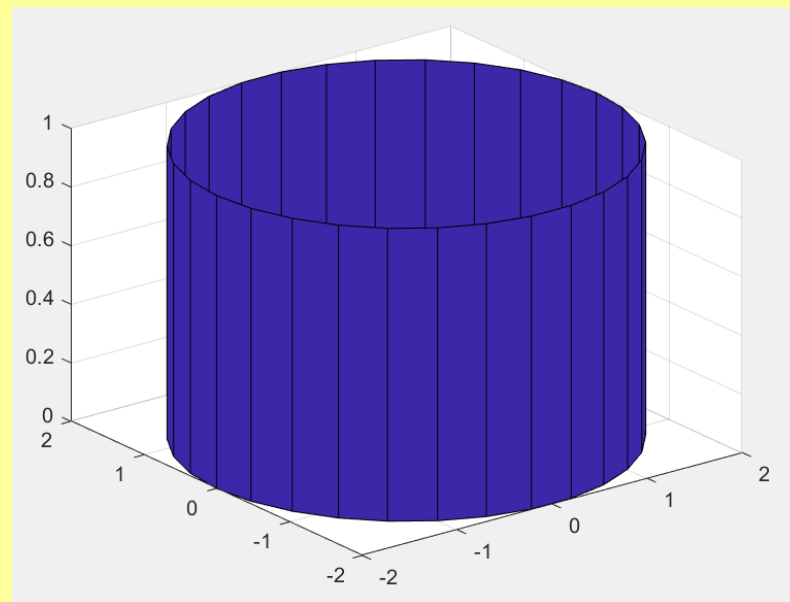


## Rysowanie brył obrotowych

### Walec

**Cylinder.** Instrukcja ta generuje walec lub powierzchnię walcową. Podstawowa składnia tej instrukcji to po prostu **cylinder** (walec o promieniu podstawy 1 i wysokości 1) lub **[x,y,z]=cylinder**. Walec należy narysować za pomocą instrukcji **surf(x,y,z)** lub **mesh(x,y,z)**.

Walec o promieniu **r** i **n** częściach, z których ta powierzchnia będzie się składać, można wygenerować przy pomocy instrukcji **[x, y, z]=cylinder(r,n)**.

