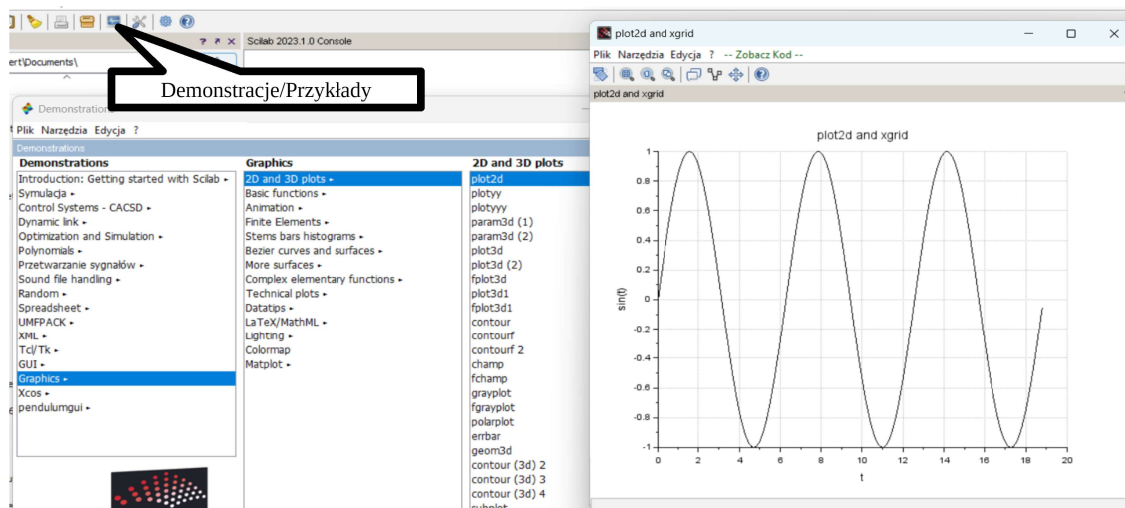


ĆWICZENIE 4: Elementy graficzne – wykresy.

Cel: Celem tego rozdziału jest zapoznanie studenta ze możliwościami jakie oferuje program SCILAB w zakresie tworzenia elementów graficznych, w szczególności różnego rodzaju wykresów. Omówione zostaną podstawowe polecenia służące tworzeniu wykresów, oraz metody modyfikacji i dostosowywania ich wyglądu.

SCILAB umożliwia tworzenie wykresów 2-wymiarowych, trójwymiarowych, w trybie subplot (wiele wykresów w jednym oknie graficznym) z zastosowaniem różnych specyficznych typów wykresów. Utworzone wykresy mogą być edytowane w zakresie opcji dotyczących wyglądu wykresu (kolory, typy linii, tytuły, znaczniki osi itp.) a także powiększane i obracane (wykresy 3D). Możliwości w zakresie tworzenia wykresów są obszernie prezentowane w *Demonstracje/Przykłady*.



Podczas tworzenia wykresów, przydatne może być korzystanie z kolorów. Dostępne standardowe kolory możemy sprawdzić poleceniem `getcolor()`. W wywołanym oknie zobaczymy paletę dostępnych kolorów standardowych. Klikając „kafelki” z kolorem, w dolnej części okna zobaczymy numer koloru, nazwę, oraz kompozycję RGB.

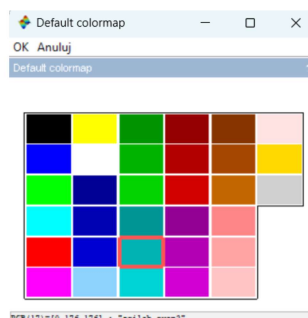
Wykresy dwuwymiarowe – podstawy:

Wykresy są tworzone w dodatkowym oknie nazywanym oknem graficznym. W SCILAB okno graficzne i jego zawartość (np. wykres) mają strukturę hierarchiczną. W oknie graficznym jest umieszczany wykres zawierający co najmniej jedną oś współrzędnych, z kolei oś/osie zawiera elementy podrzędne takie jak linie wykresu czy zestaw parametrów sterujących wyglądem obszaru wykresu (np. kolory tła oraz osi, położenie osi, rozmieszczenie. Linia wykresu zawiera z kolei zestaw parametrów sterujących jej wyglądem (styl linii lub znacznika, kolor).

Podstawowym poleceniem jest `plot2d`.

