

**INSTRUKCJE DO LABORATORIUM
Z PODSTAW CYFROWEGO PRZETWARZANIA SYGNAŁÓW**

DODATEK

A. Leśnicki, C. Stefański

PRZEGLĄD WAŻNIEJSZYCH INSTRUKCJI MATLABA

**CYFROWE PRZETWARZANIE SYGNAŁÓW
- WYBRANE PROCEDURY BIBLIOTEK MATLABA**

GRAFIKA W MATLABIE

W Dodatku umieszczono wybrane instrukcje i procedury. Gdy instrukcji lub procedury nie ma w Dodatku, jej opis można wyświetlić pod MATLABem korzystając z polecenia **help** (opisanego bezpośrednio poniżej).

D.1 WYBRANE INSTRUKCJE MATLABA

help

pomóż

wyświetlając opis funkcji, operatora lub definicji terminu określonego w pakiecie MATLAB. Użycie instrukcji **help** bez parametru spowoduje wyświetlenie listy wszystkich tematów, których opis jest dostępny. Lista ta składa się z - nazw operatorów i funkcji wbudowanych pakietu MATLAB, - nazw funkcji zewnętrznych (m-funkcji) pakietu, zarówno należących do pakietu, jak i zdefiniowanych przez użytkownika. Wyświetlane są nazwy m-funkcji znajdujących się w katalogach wymienionych w zmiennej środowiskowej MATLABPATH. Instrukcja **help temat** podaje bardziej szczegółowe objaśnienia dotyczące słowa *temat* (słowo to może być nazwą funkcji wbudowanej, kodem operatora lub pojęciem określonym w systemie MATLAB). Podobnie, użycie instrukcji **help funkcja** spowoduje wyświetlenie pierwszych wierszy komentarza napotkanych w pliku o nazwie *funkcja* (o ile taki plik istnieje w którymś z podkatalogów wymienionych w ścieżce zdefiniowanej przez zmienną środowiskową MATLABPATH).

[]

nawiasy kwadratowe: tworzenie wektorów i macierzy (tablic)

Nawiasy kwadratowe są używane do tworzenia wektorów i macierzy. Na przykład **[11 12 13; 21 22 23]** jest macierzą o dwóch wierszach i trzech kolumnach. Elementy wiersza oddzielane są spacjami lub przecinakami, kolejne wiersze kończą się średnikiem. Następujące zapisy wektora o trzech elementach są równoważne: **[1 1.25 sqrt(-1)]** lub **[1, 1.25, sqrt(-1)]**. Nie są równoważne zapisy **[1+i 2 -i3]** oraz **[1 +i 2 -i 3]**. Pierwszy z nich jest wektorem trzejelementowym, drugi - pięcioelementowym. Wewnątrz nawiasów kwadratowych mogą wystąpić wektory i macierze. Pozwala to na budowę macierzy z klatek. Na przykład **[A B; C]** jest macierzą, której pierwsze wiersze powstały z zestawienia obok siebie macierzy **A** i **B**, a pozostałe wiersze zostały przeniesione z macierzy **C**. Powyższa instrukcja zostanie wykonana tylko wtedy, gdy: liczby wierszy macierzy **A** i **B** są równe i suma liczby kolumn macierzy **A** i liczby kolumn macierzy **B** jest równa liczbie kolumn macierzy **C**. **[]** jest macierzą zerową (o wymiarze zero).

()

nawiasy okrągłe

Określają kolejność wykonywania działań arytmetycznych w zwyczajowo przyjęty sposób. Używane są także podczas wywołania funkcji - otaczają listę argumentów funkcji. Służą do oznaczania indeksów do wektorów i tablic, przy czym przyjmuje się, że **indeksy wektorów i tablic rozpoczynają się od jedności**: **X(1)** jest pierwszym elementem wektora **X**, **A(3,2)** to drugi element trzeciego wiersza tablicy **A**. Jeśli **X** i **V** są wektorami, to **X(V)** jest równoważne następującej konstrukcji: **[X(V(1)), X(V(2)), ..., X(V(M))]** (składowe wektora **V** są zaokrąglane do najbliższych liczb całkowitych i traktowane jako indeksy). Jeśli dowolny z tak wyznaczonych indeksów jest mniejszy od jedności lub większy od rozmiaru wektora **X**, to wystąpi błąd. Na przykład: **X(3)** jest trzecim elementem wektora **X**, **X([1 2 3])** jest wektorem składającym się z pierwszych trzech elementów wektora **X**; konstrukcja ta będzie miała identyczny rezultat jak konstrukcja **X([sqrt(2), sqrt(3), 4*atan(1)])**. Jeśli **X** ma **N** składowych, to **X(N:-1:1)** odwraca ich kolejność. Jeśli wektor **V** ma **M** elementów, a wektor **W** ma **N** elementów, to wyrażenie **A(V,W)** da w wyniku macierz o wymiarze **MxN**, złożoną z tych elementów **A**, których numery wierszy są określone przez wektor **V**, a numery kolumn - przez wektor **W**. Dla przykładu, instrukcja **A([1,5],:)=A([5,1],:)** zamienia wiersze o numerach 1 i 5 w macierzy **A**.

.

kropka dziesiętna

Kropka dziesiętna służy do oddzielania części ułamkowej liczby dziesiętnej: 314/100, 3.14 oraz .314e1 oznaczają jedną i tę samą liczbę. Kropka umieszczona przed takimi operatorami multiplikatywnymi jak: *****, **^**, **/**, ****, **'** oznacza działania na kolejnych elementach tablic. Na przykład: **C=A./B** jest tablicą, której elementy zostały wyznaczone z zależności **c(i,j)=a(i,j)/b(i,j)**. Dwie lub więcej kropek na końcu linii oznacza, że następną linię należy traktować jako linię kontynuacji.

,

przecinek

oddziela indeksy tablic, argumenty funkcji oraz wyrażenia zapisane w jednym wierszu. W tym ostatnim przypadku można go zastąpić średnikiem, jeżeli chcemy uniknąć wyświetlania wyników tych wyrażeń.

