Automatyzacja Obliczeń Inżynierskich

Laboratorium

MATLAB

Opracowanie:
dr inż. Jacek Kucharski
dr inż. Piotr Urbanek

ELEMENTY GRAFICZNE W MATLAB-ie

GRAFIKA DWUWYMIAROWA

Podstawową funkcją służącą do rysowania wykresów dwuwymiarowych jest funkcja plot. Przykładowy sposób jej wywołania to

plot(x,y,s)

gdzie x i y to wektory odwzorowujące funkcje y=*f*(x) a s - łańcuch znaków określający kolor i rodzaj linii.

W celu utworzenia dwóch osi rzędnych należy wykorzystać funkcję plotyy

Więcej informacji znajduje się w pliku pomocy środowiska Matlab w dziale podstawowe sposoby rysowania

Wywołanie tej funkcji powoduje otwarcie okna graficznego i umieszczenie w nim wykresu w skalach liniowych. **Skale logarytmiczne i półlogarytmiczne** można uzyskać odmianami funkcji plot: **loglog(), semilogx(), semilogy()** o tych samych argumentach.

Można pracować z wieloma oknami graficznymi tworząc kolejne poleceniem figure. Jedno z nich jest zawsze aktywne i wszystkie bieżące operacje graficzne dotyczą tego właśnie okna. Okno aktywne można oczyścić poleceniem clg, a dowolne okno graficzne (także zbiór okien) można zamknąć poleceniem close akceptującym jako parametr wektor numerów okien do zamknięcia.

Czasami wygodne jest umieszczenie kilku wykresów obok siebie w jednym oknie. Można to uczynić posługując się funkcją subplot. Istnieją dwa sposoby wywołania tej funkcji:

subplot(m,n,p)

dzieli aktywny rysunek na m w poziomie i n w pionie części oraz uaktywnia p-ty z utworzonych rysunków (m, n, p muszą być naturalne z przedziału <1,9>),

subplot('position',[współrzędne_lewego_dolnego, szerokość, wysokość])

tworzy w obrębie aktywnego rysunku nowy układ współrzędnych o podanym położeniu i wymiarach (np. subplot('position',[0.3 0 0.6 0.6]) tworzy układ współrzędnych w prawym dolnym rogu okna na 60% wysokości i szerokości okna)

Skalę wykresu można zmienić poleceniem axis. Wymaga ono jednego parametru - czteroelementowego wektora zawierającego zakresy poszczególnych skal [*xmin xmax ymin ymax*]. Argumentem tej funkcji może być również łańcuch znaków zmieniający tryby skalowania.

„Zatrzymanie" dotychczasowego rysunku w bieżącym oknie uzyskuje się poleceniem hold on|off. Funkcja ishold zwraca jedynkę gdy tryb jest włączony.

Do opisywania rysunku służą funkcje:

```
title('text')
xlabel('text')
ylabel('text')
text(x,y,'text')
```

Dodatkowo można umieścić pomocniczą siatkę współrzędnych poleceniem grid on|off.

Istnieją też inne rodzaje wykresów: we współrzędnych biegunowych, słupkowy itp.

GRAFIKA TRÓJWYMIAROWA

Wykresy trójwymiarowe wymagają specjalnego przygotowania danych. Do narysowania powierzchni konieczne są trzy macierze X, Y, Z, na podstawie których wyznaczane są współrzędne (*x,y,z*) poszczególnych punktów powierzchni w przestrzeni: x=X(i,j), y=Y(i,j), z=Z(i,j) (gdzie i,j są indeksami macierzy). Pomocną przy tworzeniu tych macierzy jest funkcja meshgrid. Typowy sposób wykorzystania tej funkcji w przypadku powierzchni *z=f(x,y)* ma postać:

```
[X,Y]= meshgrid (x,y);
Z=X.^2-Y.^2;
```