`plot2d(argument)`

Można je wywoływać z argumentami różnych typów.



A) Jeżeli argumentem jest wektor to na osi Ox umieszczane zostaną numery współrzędnych wektora (pozycje kolejnych elementów wektora), na osi Oy natomiast współrzędne tego wektora.

Przykład:

Chcemy zbadać zachowanie ciągu $a_n = \sin(n^2)$. dla $1 \leq n \leq 10000$

Tworzymy wektor z kolejnymi elementami ciągu (należy pamiętać o zakończeniu polecenia znakiem średnika ';' co zapobiegnie wyświetleniu 10000 cyfr w oknie konsoli):

```
--> A=sin([1:10000].^2);
```

W wektorze wierszowym uzyskujemy 10000 początkowych wyrazów badanego ciągu.

Chcąc narysować wykres fragmentu tego ciągu (np. wyrazy od 801-go do 1000-go) stosujemy polecenie:

--> `plot2d(A(801:1000))`

Wyrażenie $A(801:1000)$ indeksuje elementy wektora A , od elementu z pozycji 801 do elementu z pozycji 1000 co daje łącznie 200 elementów. Z tego powodu na osi Ox są liczby w zakresie do 200 co może być mylące. Aby uzyskać prawidłowe skalowanie osi Ox , odpowiadające pozycjom na jakich w wektorze A znajdują się liczby wzięte do utworzenia wykresu należy dodatkowo podać wektor zawierający liczby do oznaczenia osi Ox :

--> `plot2d(801:1000,A(801:1000))` – zastosowana tu składnia zostanie opisana w punkcie C)

UWAGA:

Jeżeli powyższe polecenie rysowania wykresu zostanie wywołane przy otwartym oknie graficznym (z wykresem utworzonym wcześniej) to nowy wykres zostanie dorysowany do już istniejącego. Jest to standardowe ustawienie programu SCILAB przydatne gdy chcemy rysować wiele wykresów w jednym układzie współrzędnych. Jeśli chcemy uzyskać tylko wykres z prawidłowo wyskalowaną osią Ox należy wykonać jedno z działań:

- 1- zamknąć okno graficzne z poprzednim wykresem (klikając w x w górnym prawym rogu okna graficznego),
- 2- zamknąć okno graficzne z poprzednim wykresem wpisując w konsoli polecenie `close`,
- 3- ‘wyczyścić’ okno graficzne z wcześniej narysowanego wykresu poleceniem `clf`.

Można też zmienić standardowe ustawienie powodując, że nowy wykres nie będzie ‘dorysowywany’ tylko zastąpi poprzedni: `gda().auto_clear = "on"`. Jeżeli chcemy aby kolejny wykres był rysowany w nowym oknie graficznym to należy najpierw aktywować/otworzyć nowe-puste okno graficzne nadając mu numer poleceniem `scf(numer)`. Nowy wykres będzie rysowany ‘nowym’ oknie graficznym.

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem `clf`.

B) Jeżeli argumentem polecenia `plot2d` jest macierz to na osi Ox umieszczane zostaną numery wierszy macierzy, na osi Oy natomiast współrzędne będące wartościami w poszczególnych wierszach macierzy.

Przykład:

Chcemy narysować wykresy funkcji trygonometrycznych $\sin$, $\cos$, w zakresie $0-2\pi$.

--> `X=linspace(0,2*%pi,201)` //– wektor 201-liczb w zakresie $0-2\pi$ rozłożonych liniowo.

--> `Y=[sin(X); cos(X)]'` //– dwukolumnowa macierz z wartościami funkcji $\sin$, $\cos$.

--> `plot2d(Y)` //– rysownie wykresów.

Jak można zauważyć oś Ox jest skalowana wg liczby wierszy w macierzy Y (0-201).

C) Jeżeli argumentem są dwa wektory `{plot2d(wektorX,wektorY)}` - (*wektorX* oraz *wektorY* muszą mieć taką samą długość) to na osiach Ox oraz Oy umieszczane są współrzędne (wartości) z odpowiednich pozycji tych wektorów. Polecenie to nadaje się do rysowania tradycyjnych wykresów ze skalowaniem osi Ox .