- ;** **średnik**
służy do oddzielania kolejnych wierszy macierzy podczas jej definiowania za pomocą wyrażeń umieszczonych w nawiasach kwadratowych []. Umieszczony na końcu wyrażenia blokuje wydruk wyników tego wyrażenia.
- %** **znak procentu**
oznacza logiczny koniec linii - tekst następujący po tym znaku aż do końca linii jest traktowany jako komentarz.
- !** **wyjście do systemu operacyjnego**
Użycie wykrzyknika spowoduje chwilowe zawieszenie pracy programu MATLAB i wyjście do systemu operacyjnego. Tekst występujący po wykrzykniku jest przekazywany do systemu operacyjnego jako polecenie do wykonania. Na przykład instrukcja `!edit fun.m` spowoduje chwilowe wyjście z programu MATLAB, wywołanie edytora tekstowego `edit` z parametrem `fun.m`. Po zakończeniu pracy w edytorze nastąpi automatyczny powrót do programu MATLAB.
- :** **dwukropek**
służy do konstrukcji wektorów o liniowo rosnących lub malejących elementach. Na przykład $\mathbf{J:K}$ jest równoważne wyrażeniu $[\mathbf{J}, \mathbf{J+1}, \dots, \mathbf{K}]$ (jeżeli $\mathbf{J} > \mathbf{K}$, to $\mathbf{J:K}$ jest wektorem pustym). $\mathbf{J:I:K}$ jest równoważne wyrażeniu $[\mathbf{J}, \mathbf{J+I}, \mathbf{J+2*I}, \dots, \mathbf{K}]$ (jeżeli $\mathbf{I} > 0$ i jednocześnie $\mathbf{J} > \mathbf{K}$ lub jeżeli $\mathbf{I} < 0$ i jednocześnie $\mathbf{J} < \mathbf{K}$, to $\mathbf{J:I:K}$ jest wektorem pustym). Użyty jako jeden z indeksów oznacza wybór całego wiersza lub kolumny. Dla przykładu, wyrażenie $\mathbf{A(2,:)}$ daje w wyniku wektor - drugi wiersz macierzy \mathbf{A} . Wyrażenie $\mathbf{A(:,I:J)}$ zwraca podmacierz składającą się z kolumn od \mathbf{I} -tej do \mathbf{J} -tej macierzy \mathbf{A} . Jeżeli \mathbf{A} jest macierzą, to wyrażenie $\mathbf{A(:)}$ da w wyniku wektor kolumnowy, składający się ze wszystkich elementów macierzy \mathbf{A} (w porządku wierszowym). Użycie konstrukcji $\mathbf{A(:)}$ po lewej stronie operatora przypisania spowoduje wypełnienie macierzy \mathbf{A} , z zachowaniem jednocześnie jej poprzedniego kształtu. Zobacz także **for**.
- '** **transpozycja macierzy lub cudzysłów**
 $\mathbf{X'}$ oznacza zespoloną i sprzężoną transpozycję macierzy lub wektora \mathbf{X} , $\mathbf{X'}$ oznacza zwykłą transpozycję (bez sprzężenia), wyrażenie **'To jest napis'** jest wektorem, którego składowymi są kody ASCII kolejnych znaków napisu zawartego między znakami cudzysłowu. Jeżeli wewnątrz tego napisu miałyby występować znak cudzysłowu, to należy go użyć dwa razy.
- +** **dodawanie**
Wyrażenie $\mathbf{X+Y}$ oznacza sumę macierzy \mathbf{X} i \mathbf{Y} . Macierze \mathbf{X} i \mathbf{Y} muszą mieć takie same wymiary. Wyrażenie $\mathbf{X+a}$, gdzie \mathbf{X} jest macierzą, \mathbf{a} - skalar, daje w wyniku macierz otrzymaną przez dodanie \mathbf{a} do każdego elementu \mathbf{X} .
- **odejmowanie**
Wyrażenie $\mathbf{X-Y}$ oznacza różnicę macierzy \mathbf{X} i \mathbf{Y} . Macierze \mathbf{X} i \mathbf{Y} muszą mieć takie same wymiary. Wyrażenie $\mathbf{X-a}$, gdzie \mathbf{X} jest macierzą, \mathbf{a} - skalar, daje w wyniku macierz otrzymaną przez odjęcie \mathbf{a} od każdego elementu \mathbf{X} .
- *** **mnożenie**
Wyrażenie $\mathbf{X*Y}$ oznacza iloczyn macierzowy \mathbf{X} i \mathbf{Y} . Wymiary obu macierzy muszą odpowiadać regułom mnożenia macierzowego, tj. liczba kolumn macierzy \mathbf{X} musi być równa liczbie wierszy macierzy \mathbf{Y} . Wyrażenie $\mathbf{a*X}$ lub $\mathbf{X*a}$, gdzie \mathbf{X} jest macierzą, \mathbf{a} - skalar, daje w wyniku macierz otrzymaną przez przemnożenie każdego elementu macierzy \mathbf{X} przez skalar \mathbf{a} . Wyrażenie $\mathbf{X.*Y}$ oznacza mnożenie typu element przez element. Macierze \mathbf{X} i \mathbf{Y} muszą wówczas mieć takie same wymiary (chyba, że jedna z nich jest skalar).
- ** **lewostronne dzielenie macierzy**
Wyrażenie $\mathbf{A\B}$ oznacza macierzowe dzielenie \mathbf{B} przez \mathbf{A} . Wynik jest równy $\mathbf{inv(A)*B}$, tyle że jest obliczany inaczej. Jeśli \mathbf{A} jest macierzą $\mathbf{N \times N}$, a \mathbf{B} jest wektorem kolumnowym o \mathbf{N} elementach lub macierzą złożoną z kilku takich wektorów, to $\mathbf{X=A\B}$ jest wyznaczane jako rozwiązanie równania $\mathbf{A*X=B}$ (metodą eliminacji Gaussa). Jeżeli macierz \mathbf{A} jest źle uwarunkowana lub prawie osobliwa, to na ekranie monitora zostanie wyświetlone ostrzeżenie. Wyrażenie $\mathbf{A\eye(A)}$ daje w wyniku macierz odwrotną do \mathbf{A} . Jeśli \mathbf{A} jest macierzą o wymiarach $\mathbf{M \times N}$, gdzie $\mathbf{M \neq N}$, a \mathbf{B} jest wektorem kolumnowym o \mathbf{N} elementach (lub macierzą złożoną z kilku takich wektorów), to w instrukcji $\mathbf{X=A\B}$ macierz \mathbf{X} jest obliczana jako rozwiązanie, w sensie najmniejszych kwadratów, nieokreślonego lub sprzecznego układu równań $\mathbf{A*X=B}$. Efektywny rząd \mathbf{K} macierzy \mathbf{A} jest określany na podstawie dekompozycji QR z wyborem elementów głównych. Wyznaczane jest takie rozwiązanie \mathbf{X} , które da co najwyżej \mathbf{K} niezerowych składowych w kolumnie. W przypadku gdy $\mathbf{K < N}$ rezultat będzie zazwyczaj inny niż zwracany przez wyrażenie $\mathbf{pinv(A)*B}$. Wyrażenie $\mathbf{A\eye(A)}$ daje w wyniku uogólnioną odwrotność macierzy \mathbf{A} . Zobacz także **/** w odróżnieniu operatora ****.

/

prawostronne dzielenie macierzy

Wyrażenie B/A oznacza macierzowe dzielenie B przez A . Wynik jest równy $\text{inv}(A)*B$, tyle, że jest obliczany inaczej. Dokładniej, wyrażenie B/A jest równoważne wyrażeniu $(A'\backslash B')$ (patrz opis operatora \backslash). Wyrażenie $B./A$ oznacza dzielenie typu element przez element. W tym przypadku macierze A i B muszą mieć takie same wymiary (chyba że jedna z nich jest skalar). **Uwaga.** Wyrażenie $3./A$ (kropka bezpośrednio po cyfrze) NIE oznacza tego samego, co wyrażenie $3./A$ (kropka oddzielona odstępem od cyfry). W pierwszym przypadku kropka zostanie potraktowana jako kropka dziesiętna, w drugim przypadku - wraz ze znakiem $/$ - będzie operatorem dzielenia typu element przez element.

^

potęgowanie.

Instrukcja $Z=X^a$, gdzie a jest skalar, daje w wyniku podniesioną do potęgi a macierz (lub skalar) X . Jeśli a jest liczbą całkowitą > 1 , to potęgę tę oblicza się przez wielokrotne mnożenie. Dla pozostałych wartości a w obliczeniach wykorzystuje się wartości i wektory własne. Instrukcja $Z=a^Y$, gdzie a jest skalar, a Y macierzą, daje w wyniku a podniesione do potęgi Y . Wyrażenie $Z=X^Y$, gdzie zarówno X jak i Y są macierzami, jest niepoprawne. Wyrażenie $Z=X.^Y$ oznacza potęgowanie typu element przez element. Macierze X i Y muszą w takim przypadku mieć takie same wymiary (chyba że jedna z nich jest skalar).

<

<=

>

>=

==

~=

operatory relacji

Wyrażenie $C=A\text{opr} B$, gdzie opr jest operatorem relacji ($<$ mniejsze niż, $>$ większe niż, $<=$ mniejsze lub równe, $>=$ większe lub równe, $==$ równe, $~=$ różne od) umożliwia porównanie macierzy A i B element po elemencie i zwraca macierz C o tych samych wymiarach co A i B i elementach równych 1 na pozycjach, dla których relacja opr jest spełniona oraz równych 0, tam gdzie nie jest spełniona. Macierze A i B muszą mieć takie same wymiary (chyba że jedna z nich jest skalar; w takim przypadku skalar ten jest porównywalny z każdym elementem drugiej macierzy).