W pierwszym kroku powstają macierze X i Y złożone z odpowiednio powielonych wektorów x i y, a następnie budowana jest macierz Z zawierająca wartości funkcji *z=f(x,y)* dla poszczególnych argumentów.
Tak przygotowane dane mogą być wykreślone przy pomocy jednej z dostępnych funkcji:

mesh(X,Y,Z)
meshc(X,Y,Z)
meshz(X,Y,Z)
surf(X,Y,Z)
surfc(X,Y,Z)
surfl(X,Y,Z)
waterfall(X,Y,Z)

Funkcja view pozwala **na zmianę miejsca, z którego oglądamy rysunek**. Jej parametrami mogą być az i el (azymut i elewacja) lub x,y,z (współrzędne punktu obserwacji). Można też wykorzystać gotowe dane wpisując jako parametr 2 (obserwacja dwuwymiarowa), 3 (standardowa obserwacja trójwymiarowa).

Zmiana sposobu kolorowania powierzchni jest możliwa przy pomocy polecenia shading:
shading flat (domyślny)
shading interp
shading faceted

Opisywanie wykresów trójwymiarowych jest podobne jak w 2D tylko z uwzględnieniem trzeciej współrzędnej.

WYKRESY POZIOMICOWE

Wykres poziomicowy można uzyskać za pomocą funkcji contour(X,Y,Z). Dodatkowym parametrem może być n-liczba poziomic, lub wektor v z wysokościami dla kolejnych poziomic.
Jeżeli podany zostanie parametr wyjściowy tj.;
c=contour(X,Y,Z,v)

to funkcja zwróci macierz wartości wykorzystywaną przez funkcję clabel(c) do opisu tych poziomic na wykresie.

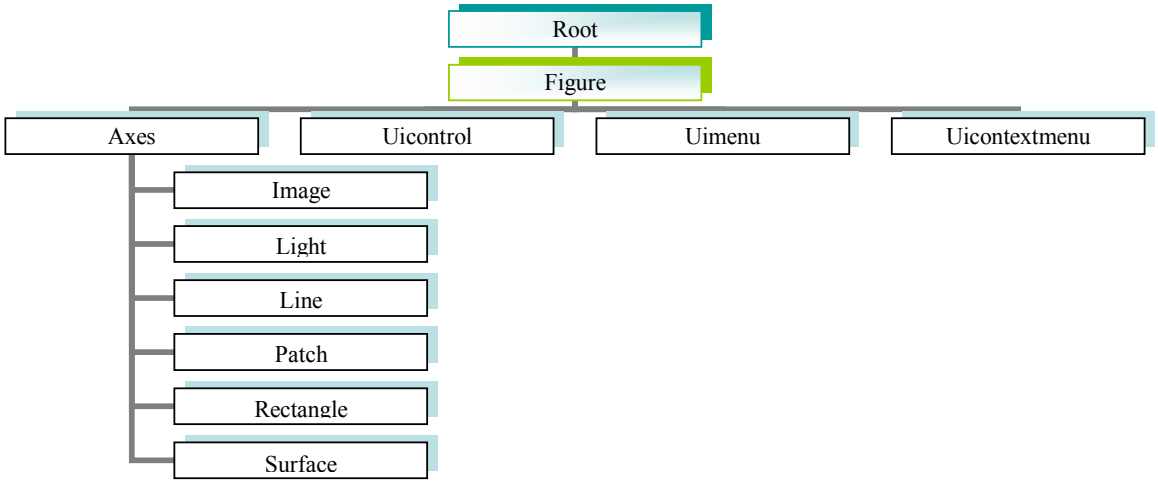
OBIEKTOWY SYSTEM GRAFICZNY

Funkcje graficzne w MATLABie są funkcjami **wysokiego poziomu**. Tworzą gotowe rysunki **nie dając zazwyczaj użytkownikowi możliwości ich modyfikacji**. Każdy fragment rysunku stanowi jednak pewien obiekt graficzny i ma przypisany swój unikalny *identyfikator* (handle) będący liczbą rzeczywistą.