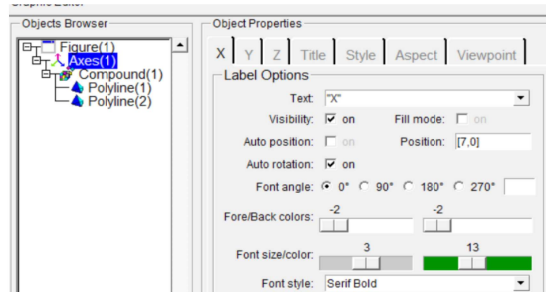
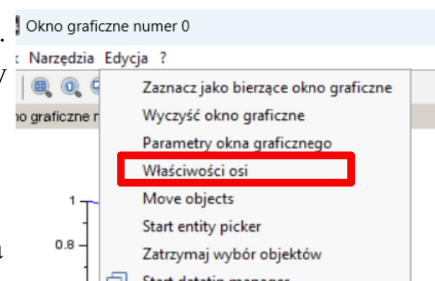
Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem `clf`.

--> `plot2d(X,Y)` – odpowiednie skalowanie na osi Ox . Wartości z wektora X skalują oś Ox .

Uzyskany wykres jest dość ‘surowy’ i jeśli miałby być wykorzystany np. w prezentacji wymaga dopracowania. Poddajmy go edycji aby poprawić wygląd. Jeśli chcemy edytować utworzony wykres wybieramy w menu okna wykresu polecenie *Edycja-Właściwości osi*.

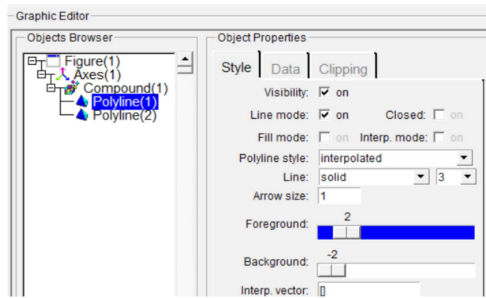
W nowo otwartym oknie możemy zmieniać wszystkie parametry odpowiedzialne za wygląd wykresu.

- Aby zmienić kolor linii wykresów rozwijamy *Axes-Compound* i zaznaczamy *Polyline (1)* lub *Polyline (2)*.
- Aby zmienić kolor linii ustawiamy suwak *Foreground* w pozycję 2 dla koloru niebieskiego lub 5 dla koloru czerwonego.
- Typ linii pozostawiamy *solid*, grubość linii zmieniamy na 3 (rysunek po lewej).
- Zaznaczając *Axes(1)* możemy wprowadzić podpisy poszczególnych osi X , Y , Z oraz w zakładce *Title* podpis całego

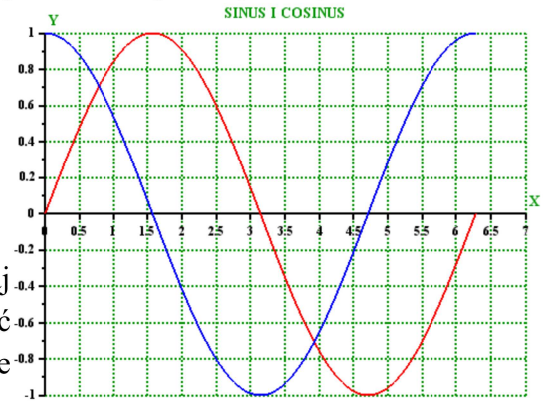


wykresu (rysunek po prawej). Dla osi X, Y wybieramy *Font size 3*, *Font color 13*, *Grid color 13*. W polach *Text* wstawić podpisy wpisać pozycję tytułów X wybrać *Location*

- W zakładce *Title* *COSINUS*', wybrać *Font size 3*, *Font color 13*. Zapisać wykres do pliku poleceniem *Plik-Zapisz*, nazwa pliku *W1.scg*.



w polu *Text* wpisać tytuł wykresu '*SINUS I*



Możliwości edycji wykresu jest znacznie więcej.

Np w zakładce *Style* można zmieniać rozmiar, kolor oraz rodzaj czcionki dla znaczników osi, kolor osi, kolor tła wykresu i grubość linii siatki. Należy samodzielnie przetestować pozostałe dostępne opcje edycji. Uzyskany rysunek powinien wyglądać jak obok.

Wszystkie opcje edycyjne wykresu mogą być również podane w formie poleceń tekstowych. Ma to znaczenie szczególnie jeśli wykres w końcowej postaci ma być efektem działania skryptu i nie będzie podlegał dodatkowej 'ręcznej' edycji. Więcej informacji o opcjach polecenia można znaleźć w pomocy (-->help plot2d)

W tym miejscu podamy sposób wywoływania opcji dotyczących koloru linii wykresu oraz zastosowania znaczników punktów wykresu. Jeśli zastosujemy składnię : *plot2d(x,y,style)* to parametr *style* określa odpowiednie kolory linii lub typy znaczników.