&

|

~

operatory logiczne

Wyrażenie $C=A\text{opl} B$, gdzie opl jest operatorem logicznym ($\&$ iloczyn logiczny, $|$ suma logiczna, \sim negacja) daje w wyniku macierz C , której elementy są jedynkami tam, gdzie odpowiadające sobie elementy macierzy A i B spełniają relację logiczną opl , a zerami w przeciwnym przypadku. Macierze A i B muszą mieć takie same wymiary (chyba że jedna z nich jest skalar; w takim przypadku skalar ten jest porównywany z każdym elementem drugiej macierzy). Przy wyznaczaniu wyniku operacji logicznych elementy o wartości 0 są traktowane jako fałsz (wartość logiczna 0), a elementy różne od zera jako prawda (wartość logiczna 1).

abs

wartość bezwzględna

abs(X) - wartość bezwzględna elementów macierzy X . Jeżeli X jest macierzą zespoloną, **abs(X)** zwraca moduły (pierwiastek z sumy kwadratów części rzeczywistej i urojonej) kolejnych elementów macierzy X .

all

wszystkie (niezerowe)?

all(V), jeżeli V jest wektorem, zwraca wartość 1, gdy wszystkie elementy tego wektora są różne od zera; w przeciwnym przypadku zwracane jest 0. Jeżeli V jest macierzą, **all(V)** zwraca wektor wierszowy składający się z zer i jedynek w zależności od tego, czy odpowiednia kolumna V zawiera przynajmniej jeden element zerowy, czy nie. Patrz też **any**.

ans

odpowiedź (ANSwer)

Zmienna systemowa przechowująca wartość ostatnio obliczonego wyrażenia, która nie została przypisana innej zmiennej przez użytkownika

any

którykolwiek (niezerowy)?

any(V) - jeżeli V jest wektorem, to funkcja zwraca wartość 1, gdy przynajmniej jeden element tego wektora jest różny od zera; w przeciwnym przypadku zwracane jest 0. Jeżeli V jest macierzą, **any(V)** zwraca wektor wierszowy składający się z zer i jedynek w zależności od tego, czy odpowiednia kolumna V zawiera wyłącznie elementy zerowe, czy nie. Patrz też **all**.

acos

arcus cosinus

acos(X) - funkcja arcus cosinus wszystkich elementów macierzy (skalara) X . Jeśli macierz X zawiera element o wartości bezwzględnej większej od jedności, to otrzymana macierz wynikowa będzie zespolona.

asin	arcus sinus asin(X) - funkcja arcus sinus wszystkich elementów macierzy (skalara) X . Jeśli macierz X zawiera element o wartości bezwzględnej większej od jedności, to otrzymana macierz wynikowa będzie zespolona.
atan	arcus tangens atan(X) - funkcja arcustangens wszystkich elementów macierzy (skalara) X
atan2	arcus tangens dwóch zmiennych atan2(Y, X) czteroćwiartkowa funkcja arcus tangens ilorazu rzeczywistych elementów X i Y .
axis	osie (skaluj osie) Funkcja axis(V) , gdzie V=[Xmin Xmax Ymin Ymax] , ustala skalę wykresów w taki sposób, że oś pozioma rozciąga się od Xmin do Xmax , a oś pionowa - od Ymin do Ymax . Użycie axis bez parametrów przywraca automatyczne skalowanie osi (lub, jeżeli oś była skalowana automatycznie, zachowuje skalę z ostatniego rysunku dla następnych wykresów). Funkcja zwraca czteroelementowy wektor [Xmin,Xmax,Ymin,Ymax] określający skalę na osiach wykresu w momencie wywołania funkcji (umożliwia to jednocześnie ustawienie nowej skali i zapamiętanie starej). Wywołanie axis('square') ustawia skalę na osiach w taki sposób, by jednostki na osi poziomej i pionowej miały w przybliżeniu tę samą długość. Wywołanie axis('normal') przywraca normalny sposób skalowania.
break	przerwij (przerwij wykonywanie pętli programowej)
casesen	rozdzielaj/nie rozdzielaj (dużych i małych liter) Instrukcja casesen off spowoduje, że w nazwach zmiennych i funkcji nie będą rozdzielane małe i duże litery. Instrukcja casesen on włącza rozdzielanie przez MATLAB-a dużych i małych liter w nazwach zmiennych i funkcji. Standardowo po wywołaniu, program MATLAB odróżnia małe i duże litery alfabetu.
ceil(X)	sufit Zaokrąglenie elementów macierzy X do najbliższych liczb całkowitych w kierunku plus nieskończoności
chdir	zmień katalog
clc	wymaż ekran tekstowy zaś zawartość ekranu graficznego pozostaw nie zmienioną.
clear	usuń (z pamięci) Instrukcja clear usuwa wszystkie zmienne z pamięci roboczej, instrukcja clear X usuwa zmienną X (lub skompilowaną M-funkcję o nazwie X) z pamięci operacyjnej. Wywołanie clear functions usuwa z pamięci operacyjnej wszystkie skompilowane M-funkcje.
clg	wymaż ekran graficzny zaś zawartość ekranu tekstowego pozostaw nie zmienioną
clock	zegar (data i czas) Zwraca wektor liczbowy o postaci [rok miesiąc dzień godzina minuta sekunda] zawierający aktualną datę i czas. Sekunda jest określona z dokładnością jednej do dwóch cyfr dziesiętnych po kropce.
conj	sprzężenie zespolone conj(X) - zespolone sprzężenie elementów macierzy X .
contour	wykres warstwicowy contour(Z) – tworzy wykres warstwicowy funkcji dwóch zmiennych, zapamiętanej w macierzy Z . Elementy tej macierzy traktowane są jako wartości współrzędnej wysokości. Element Z(1,1) opisuje punkt leżący w lewym górnym rogu wykresu. Dopuszczalne jest użycie dodatkowych, opcjonalnych argumentów. contour(Z,M) , gdzie M jest składową, wykreśli wykres o M poziomicach (standardowo sama funkcja dobiera liczbę poziomich); contour(Z,V) , gdzie V jest wektorem, wykreśli wykres o length(V) poziomich, umieszczonych na wysokościach określonych przez kolejne elementy wektora V ; contour(Z,N,X,Y) lub contour(Z,V,X,Y) gdzie X i Y są wektorami, określa wartości na osiach X i Y wykresu (wektory X i Y muszą mieć długości równe odpowiednio liczbie kolumn i wierszy macierzy Z); contour(...,'linetype') wykoną wykres za pomocą linii o rodzaju i kolorze określonym przez ciąg znaków linetype , podobnie jak w funkcji plot . Instrukcja C=contour(...) zwraca dwuwierszową macierz opisującą kolejne linie konturów. Każdy ciągły segment wykresu zawiera wysokość konturu, liczbę par (x, y) składających się na kontur oraz wartości współrzędnych tych par. Poszczególne segmenty są łączone ze sobą w następujący sposób C=[level#1 x1 x2 x3 ...level#2 x2 x2 x3 ...; #pairs1 y1 y2 y3 ...#pairs2 y2 y2 y3 ...] Patrz też: mesh , gradient i quiver .