Każdy obiekt zawiera strukturę danych – rekord, w którym przechowywane są jego parametry. Pola rekordu są własnościami obiektu czyli zmiennymi (liczbami, macierzami, łańcuchami znaków, zmiennymi typu wyliczeniowego) określającymi jakiś parametr jego wyglądu, sposób działania lub powiązania z innymi obiektami. Każdy typ obiektu posiada odrębny zestaw własności (pewne własności są wspólne dla wszystkich typów)

Obiekty powiązane są ze sobą w sposób hierarchiczny (drzewiasty) i obiekt ma zawsze jednego przodka i dowolną ilość potomków. Korzeniem drzewa (root) jest ekran komputera i nie ma przodka, a jego identyfikatorem jest liczba 0. Jego potomkami są okna graficzne – rysunki (figures), te z kolei mają potomków w postaci układów współrzędnych (axes) oraz elementów graficznego systemu komunikacji z użytkownikiem, tj. przycisków suwaków menu itp. (tworzonych przez funkcje uicontrol uimenu i uicontextmenu). Potomkami układów mogą być linie, powierzchnie teksty, obrazy i płaty. **Taka hierarchia musi być przestrzegana**.

Pełne informacje o hierarchii obiektów oraz ich własnościach można znaleźć w pliku pomocy środowiska Matlab



Operacje na obiektach można podzielić na dwie grupy:

- tworzenie i usuwanie obiektów,
- zmiany w rekordzie danych związanych z obiektem

Do odczytywania i zmian własności obiektów służą funkcje get i set:

get (id) – wyświetla listę własności obiektu i ich wartość

wartość = get(id, nazwa_własności) – zwraca wartość wyspecyfikowanej własności.

id – identyfikator obiektu lub wektor identyfikatorów,

nazwa_własności – ciąg znaków zawierający nazwę własności **lub jej jednoznaczny początek**

set (id) – wyświetla listę własności obiektu i ich możliwe wartości

set(id, nazwa_własności, wartość) – zmienia wartość wyspecyfikowanej własności na podaną.

reset(id) – przywraca standardowe własności obiektu.

Do usuwania obiektów służą:

delete(id) – usuwa obiekt *id* wraz z jego wszystkimi potomkami,
close(id) – usuwa obiekt *id*,
close - zamyka aktywne okno graficzne,
clf – czyści aktywne okno graficzne,
cla – czyści obiekty z aktywnego układu współrzędnych.

Do tworzenia obiektów służą oprócz funkcji wysokiego poziomu również funkcje o nazwach identycznych jak nazwy typów obiektów. Wszystkie funkcje zwracają identyfikatory tworzonych obiektów.

id=figure – tworzy okno rysunku i zwraca jego identyfikator,
id=axes('position',[lewy,dolny,szerokość,wysokość]) – tworzy układ współrzędnych,
id=line(x,y,z,nazwa_własność1, wartość1, nazwa_własność2, wartość2,...) -
id=text(x,y,z,nazwa_własność1, wartość1, nazwa_własność2, wartość2,...)
id=patch(x,y,z,c,nazwa_własność1, wartość1, nazwa_własność2, wartość2,...)
id=surface(x,y,z,c,nazwa_własność1, wartość1, nazwa_własność2, wartość2,...)
id=image(x,y,c)

Obiekty aktywne:
Obiektem aktywnym jest obiekt w obrębie którego ostatnio kliknięto myszką. W danej chwili tylko jeden obiekt może być aktywnym.

id=gco – zwraca identyfikator aktywnego obiektu w aktywnym rysunku,
id-gcf – podaje identyfikator aktywnego rysunku,
id=gca – podaje identyfikator aktywnego układu współrzędnych,

GRAFICZNY SYSTEM KOMUNIKACJI Z UŻYTKOWNIKIEM

Koncepcja graficznego systemu komunikacji z użytkownikiem polega na realizacji dwóch zasad:

- budowaniu wyglądu aplikacji z prostych gotowych elementów mających intuicyjną interpretację i sugestywny wygląd (korzysta się z gotowych obiektów typu przyciski, suwaki, przełączniki)
- tworzeniu aplikacji o działaniu sterowanym zdarzeniami.