Jeśli parametr *style* jest dodatnią liczbą całkowitą, wykres jest rysowany linią ciągłą w kolorze zależnym od wartości tej liczby. Liczby odpowiadające kolorom znajdziemy używając polecenia *getcolor()*.

Jeśli parametr *style* jest ujemną liczbą całkowitą lub zerem, podane punkty wykresu są rysowane za pomocą znaczników. Typ znacznika zależy od wartości bezwzględnej parametru *style*. Poniżej podano znaczniki i przypisane im liczby (należy zastosować ujemne wartości).

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.



```
-->plot2d(X,Y(:,1),2)
```

```
-->clf
```

```
-->plot2d(X,Y(:,1),5)
```

```
-->clf
```

```
-->plot2d(X,Y(:,1),-3)
```

```
-->clf
```

```
-->plot2d(X,Y(:,1),-5)
```

Aby utworzyć nowe (kolejne) okno graficzne do rysowania wykresu, należy użyć polecenia *scf()*. Jeżeli w nawiasie polecenia *scf()* podamy liczbę to będzie ona numerem utworzonego okna, jeśli nawias pozostanie pusty to program przydzieli dla nowego okna graficznego kolejny numer.

W SCILAB dostępne jest również polecenie *plot()* zapożyczony z Matlab. Opcje polecenia *plot()* są również takie same jak w Matlab. Dla wielu, składnia polecenia *plot* może być bardziej przyjazna ponieważ używa oznaczeń literowych i symboli zamiast cyfr. Aktualnie polecenie *plot()* staje się również standardowym poleceniem w SCILAB.

Podstawowa składnia polecenia: *plot(x,y,LineSpec)*

Parametr *LineStyle* określa styl linii, styl znacznika i kolor wykreślanych linii. Może być w postaci jednego lub dwóch znaków. Należy pamiętać, że parametr *LineStyle* powinien być zapisany w apostrofach.

Oznaczenia typu linii:

- linia ciągła, -- linia kreskowa, : linia kropkowa, -. linia kropka-kreska.

Oznaczenia typu znacznika:

+ o * . x s(kwadrat) d(diament) ^ v > < p(pentagram).

Oznaczenia koloru: k-czarny, w-biały, r-czerwony, g-zielony, c-niebieski, c-cyjan, m-magneta, y-żółty.

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

```
-->plot(X,Y,'r—')      -->plot(X,Y,'bs')
```

Dodatkową funkcjonalnością jaką posiada polecenie *plot* (a nie posiada polecenie *Plot2d*) jest możliwość wykreślenia wykresu funkcji wbudowanej (np. *sin*, *cos*) oraz funkcji utworzonych przez użytkownika poprzez ich zaprogramowanie.

Podstawowa składnia takiego polecenia: *plot(x, func)*, gdzie *x* jest wektorem lub macierzą argumentów wejściowych funkcji, *func* jest standardową nazwą funkcji wbudowanej lub nazwą nadaną przez użytkownika podczas programowania funkcji użytkownika.

Przykład:

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

Sporządzić wykresy funkcji *sin(x)* oraz *cos(x)* dla *x* w zakresie $< 7, 7 >$

W pierwszym kroku tworzymy wektor argumentów funkcji:

następne wywołujemy polecenia kreślenia wykresów wbudowanych funkcji trygonometrycznych:

Zdefiniujemy funkcję użytkownika o nazwie *gauss* obliczającą wartości rozkładu normalnego. Wykorzystamy do tego celu polecenie *deff* umożliwiające definicję własnej funkcji w tzw. trybie in-line. (Szerzej na temat tworzenia własnych funkcji będzie później, w części dotyczącej instrukcji programistycznych).

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

Następnie możemy narysować wykres tak zdefiniowanej funkcji:

Proszę zwrócić uwagę na zastosowane parametry sterujące wyglądem uzyskiwanych wykresów (rodzaj linii, kolor, rodzaj markera).

W wielu opracowaniach często zachodzi konieczność prezentowania danych na wykresach typu histogram.