cos	cosinus cos(<i>X</i>) - kosinus elementów macierzy <i>X</i> .
cumprod	kumulowany iloczyn cumprod(<i>X</i>) - jeżeli <i>X</i> jest wektorem, funkcja zwraca wektor, którego <i>k</i> -ty element jest iloczynem <i>k</i> początkowych elementów wektora <i>X</i> . Jeżeli <i>X</i> jest macierzą, funkcja zwraca macierz, której kolumny zostały wyznaczone w opisany sposób.
cumsum	kumulowana suma cumsum(<i>X</i>) - jeżeli <i>X</i> jest wektorem, funkcja zwraca wektor, którego <i>k</i> -ty element jest sumą <i>k</i> początkowych elementów wektora <i>X</i> . Jeżeli <i>X</i> jest macierzą, funkcja zwraca macierz, której kolumny zostały wyznaczone w opisany sposób.
dc2sc	konwertuj współrzędne danych na współrzędne ekranu dc2sc - konwertuje współrzędne danych (min-maks) na współrzędne ekranu (0- 1). Instrukcja [<i>Xs</i>,<i>Ys</i>]=dc2sc(<i>Xd</i>,<i>Yd</i>) służy do konwersji wektorów <i>Xd</i> i <i>Yd</i> z aktualnego układu współrzędnych danych na odpowiadające współrzędne ekranu <i>Xs</i> , <i>Ys</i> . We współrzędnych ekranu punkt (0,0) leży w lewym dolnym rogu ekranu, a punkt (1,1) - w prawym górnym rogu. W układzie współrzędnych danych punkt (<i>Xmin</i> , <i>Ymin</i>) jest lewym dolnym narożnikiem prostokąta wykresu, a punkt (<i>Xmax</i> , <i>Ymax</i>) jest prawym górnym narożnikiem tego prostokąta. Wektor współrzędnych [<i>Xmin</i> <i>Xmax</i> <i>Ymin</i> <i>Ymax</i>] określa aktualne wartości skali na osiach układu współrzędnych w taki sposób, jak są one zwracane przez funkcję axis . Patrz też: sc2dc .
delete	usuń delete nazwa_pliku - usuwa plik o podanej nazwie z dysku.
det	wyznacznik det(<i>X</i>) - wyznacznik macierzy kwadratowej <i>X</i> .
diag	macierz (około)diagonalna diag(<i>V</i>,<i>K</i>) jest instrukcją tworzenia macierzy o obsadzonej <i>K</i> -tej diagonali; diag(<i>V</i>,<i>K</i>) - w przypadku, gdy <i>V</i> jest wektorem, zwraca macierz kwadratową rzędu <i>N</i>+abs(<i>K</i>) , w której elementy wektora <i>V</i> umieszczono na <i>K</i> -tej przekątnej; <i>N</i> oznacza tu długość wektora <i>V</i> , <i>K</i>=0 – główną przekątną, <i>K</i>> 0 - przekątną leżącą powyżej, a <i>K</i>< 0 - przekątną leżącą poniżej diagonali; diag(<i>V</i>) jest równoważne diag(<i>V</i>,0) . Na przykład diag(-<i>M</i>:<i>M</i>) + diag(ones(2*<i>M</i>,1),1) + diag(ones(2*<i>M</i>,1),-1) daje w wyniku macierz trójdziagonalną rzędu 2*<i>M</i>+1 . Jeżeli <i>V</i> jest macierzą, wówczas diag(<i>V</i>, <i>K</i>) zwraca wektor kolumnowy utworzony z elementów <i>K</i> -tej diagonali macierzy <i>V</i> . Polecenie diag(<i>V</i>) zwraca główną przekątną macierzy <i>V</i> .
diary	raport (generuj/zawieś/włącz) diary nazwa_pliku - generowanie raportu. Od momentu wykonania tej instrukcji kopie wszystkich poleceń z klawiatury i wyprowadzanych na ekran wyników będą dodatkowo zapisywane w pliku o podanej nazwie; diary off zawiesza raport, diary on ponownie włącza raport.
dir	wyświetl zawartość katalogu
disp	wyświetl na ekranie disp(<i>X</i>) wyświetla na ekranie kolejne elementy macierzy <i>X</i> (nie podając jej nazwy). Macierz <i>X</i> może zawierać tekst.
echo	włącz/wyłącz wyświetlanie na ekranie echo on Włącza wyświetlanie na ekranie wykonywanych poleceń z M-plików typu skryptowego. echo off Wyłącza wyświetlanie tych poleceń, echo mplik on włącza wyświetlanie wykonywanych poleceń z pliku typu funkcyjnego o podanej nazwie, echo mplik off wyłącza wyświetlanie wykonywanych poleceń z pliku o podanej nazwie, echo mplik przełącza wyświetlanie, echo on all włącza wyświetlanie wykonywanych poleceń z wszystkich plików typu funkcyjnego, echo off all wyłącza wyświetlanie wykonywanych poleceń z plików typu funkcyjnego. Patrz także: function .
eig	wartości i wektory własne Funkcja eig(<i>X</i>) zwraca wektor zawierający wartości własne kwadratowej macierzy <i>X</i> . Wyrażenie [<i>V</i>,<i>D</i>]=eig(<i>X</i>) zwraca macierz diagonalną <i>D</i> utworzoną z wartości własnych i macierz <i>V</i> , której kolumny są, odpowiadającymi kolejnym wartościom własnym, wektorami własnymi, tak że <i>X</i>*<i>V</i>=<i>V</i>*<i>D</i> .
else	Patrz if
elseif	Patrz if

end	koniec zakresu bloku polecen instrukcji for , while oraz if . Instrukcja end jest kojarzona z ostatnią niezakończoną instrukcją for , while lub if
eps	względna dokładność obliczeń Zmienna, której początkowa wartość równa jest różnicy między 1.0 a następną reprezentowalną w arytmetyce komputera liczbą zmiennoprzecinkową. Liczba eps może być używana do określenia dokładności w obliczeniach iteracyjnych.
error	komunikat o błędzie error('komunikat') - wyświetla tekst komunikat i powoduje zatrzymanie programu z sygnalizacją błędu.
eval	wyznacz eval(X) - interpretuje i wykonuje tekst zawarty w zmiennej X jako instrukcję lub wyrażenie języka MATLAB. Przykłady: (1) <pre>t='1/(i+j-1)'; for i=1:n for j=1:n a(i,j)=eval(t); end end</pre> generuje macierz Hilberta (2) <pre>S=[' x=3' ' y=4' 'z=sqrt(x*x+y*y)']; for k=1:3 eval(S(k,:)); end</pre> Napisy tworzące wiersze jednej macierzy muszą mieć tę samą długość.
exist	czy istnieje? exist('A') zwraca wartość: 1 - gdy w pamięci istnieje zmienna A , 2 - gdy na dysku istnieje plik A lub A.m , 0 - w pozostałych przypadkach.
exit	zakończ sesję z MATLAB-em Podobny efekt ma Ctrl-Z lub quit .
exp	funkcja wykładnicza exp(X) - funkcja wykładnicza elementów macierzy X .
expm	macierzowa funkcja wykładnicza Funkcja expm jest obliczana za pomocą aproksymacji Pade. Można uważać, że funkcja expm jest obliczana w taki sposób, że jeżeli $[V, D] = \text{eig}(X)$, to $\text{expm}(X) = V * \text{exp}(D) / V$. Jeżeli macierz X ma niedodatnie wartości własne, to wynik będzie zespolony.
eye	macierz jednostkowa eye(N) generuje macierzą jednostkową $N \times N$, zaś eye(M,N) lub eye([M,N]) tworzy macierz $M \times N$ z jedynkami na diagonalu i zerami poza. Ponadto eye(size(A)) ma ten sam rozmiar, co A .
feval	wywołanie funkcji pośrednie feval(F, x1, ..., xn) wywołuje funkcję o nazwie zapisanej w zmiennej tekstowej F z parametrami x1, ..., xn Na przykład <pre>F='foo' feval(F,9.64)</pre> jest równoważne wywołaniu foo(9.64) .
fft	dyskretna transformata Fouriera fft(X) jest dyskretną transformatą Fouriera wektora X . Jeśli długość wektora X jest potęgą dwójki, użyty zostanie szybki algorytm FFT (<u>radix-2</u>). W przeciwnym przypadku użyty zostanie algorytm wolniejszy. fft(X,N) jest N -punktową transformatą wektora X , uzupełnionego zerami do długości N lub obciętego do N elementów (jeżeli ma długość większą niż N). Jeśli X jest macierzą, wówczas operacja fft jest wykonywana dla każdej kolumny tej macierzy.

filter **filtr cyfrowy**

Instrukcja **Y=filter(B,A,X)** filtruje dane zawarte w wektorze **X** za pomocą filtru cyfrowego opisanego wektorami **A** i **B**, zgodnie z równaniem różnicowym

$$Y(n)=B(1)*X(n) + B(2)*X(n-1) + \dots + B(nb+1)*X(n-nb)-A(2)*Y(n-1)- \dots -A(na+1)*Y(n-na),$$

gdzie $na=length(A)-1$, $nb=length(B)-1$.