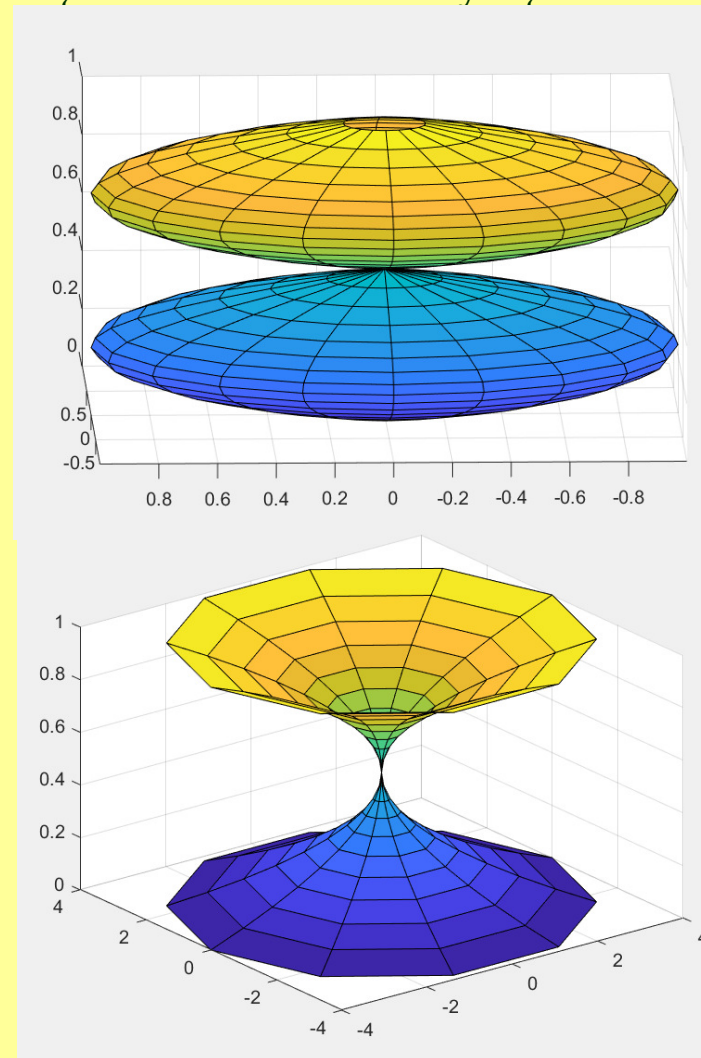


## Rysowanie brył obrotowych

Aby wygenerować powierzchnię cylindryczną o innym kształcie, można zamiast wartości  $r$  wpisać dowolną funkcję, która będzie stanowić krzywą bazową dla generowanej powierzchni.

```
x=[-3:0.2:3];  
[x,y,z]=cylinder(sin(x),20);  
surf(x,y,z)  
axis equal;
```

```
input('powierzchnia cylindryczna  
generowana przez parabolę');  
t=[-2:0.2:2];  
[x,y,z]=cylinder(t.^2,10);  
surf(x,y,z)
```



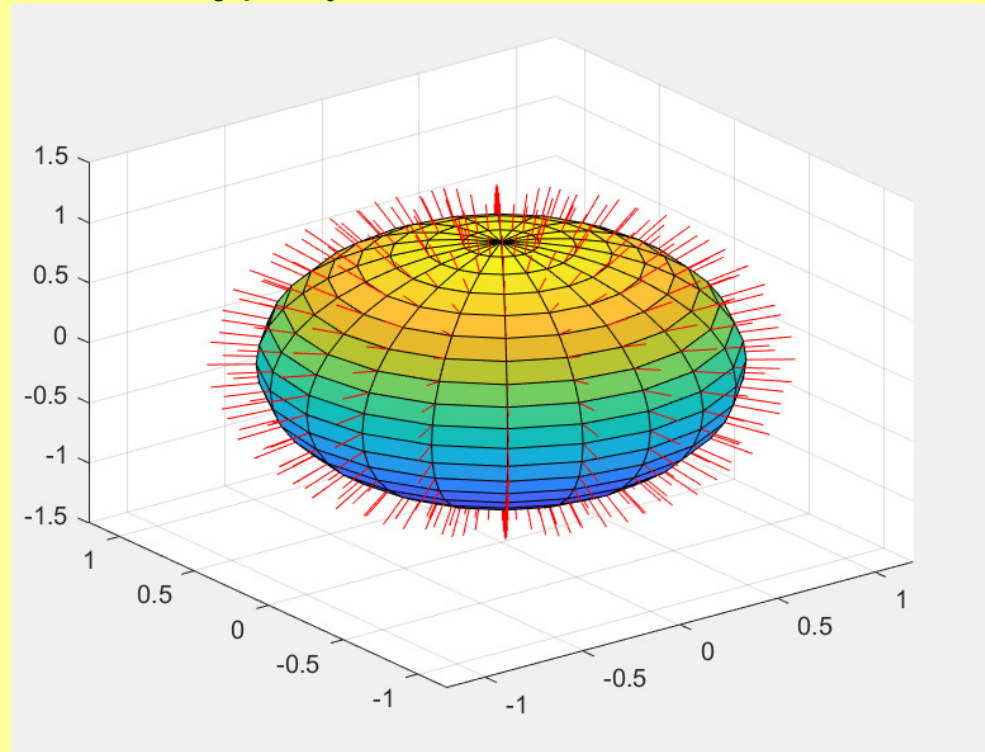


## Rysowanie brył obrotowych

### Powierzchnia sferyczna

Polecenie **sphere** generuje siatkę sfery. Potem trzeba ją narysować (mesh lub surf).

```
[x, y, z]=sphere;  
colormap bone;  
surfc(x, y, z);
```



Polecenie **surfnorm** rysuje wektory normalne

```
surfnorm(x, y, z)
```



**DZIĘKUJĘ ZA UWAGĘ**