Tego typu wykresy są graficzną prezentacją jednowymiarowych danych (zazwyczaj w postaci prostokątów/słupków). Scilab oferuje funkcje do tworzenia wykresów histogramowych o wyglądzie ‘płaskim’ oraz trójwymiarowym. Mają tu zastosowanie funkcje: *histplot()*, *bar()*, *barh()*, *bar3d()*. Działanie wymienionych funkcji przedstawione zostaną na przykładach. Aby zapoznać się ze szczegółami składni skorzystaj z pomocy programu.

Przykłady:

Tworzymy wektor z danymi do zaprezentowania na histogramie (wektor z 1000 liczb losowych z rozkładem normalnym).

Tworzymy histogram dla 20-u przedziałów danych

Tworzymy prezentację słupkową pierwszych 10-u elementów wektora *dane*

```
-->clf
-->bar(dane(1:10))
-->clf
-->barh(dane(1:10))
```

Tworzymy macierz losową 6x3 z danymi do zaprezentowania na trójwymiarowym wykresie słupkowym.

```
-->dane3d=5*rand(6,3)
```

Tworzymy prezentację słupkową macierzy *dane3d*

```
-->clf
-->bar3d(dane3d)
```

Rysowanie wykresów na wielu układach współrzędnych w jednym oknie graficznym – polecenie *subplot*

Polecenie *subplot(m,n,p)* dzieli okno graficzne na macierz $m \times n$ sub-okien oraz wyznacza sub-okno o numerze p do rysowania bieżącego wykresu. Umożliwia to tworzenie wielu wykresów z różnymi układami współrzędnych w jednym oknie graficznym. Sub-okna są zliczane wierszowo.

Na rysunku poniżej przedstawiono układ sub-okien 2x2 ($m=2, n=2$) z oznaczoną numeracją sub-okien.

Np. polecenie *subplot(2,2,3)* spowoduje aktywowanie sub-okna o numerze 3. Polecenie *plot2d/plot* spowoduje narysowanie wykresu w aktywnym sub-oknie. Poniżej przykład rysowania wykresów w sub-oknach z pokazaniem innych typów wykresów 2d.

Przykłady:

25

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

```
--> x1=linspace(-3,3,101)
--> y1=exp(-(x1.*x1))
--> subplot(2,2,1)
--> plot2d(x1,y1) //normalny wykres
--> subplot(2,2,2)
--> plot2d2(x1,y1,3) //wykres stały pomiędzy punktami (schodkowy), kolor 3-zielony
--> subplot(2,2,3)
--> plot2d3(x1,y1,5) //wykres słupkowy, kolor 5-czerwony
--> subplot(2,2,4)
--> plot2d4(x1,y1,2) //wykres „strzałkowy”, kolor 2-niebieski
```

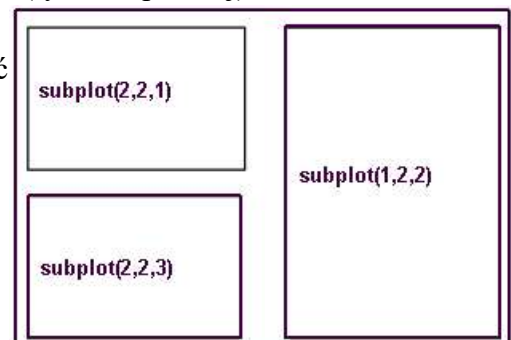
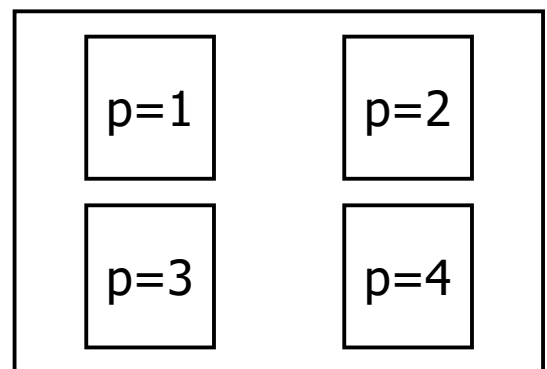
Zapisać wykres do pliku *W2.scg*

Kolejność aktywowania sub-okien może być dowolna. Pierwsze wydane polecenie *subplot* spowoduje otwarcie okna graficznego i aktywowanie w otwartym oknie obszaru odpowiadającego parametrowi p . Kolejne polecenia *subplot* aktywują odpowiednie obszary we wcześniej otwartym oknie graficznym. W jednym oknie graficznym można aktywować obszary z różnych układów sub-okien (rysunek poniżej).