Wywołanie z dodatkowymi argumentami **[Y,Zf]=filter(B,A,X,Zi)** pozwala określić warunki początkowe **Zi** i otrzymać warunki końcowe **Zf** na elementach opóźniających.

find **znajdź**

find(X) zwraca wektor zawierający indeksy kolejnych niezerowych elementów wektora **X**.

I=find(X>100) zwraca indeksy elementów **X**, które są większe od 100.

finite **skończone?**

finite(X) zwraca macierz, której elementy są równe: 1 - gdy odpowiadające elementy macierzy **X** są skończone, 0 - w przeciwnym przypadku. Patrz także: **isnan**.

fix **część całkowita**

fix(X) zaokrągla elementy macierzy **X** do liczb całkowitych (w kierunku zera)

floor **podłoga**

floor(X) zaokrągla elementy macierzy **X** do liczb całkowitych (w kierunku minus nieskończoności).

flops **licznik operacji zmiennoprzecinkowych.**

flops(0) zeruje licznik. Licznik jest zwiększany o: 1 - przy operacjach dodawania, odejmowania, mnożenia lub dzielenia liczb rzeczywistych oraz wyznaczania funkcji elementarnych o argumentach i wynikach rzeczywistych, 2 - przy operacjach dodawania i odejmowania liczb zespolonych, 6 - przy operacjach mnożenia i dzielenia liczb zespolonych.

Funkcje elementarne w dziedzinie zespolonej zwiększają licznik zależnie od stopnia swojej złożoności.

for **Instrukcja pętli programowej**

Składnia instrukcji jest następująca: **for zm=expr, instr_1; ...; instr_n; , end**

gdzie: **zm** jest zmienną, **expr** jest wyrażeniem, dającym w wyniku (w ogólnym przypadku) macierz, **instr_k** jest instrukcją języka MATLAB. Na początku pod zmienną **zm** podstawiana jest pierwsza kolumna macierzy otrzymanej po obliczeniu wyrażenia **expr**, po czym wykonywane są instrukcje **instr_1; ...; instr_n;** z wnętrza bloku pętli **for**. Następnie pod zmienną **zm** podstawiana jest kolejna kolumna macierzy **expr** i wykonywane są instrukcje z wnętrza pętli. Postępowanie takie kontynuowane jest aż do wyczerpania wszystkich kolumn macierzy otrzymanej z wyrażenia **expr**. Wyrażenie to najczęściej jest wektorem wierszowym o postaci **m:n** i wtedy jego kolumny są po prostu skalarami. Przykład:

for i=1:n, for j=1:n, a(i,j)=1/(i+j-1); end, end

format **format wyprowadzania danych**

W MATLABie obliczenia są zawsze wykonywane na liczbach zmiennoprzecinkowych o pełnej precyzji, natomiast ich wyniki można wyprowadzać w następujących formatach:

	<i>domyślny</i> ; to samo co short ,
short	<i>stałoprzecinkowy krótki</i> (o 5 cyfrach znaczących),
long	<i>stałoprzecinkowy długi</i> o (15 cyfrach znaczących),
short e	<i>zmiennoprzecinkowy krótki w notacji wykładniczej</i> (5 cyfr znaczących)
long e	<i>zmiennoprzecinkowy długi w notacji wykładniczej</i> (15 cyfr znaczących)
hex	szesnastkowy
+	"testowy" - zamiast liczb wyprowadzane są symbole "+", "-" i spacja zależnie od tego, czy wynik jest dodatni, ujemny czy równy zero; części urojone liczb zespolonych nie są wyprowadzane,
bank	"finansowy" (stałoprzecinkowy dla obliczeń finansowych),
compact	"zwarty" - nie są wyprowadzane dodatkowe puste linie, oddzielające kolejne wyniki,
loose	"przejrzysty" - wyprowadzane są dodatkowe puste linie, oddzielające kolejne wyniki.

fprintf **wyprowadź dane do pliku**

Działanie funkcji **fprintf('plik','format',x)** jest równoważne z działaniem funkcji

sprintf('format',x), z tą różnicą, że wyniki są kierowane (dopisywane) do pliku dyskowego **plik**

function **funkcja**

Funkcje języka MATLAB dzielą się na wbudowane i zewnętrzne (m-pliki lub *rnex-pliki*). Nowe funkcje mogą być dołączane do MATLAB-a w postaci programu wykorzystującego istniejące funkcje i zapisanego w postaci pliku tekstowego, o nazwie identycznej z nazwą funkcji i rozszerzeniu **.m**. Na początku pliku musi

znajdować się linia **function**, definiująca składnię nowej funkcji. Na przykład, po utworzeniu na dysku pliku tekstowego o nazwie **stat.m** i o zawartości

```
function [mean,stdev]=stat(x)
n=prod(size(x));
mean=sum(x)./n;
stdev=sqrt(sum(x.^2)/n-mean^2);
```

zostanie zdefiniowana nowa funkcja **stat**, za pomocą której oblicza się średnią i odchylenie standardowe ciągu danych zawartych w wektorze. Funkcja ta wymaga podania jednego argumentu **x** i zwraca dwie wartości: **mean** i **stdev**. Zmienne użyte wewnątrz funkcji są zmiennymi lokalnymi. Z zewnątrz dostępne są jedynie parametry wejściowe, podawane przy wywołaniu funkcji, i parametry wyjściowe, zwracane po wykonaniu funkcji. Przykładowe wywołanie funkcji **stat**:

```
[m,s]=stat(1:2:100); sprintf('Średnia: %g, odchylenie standardowe: %g ', m, s);
```

Patrz: **script** - dla procedur, które mają pracować globalnie na wszystkich zmiennych w pamięci; patrz też: **echo**.

getenv **pobierz zmienną środowiskową**

getenv('envvar') Funkcja przeszukuje zbiór zmiennych środowiskowych systemu operacyjnego, poszukując zmiennej o nazwie **envvar**, a następnie zwraca przypisany tej zmiennej ciąg znaków (lub macierz pustą, jeżeli nie ma zmiennej środowiskowej o podanej nazwie).

ginput **pobierz współrzędne**

kursora myszy lub celownika krzyżowego na ekranie graficznym. Wywołanie **[x,y]=ginput(n)** spowoduje wyświetlenie na ekranie graficznym celownika krzyżowego w postaci dwóch prostopadłych linii, następnie odczeka, aż użytkownik **n**-krotnie wciśnie przycisk myszy (lub dowolny klawisz na klawiaturze) i zwróci **n**-elementowe wektory **x** i **y**, zawierające współrzędne celownika w chwilach naciśnięcia przycisku myszy lub klawisza klawiatury. Celownik krzyżowy można przesuwac za pomocą myszy lub klawiszy kursora z klawiatury. Za pomocą instrukcji **[x,y]=ginput** zwracany jest wektor współrzędnych o długości równej liczbie wybranych przez użytkownika punktów do momentu naciśnięcia klawisza ENTER.