  Każdy taki obiekt ma własność o nazwie 'CallBack', pod którą podstawia się nazwę procedury (m-pliku), która ma być wykonana po wystąpieniu dopuszczalnej dla danego obiektu akcji.

Tworzenie elementów graficznego systemu komunikacji jest oparte na kilku funkcjach:

id=uicontrol(idf,nazwa_własność1, wartość1, nazwa_własność2, wartość2,...)

Tworzy w rysunku *idf* nowy obiekt, którego rodzaj określa się we własności 'Style', a w kolejnych parach (własność – wartość) określa się jego cechy.

id=uimenu(idf,nazwa_własność1, wartość1, nazwa_własność2, wartość2,...)

Tworzy w głównym menu rysunku *idf* dodatkowe menu lub jeżeli *idf* jest identyfikatorem istniejącego menu to tworzy dla niego podmenu. Nazwę tworzonego menu określa się we własności 'label', a w kolejnych parach (własność – wartość) określa się inne cechy.

Wszystkie powstałe w ten sposób elementy są potomkami obiektów typu *figure*. Ich własności mogą być zmieniane tak samo jak innych obiektów przy użyciu funkcji get i set.

id = uicontextmenu – funkcja wywołująca okno z wartościami umieszczonych obiektów,

answer = inputdlg – generuje okienko dialogowe do wprowadzania danych. Wraz z tą funkcją używa się również funkcji deal, której zadaniem jest przypisanie wejść do wyjść.

Program ćwiczenia

1.Zbudować m-plik skryptowy realizujący następujące operacje graficzne:

- rysowanie wykresu funkcji jednej zmiennej, t.j. $y=f(x)$, (należy wprowadzić zakres wartości zmiennej x tzn. x_{min} i x_{max} , oraz nazwę funkcji w wierszu poleceń)
- rysowanie wykresu wzajemnej zależności danych liczbowych zapisanych w wektorach X i Y, t.j. zbioru punktów (x_i , y_i): $x_i \in X$, $y_i \in Y$, (wektory X i Y należy wprowadzać w wierszu poleceń)
- opisanie otrzymanych wykresów dwuwymiarowych za pomocą okna dialogowego (funkcja inputdlg),
- rysowanie wykresu 3D wybranej funkcji dwóch zmiennych, t.j. $z=f(x,y)$, (niezbędne dane należy wprowadzać w wierszu poleceń),
- zmianę punktu obserwacji otrzymanego wykresu trójwymiarowego za pośrednictwem okna dialogowego,
- rysowanie wykresu poziomicowego odpowiadającego utworzonemu wcześniej wykresowi trójwymiarowemu.

2. Zbudować m-plik skryptowy tworzący interakcyjne narzędzie ułatwiające prezentację wykresów dwuwymiarowych poprzez wykorzystanie obiektowego, graficznego systemu komunikacji z użytkownikiem. Plik ten powinien otwierać nowe okno graficzne zawierające następujące elementy:

- opcje menu głównego umożliwiające:
  - wybór liczby osi y wykresu,
  - wprowadzanie danych,
- przycisk uruchamiający rysowanie wykresu,
- odpowiednio rozmieszczone suwaki umożliwiające zmianę wartości początkowej i końcowej osi X na wykresie,

Dodatkowo program powinien umożliwiać wykonywanie następujących czynności:

- sekwencyjną zmianę rodzaju linii wykresu na skutek kolejnych kliknięć lewym przyciskiem myszy na wykresie,
- zmianę koloru linii wykresu za pomocą menu kontekstowego uruchamianego prawym przyciskiem myszy,
- włączanie i wyłączanie siatki na skutek kliknięć lewym przyciskiem myszy w obszar wykresu.