Przykład:

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

```
--> x=linspace(-3,3,40);
--> y=linspace(-5,5,40);
--> [X,Y]=meshgrid(x,y);
--> Z=(2*X.^2-Y.^2).*exp(-X.^2-0.5*Y.^2);
--> subplot(2,2,1)
```



```
--> surf(X,Y,Z) // o wykresach 3d będzie w dalszej części
--> subplot(223)
--> contour(x,y,Z,10)
--> subplot(122)
--> plot2d(x,Z)
```

Zapisać wykres do pliku *W3.scg*

Scilab oferuje również inną funkcję do tworzenia sub-okien okna graficznego: *xsetech()*. Skorzystaj z pomocy aby uzyskać więcej informacji.

Uwaga: zamiast polecenia *plot2d* można stosować polecenie *plot*.

Wykresy trójwymiarowe

Podstawowym poleceniem jest: *plot3d(wektorX,wektorY,wartości)*,

wektor *wektorX* ma długość n_x , wektor *wektorY* ma długość n_y (liczby n_x , n_y nie muszą być równe), *wartości* jest macierzą o rozmiarze $n_x \times n_y$. Składnia polecenia *plot3d()* jest podobna do składni *plot2d()*. Aby poznać szczegóły dotyczące parametrów funkcji *plot3d()* należy skorzystać z pomocy programu.

Przykład:

Sporządzić wykres funkcji $f(x, y) = x^2 + y^2$ w zakresie zmiennych $x, y: [-3, 3]$.

Tworzymy wektory x, y z wartościami zmiennych;

```
--> x=linspace(-3,3,51)
```

```
--> y=linspace(-3,3,61)
```

Tworzymy macierze xx oraz yy poprzez replikowanie wektorów x i y korzystając z polecenia *ndgrid*.

```
--> [xx,yy]=ndgrid(x,y);
```

Macierz xx powstaje poprzez replikowanie wektora x w n_y -kolumnach a macierz yy powstaje poprzez replikowanie wektora y w n_x wierszach.

Następnie obliczamy wartości funkcji:

```
--> z=xx.*xx+yy.*yy;
```

i rysujemy wykres:

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

```
--> plot3d(x,y,z)
```

Zapisać wykres do pliku *W4.scg*

Innym sposobem tworzenia wykresów trójwymiarowych jest zastosowanie funkcji *surf()*.

Przykład:

Sporządzić wykres funkcji $f(x, y) = (2x^2 - y^2) \cdot \exp(-x^2 - 0.5y^2)$, w zakresie zmiennych $x[-4, 4]$, $y[-3, 3]$.

Tworzymy wektory x, y z wartościami zmiennych;

```
--> x=linspace(-4,4,40)
```

```
--> y=linspace(-3,3,40)
```

Tworzymy macierze xx oraz yy poprzez replikowanie wektorów x i y korzystając z polecenia *ndgrid* lub *meshgrid*

```
--> [X,Y]=ndgrid(x,y);
```

```
--> [Xm,Ym]=meshgrid(x,y);
```

Następnie obliczamy wartości funkcji:

```
--> Z=(2*X.^2-Y.^2).*exp(-X.^2-0.5*Y.^2);
```

```
--> Zm=(2*X.^2-Y.^2).*exp(-X.^2-0.5*Y.^2);
```

i rysujemy wykres w dwóch oddzielnych oknach graficznych:

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

```
-->surf(X,Y,Z)
```

Zapisać wykres do pliku *W5.scg*

```
-->scf()
```

```
-->surf(Xm,Ym,Zm)
```

Wykresy krzywych w przestrzeni 3D.

Do wykreślania pojedynczej krzywej w przestrzeni trójwymiarowej służy funkcja *param3d(x,y,z)*, gdzie *x*, *y*, *z* są wektorami o takim samym rozmiarze. Elementy wektorów z tych samych pozycji opisują punkty w przestrzeni (są współrzędnymi punktów w przestrzeni).

Przykład: rysowanie linii śrubowej.

```
--> t=linspace(0,4*%pi,101);
```

```
--> x=2*cos(t);
```

```
--> y=2*sin(t);
```

```
--> z=4*t;
```

Zamknąć okno graficzne lub „wyczyścić” jego zawartość poleceniem *clf*.

```
--> param3d(x,y,z)
```

```
-->scf()
```

```
--> param3d(x,y,zeros(z))
```

Zapisać wykres do pliku *W6.scg*