[x,y,key]=ginput(n) zwraca dodatkowo kod wciśniętego klawisza (1, 2 lub 3 w przypadku wciśnięcia przycisków myszy, licząc od lewej, lub kody ASCII w przypadku wciśnięcia klawisza z klawiatury). Użycie drugiego, opcjonalnego argumentu wejściowego **[...]=ginput(n,'sc')** spowoduje, że otrzymane wartości współrzędnych będą wyrażone we współrzędnych ekranu (gdzie punkt (0,0) odpowiada lewemu dolnemu, a punkt (1,1) - prawemu górnemu rogowi ekranu), a nie we współrzędnych danych.

global **definicja zmiennych globalnych.**

Instrukcja **global x,y,z** spowoduje, że zmienne **x**, **y** i **z** będą miały zasięg globalny.

grid **wyświetl pomocniczą siatkę na aktualnym wykresie**

hess **forma Hessenberga macierzy**

Forma Hessenberga macierzy ma zera poniżej pierwszej podprzekątnej. Jeżeli macierz jest symetryczna lub hermitowska, to forma Hessenberga tej macierzy jest trójdzielna. Wywołanie **[P,H]=hess(A)** zwraca macierz unitarną **P** i macierz Hessenberga **H** spełniające warunki **A=P*H*P'** i **P'*P=eye(P)**. Samo **hess(A)** zwraca macierz **H**.

hold **zatrzymaj aktualny wykres na ekranie**

hold on włącza tryb zatrzymywania rysunków, **hold off** powraca do normalnego trybu, gdzie następny rysunek wymazuje poprzedni. Samo **hold** przełącza tryb zatrzymywania rysunku. Następne wywołania funkcji **plot** będą uzupełniały (o ile są wykonywane w stanie **on** przełącznika **hold**) aktualny rysunek o kolejne wykresy i wykorzystywały istniejące zakresy wartości na osiach współrzędnych.

home **umieść kursor w lewym górnym rogu ekranu**

(bez wymazywania zawartości ekranu). Patrz także **clc**.

if **Instrukcja warunkowa.**

Najprostsza składnia instrukcji warunkowej jest następująca: **if cond, instr, ... ; end**, gdzie **cond** jest warunkiem - może to być zmienną lub wyrażeniem, najczęściej jest to relacja. Ciąg instrukcji **instr, ...;** występujący po warunku **cond** będzie wykonany tylko wtedy, gdy macierz otrzymana w wyniku wyznaczenia wartości warunku **cond** ma wszystkie wartości różne od zera. Szczególnie jeśli **cond** jest skalar, to ciąg instrukcji **instr, ...;** zostanie wykonany wtedy, gdy **cond** będzie różne od zera.

Przykład: za pomocą instrukcji **if a < 0, a=-a; end** wyznaczany jest moduł liczby rzeczywistej **a**.

Możliwe są następujące warianty instrukcji warunkowej:

```
if cond, instr, ... ; else instr, ... ; end
```

oraz

```
if cond1, instr, ... ; elseif cond2, instr ... ; else instr, ... ; end
```

Przykład:

```
if i == j, A(i,j)=1; elseif abs(i-j) == 1, A(i,j)=-1; A(j,i)=-1;
elseif i > j+2, A(i,j)=2; else A(i,j)=0; end
```

imag **część urojona**

imag(X) - część urojona **X**. W języku MATLAB liczbę urojoną można wprowadzać w następujące sposoby: – jako pierwiastek z liczby -1.0 (np. **k=sqrt(-1)**), – przy użyciu predefiniowanych zmiennych **i** lub **j** (np. **z=1+2*i** lub równoważnie **z=1+2*j**), w takim przypadku nie wolno używać **i** lub **j** jako nazw zmiennych.

inf **(plus) nieskończoność**

inf reprezentuje "wartość" +nieskończoność według normy IEEE; jest np. wynikiem działania **1/0**.

input **wprowadź (liczbę lub ciąg znaków z klawiatury)**

Instrukcja **n=input('Ile jablek?')** spowoduje wyświetlenie na ekranie zawartego w wywołaniu pytania i będzie czekać na wprowadzenie przez użytkownika odpowiedniej liczby, a następnie jako wynik zwróci wprowadzoną liczbę. Instrukcja **input('Ile jablek?','s')** zwróci jako wynik ciąg znaków wprowadzony przez użytkownika.

inquire **"jak się masz MATLABie"?**

inquire zwraca wartości opisujące wewnętrzny stan MATLAB-a: **inquire('echo')** zwraca 1 - gdy **echo** jest włączone, 0 - gdy jest wyłączony; **inquire('casesen')** wyprowadza 1 - gdy włączono rozróżnianie dużych i małych liter (**casesen on**), 0 - w przeciwnym przypadku; **inquire('hold')** wypisuje 1 - gdy **hold** jest włączony, 0 - gdy jest wyłączony; instrukcja **[x,u]=inquire('axis')** zwraca **x=1** - gdy osie są „zamrożone” lub **x=0** przy automatycznym skalowaniu osi, a drugi zwracany parametr jest wektorem opisującym skalę na osiach: **v=[Xmin Xmax Ymin Ymax]**. Wywołanie **inquire('subplot')** zwraca 3-cyfrową liczbę opisującą sposób podziału ekranu graficznego (por. **subplot**); **[f,d]=inquire('format')** zwraca ciągi znaków **f** i **d** określające format wyświetlania danych. Użycie instrukcji: **eval(f)**, **eval(d)** spowoduje odtworzenie aktualnego formatu; wreszcie **[s,fname]=inquire('diary')** zwraca: **s=1**, jeżeli raport jest wyłączony i **s=0**, jeżeli raport jest włączony. Nazwa aktualnego pliku, w którym zapisywany jest raport, zwracana jest w zmiennej tekstowej **fname**. UWAGA. W następnych wersjach MATLABa funkcja **inquire** może nie występować (będzie zastępowana przez **get**).

inv **odwróć (macierz)**

inv(X) oblicza odwrotność macierzy kwadratowej **X**. Jeśli macierz jest źle uwarunkowana lub prawie osobliwa funkcja wyświetla ostrzeżenie.

isnan **nie liczba?**

isnan(x) jest funkcją zwracającą wartość 1, gdy jej argument **x** jest NaN, według normy IEEE (NaN - Not a Number), lub 0, gdy argument nie jest NaN (jest liczbą lub łańcuchem znaków).

isstr **czy łańcuch?**

isstr(s) to funkcja zwracająca wartość 1, gdy jej argument **s** jest ciągiem znaków.

i **jednostka urojona**

W języku MATLAB liczby urojone mogą być wprowadzane za pomocą predefiniowanych zmiennych **i** lub **j**, np. **2+3j** można wprowadzić w dowolny z następujących sposobów: **2+3*i**, **2+3*j**, **2+3*sqrt(-1)**.

keyboard **"rządź klawiaturą"**

Przekazania sterowania klawiaturze dokonujemy instrukcją **keyboard**. Instrukcja ta umieszczona w **m-pliku** powoduje zatrzymanie wykonania procedury lub skryptu i przekazanie sterowania klawiaturze (użytkownikowi). Możliwe jest wówczas wykonywanie dowolnych instrukcji języka MATLAB (sprawdzenie wartości zmiennych, ich modyfikacja, wywoływanie innych funkcji). Sposób pracy jest identyczny z trybem konwersacyjnym MATLABa, a specjalny stan pracy jest sygnalizowany znakiem zachęty **K>**. Pracę w trybie klawiatury można zakończyć, wciskając znak sterujący interpretowany jako znacznik końca pliku (CTRL-Z w systemie DOS, CTRL-D) w systemach UNIX. Można też wpisać **return**. Sterowanie zostanie przekazane do **m-pliku**, z którego wywołano instrukcję **keyboard**.

linspace **twórz wektor wartości rozmieszczonych liniowo**

linspace(x1,x2) tworzy wierszowy wektor 100 wartości rozłożonych w jednakowych odstępach (liniowo) pomiędzy **x1** i **x2**; **linspace(x1,x2,N)** rozmieszcza tak **N** wartości. Zobacz też: **logspace**, **:**.

load **ładuj**

load ładuje zmienne zapisane w binarnym pliku dyskowym. Instrukcja **load fname** służy do załadowania zmiennych zapisanych uprzednio w pliku **fname.mat** za pomocą instrukcji **save**. Instrukcja **load** bez argumentu ładuje zmienne z pliku **matlab.mat**. Możliwe jest również ładowanie zmiennych z pliku

tekstowego - w takim przypadku dane z pliku przypisywane są zmiennej o nazwie identycznej z nazwą pliku dyskowego. Patrz też **save**.

log **logarytm naturalny**

log(X) to logarytm naturalny elementów macierzy **X**. Jeśli w **X** występują wartości ujemne, to wynik będzie zespolony.

loglog **wykreśl w skali log-log**

loglog(...) działa tak samo jak **plot(...)**, z wyjątkiem, że na obu osiach są stosowane skale logarytmiczne. Zobacz też: **plot**, **semilogx**, **semilogy**.

logspace **twórz wektor wartości rozmieszczonych logarytmicznie**

logspace(d1,d2) tworzy wektor wierszowy o 50 punktach rozmieszczonych w logarytmicznie jednokowych odstępach pomiędzy 10^{d1} i 10^{d2} . Jeżeli **d2** wynosi **pi**, wtedy punkty rozmieszczane są między 10^{d1} i **pi**; **logspace(d1,d2,N)** generuje **N** punktów. Zobacz też: **linspace**, **:**.

lu **rozkład LU macierzy**

lu(X) daje w wyniku rozkład LU macierzy (czynniki eliminacja Gaussa). Wywołanie **[L,U]=lu(X)** zwróci górną macierz trójkątną **U**, „psychologicznie dolną” macierz trójkątną **L** (tj. iloczyn dolnej macierzy trójkątnej i macierzy permutacji). Macierze te spełniają związek: **X=L*U**. Wywołanie **[L,U,P]=lu(X)** spowoduje zwrot górnej macierzy trójkątnej **U**, dolnej macierzy trójkątnej **L** i macierzy permutacji **P**, spełniających warunek **P*X=L*U**. Samo **lu(X)** zwraca wynik działania procedury ZGEFA z biblioteki LINPACK.

max **największy?**

max(X) oblicza wartość maksymalną dla **X**. Jeśli **X** jest wektorem, **max(X)** zwraca największy element tego wektora. Jeśli **X** jest macierzą, to zwracany jest wektor wierszowy, którego *i*-ty element jest równy największemu elementowi *i*-tej kolumny macierzy **X**. Wywołanie **[y,idx]=max(x)** umożliwia dodatkowo zwrot wektora **idx**, zawierającego indeksy wierszy maksymalnych elementów w poszczególnych kolumnach. Wywołanie **max(X,Y)** zwraca macierz o takich samych wymiarach jak **X** i **Y**, której elementy są równe większym z elementów **X** i **Y** na odpowiadającej pozycji. W przypadku argumentów zespolonych używany jest moduł: **max(abs(X))**.

mesh **wykres trójwymiarowy**

Wywołanie **mesh(Z)** spowoduje wyświetlenie trójwymiarowego wykresu macierzy **Z** - wartości **Z** będą określały wysokość wykresu nad płaszczyzną podstawy. Wywołanie **mesh(Z,M)** określa dodatkowo kąty obserwacji. Dwielementowy wektor **M=[az,el]** winien zawierać: azymut **az** oraz elewację **el** (obie wyrażone w stopniach). Azymut obraca się względem osi **z**, przy czym wartości dodatnie oznaczają obrót punktu widzenia w kierunku przeciwnym do ruchu wskazówek zegara (lub obrót rysunku w kierunku zgodnym z ruchem wskazówek zegara). Dodatnie wartości elewacji **el** oznaczają przesunięcie nad rysunek, wartości ujemne - pod rysunek. Wartością standardową jest **M=[37.5 30]**. Wywołania postaci: **mesh(Z,S)** i **mesh(Z,M,S)** pozwalają na dobór współczynników skali na osiach **x**, **y** i **z**. Trójelementowy wektor **S=[sx,sy,sz]** ustala rozmiar obiektu w każdym z trzech wymiarów zgodnie z wzajemnym stosunkiem między wartościami skalarów **sx**, **sy** i **sz**. Patrz także: **meshdom** i **contour**.

meta **składaj (wykres)**

Instrukcja **meta fname** otwiera plik **fname.met** i wpisuje do niego aktualny wykres. Kolejne użycie instrukcji **meta** spowoduje dopisanie do uprzednio określonego pliku kolejnych wykresów. Jeżeli przy pierwszym użyciu instrukcji nie podano nazwy pliku, przyjmowana jest nazwa standardowa **matlab.met**. Plik zawierający wykresy może być przeglądany lub wyświetlany przy użyciu post-procesora graficznego gpp.

min **najmniejszy?**

Wartość minimalna. Patrz opis funkcji **max**

NaN **nie_liczba**

Zgodna z normą IEEE reprezentacja wyniku operacji arytmetycznej nieokreślonej, takiej jak np. **0/0** (nie będącej liczbą: **Not a Number**).

nargin **liczba argumentów wejściowych (larwe)**

Predefiniowana stała dostępna wewnątrz ciała funkcji użytkownika oznaczająca liczbę parametrów wejściowych (tj. umieszczonych w nawiasach okrągłych za nazwą funkcji) użytych przy wywołaniu funkcji.

nargout **liczba argumentów wyjściowych (larwy)**

Predefiniowana stała dostępna wewnątrz ciała funkcji użytkownika oznaczająca liczbę parametrów wyjściowych (tj. po lewej stronie instrukcji przypisania-wywołania funkcji) użytych przy wywołaniu funkcji.

norm **norma (wektora lub macierzy)**

Dla macierzy:

norm(X) jest największą wartością osobliwą macierzy **X****norm(X,1)** jest 1-normą macierzy **X**, tj. największą sumą kolumnową $\max(\text{sum}(\text{abs}(\text{real}(\mathbf{X})) + \text{abs}(\text{imag}(\mathbf{X}))))$ **norm(X,2)** jest równoważne **norm(X)**,**norm(X,inf)** jest normą „nieskończoną” macierzy **X**, tj. największą sumą wierszową $\max(\text{sum}(\text{abs}(\text{real}(\mathbf{X}')) + \text{abs}(\text{imag}(\mathbf{X}'))))$,**norm(X,'fro')** jest f-normą macierzy **X**: $\sqrt{\text{sum}(\text{diag}(\mathbf{X}' * \mathbf{X}))}$,

Dla wektorów:

norm(V,p) = $\text{sum}(\text{abs}(\mathbf{V})^p)^{1/p}$,**norm(V)** = **norm(V,2)****norm(V,inf)** = $\max(\text{abs}(\mathbf{V}))$ **norm(V,-inf)** = $\min(\text{abs}(\mathbf{V}))$ **num2str** **zamień liczbę w łańcuch znakowy**

t = num2str(x) zamienia skalar **x** na łańcuch znakowy o około czterech cyfrach i wykładniku, jeżeli jest niezbędny. Może być dołączony opcjonalny parametr ustalający żadaną precyzję: **t = num2str(x,prec)** zamienia skalar **x** na łańcuch znakowy z reprezentujący zapis liczby z dokładnością określoną przez **prec**. Patrz też **str2num** i **int2str**.

ones **jedyński**

ones generuje macierz o wszystkich elementach równych 1. I tak **ones(N)** zwraca macierz o wymiarze **N*N**, **ones(N,M)** zwraca macierz o **N** wierszach i **M** kolumnach, a **ones(X)**, gdzie **X** jest macierzą, zwraca macierz o takim samym wymiarze jak **X**.

pack **"upakuj"**

czyli „odświeżyć” pamięć. Podczas dłuższych sesji z zastosowaniem MATLABa pamięć robocza programu może ulec fragmentacji (może być wiele „dziur” po usuniętych zmiennych), co powoduje problemy z tworzeniem nowych zmiennych. Instrukcja **pack** umożliwia zapis wszystkich zmiennych z obszaru roboczego na dysk, wymazanie całej pamięci, a następnie z powrotem załadowanie zmiennych z dysku do pamięci operacyjnej.

pause **pauzuj**

pause wstrzymuje wykonanie programu do czasu naciśnięcia klawisza przez użytkownika, a **pause(n)** wstrzymuje wykonanie programu na **n** sekund.

pi **liczba π (3,1415926535897...)****plot** **wykreśl**

plot służy do tworzenia wykresów. Aby utworzyć wykres, którego kolejne punkty mają współrzędne określone przez wektory **X** i **Y**, należy użyć instrukcji **plot(X,Y)**. Jeśli jeden z argumentów **X** lub **Y** jest macierzą, to wówczas zostanie wyświetlonych wiele wykresów: ten z argumentów, który jest wektorem, będzie kojarzony z kolejnymi wierszami lub kolumnami (zależnie od tego, który wymiar pasuje) argumentu będącego macierzą. Innym sposobem wyświetlania kilku wykresów przy jednym użyciu funkcji **plot** jest użycie kilku par argumentów: **plot(X1,Y1,X2,Y2,...)**. Do każdej pary argumentów **X Y** można dodać opcjonalny trzeci argument - ciąg znaków określający rodzaj i kolor linii wykresu i markerów.

Rodzaje linii: - ciągła, -- przerywana, kropkowa, -. linia typu „kreska - kropka”.

Rodzaje markerów (sposobów oznaczania punktów wykresu określonych przez wektory **X** i **Y**):

. (kropka), + znak plus, * gwiazdka, x krzyżyk (litera 'x'), o kółko (litera 'o').

Kolory: r czerwony (red), g zielony (green), b niebieski (blue), w biały (white), i kolor tła (invisible), c1...c15 kolor o podanym numerze (z palety 16 kolorów).

Ponadto dopuszczalne są następujące warianty instrukcji **plot**:

plot(Y) - wykres wektora **Y** według indeksów jego elementów,

plot(Y) - gdzie **Y** oznacza wektor lub macierz zespoloną, jest równoważne instrukcji

plot(real(Y),imag(Y)) (w przypadku użycia argumentów zespolonych w innych postaciach instrukcji **plot**, część urojona zostanie zignorowana)

Przykłady:

plot(x,y,'w--') - wykres będzie wyświetlony linią przerywaną w kolorze białym,

plot(x,y,x1,y1,'-',x2,y2,'--') - trzy wykresy, dla drugiego i trzeciego określono rodzaje linii.

Patrz także: **semi**, **loglog**, **polar**, **grid**, **shg**, **cle**, **clg**, **title**, **xlabel**, **ylabel**, **axis**, **hold**, **mesh**, **contour**, **subplot**.

polar **wykreśl w współrzędnych biegunowych**

polar(phi,r) wykreśla wykres we współrzędnych biegunowych **r**, **phi**. Kąt **phi** wyrażony jest w radianach. Używanie różnych rodzajów linii i kolorów - tak samo jak w instrukcji **plot**; siatka współrzędnych biegunowych może być narysowana za pomocą instrukcji **grid**.

polyline **łamana**

polyline to narzędzie do rysowania linii łamanej. Dopuszczalne są następujące sposoby wywołania funkcji: **polyline(X,Y)** umożliwia wykreślenie łamanej, której kolejne punkty mają współrzędne określone przez wektory **X** i **Y**,

polyline(X,Y,'typ') pozwala na określenie rodzaju i koloru linii (por. opis instrukcji **plot**),

polyline(X,Y,'sc') interpretuje dane zawarte w wektorach **X** i **Y** jako współrzędne ekranu, gdzie punkt (0,0) odpowiada lewemu dolnemu, a punkt (1,1) - prawemu górnemu rogowi ekranu (standardowo używane są współrzędne danych).

Patrz także: **polymark**, **text**, **sc2dc** i **dc2sc**.

polymark **rysuj znaczniki**

polymark to narzędzie do rysowania znaczników (markerów). Dopuszczalne są następujące sposoby wywołania funkcji:

polymark(X,Y) umożliwia wykreślenie markerów w punktach o współrzędnych określonych przez wektory **X** i **Y**,

polymark(X,Y,'yp') pozwala na określenie rodzaju i koloru markera (por. opis instrukcji **plot**),

polymark(X,Y,'se') interpretuje dane zawarte w wektorach **X** i **Y** jako współrzędne ekranu, gdzie punkt (0,0) odpowiada lewemu dolnemu, a punkt (1,1) - prawemu górnemu rogowi ekranu (standardowo używane są współrzędne danych).

Patrz także: **polyline**, **text**, **sc2dc** i **dc2sc**.

prod **iloczyn elementów**

prod(X) oblicza iloczyn elementów wektora **X**. Jeżeli **X** jest macierzą, wówczas zwracany jest wektor, którego elementy są równe iloczynom kolejnych kolumn tej macierzy.

prtsc **drukuj ekran graficzny**

Wydruk zawartości ekranu graficznego na drukarce; **prtsc('ff')** spowoduje wysunięcie papieru do nowej strony.

qr **dekompozycja QR macierzy**

qr(X) dokonuje dekompozycji QR macierzy **X**. Możliwe są następujące sposoby wywołania funkcji:

[Q,R]=qr(X) zwraca górną macierz trójkątną **R** o tym samym wymiarze, co **X** oraz macierz unitarną **Q** spełniającą warunek: $\mathbf{X}=\mathbf{Q}*\mathbf{R}$; **[Q,R,E]=qr(X)** zwraca macierz permutacji **E**, górną macierz trójkątną **R** z malejącymi elementami na diagonalu i macierz unitarną **Q** takie, że: $\mathbf{X}*\mathbf{E}=\mathbf{Q}*\mathbf{R}$; samo **qr(X)** zwraca wynik działania procedury ZQRDC z biblioteki matematycznej LINPACK (**triu(qr(X))** jest równe **R**).

quit, exit **wyjdź**

quit, a także **exit** powoduje wyjście z programu MATLAB.

rand **generuj liczby losowe**

rand bez argumentów zwraca pojedynczą liczbę losową, **rand(n)** zwraca macierz liczb losowych o wymiarze $n \times n$, **rand(n,m)** zwraca macierz o wymiarze $n \times m$, **rand(X)** zwraca macierz o takim samym wymiarze jak argument **X**. Po uruchomieniu programu MATLAB funkcja **rand** generuje liczby losowe o rozkładzie równomiernym w przedziale (0,1). Wywołanie **rand('normal')** zmienia rozkład na normalny (gaussowski) o wartości średniej równej zero i wariancji równej jedności, a wywołanie **rand('uniform')** spowoduje powrót do rozkładu równomiernego; **rand('dist')** zwraca ciąg znaków opisujący aktualny rozkład ('uniform' lub 'normal'); **rand('seed')** zwraca wartość punktu startowego generatora; **rand('seed',n)** ustawia nowy punkt startowy generatora liczb losowych na wartość **n**; **rand('seed',0)** przywraca standardowy punkt startowy generatora liczb losowych.

real **część rzeczywista**

real(X) wylicza część rzeczywistą elementów macierzy **X**.

Patrz też: **imag** i **abs**.

rem **pozostałość**

rem(x,y) podaje "resztę" z dzielenia **x** przez **y**. Ta "reszta" **rem(x,y)** jest równa $\mathbf{x}-\mathbf{y}*n$, gdzie $\mathbf{n}=\text{fix}(\mathbf{x}./\mathbf{y})$ jest częścią całkowitą wartości $\mathbf{x}./\mathbf{y}$. Gdy **x** i **y** są dodatnie, **rem(x,y)** pokrywa się z matematyczną resztą.

return **powrót**

Powrót do funkcji wywołującej (lub klawiatury).

round **zaokrąglij**

round(X) zaokrągla elementy macierzy **X** do najbliższych im liczb całkowitych.

save **zachowaj**

Zapis zmiennych z pamięci operacyjnej w pliku dyskowym. Dopuszczalne są następujące sposoby użycia instrukcji **save**:

save zapisuje wszystkie zmienne w pliku **matlab.mat**,

save fname zapisuje wszystkie zmienne w pliku **fname.mat**,

save fname X zapisuje tylko zmienną **X** w pliku **fname.mat**,

save fname X Y Z zapisuje zmienne **X**, **Y**, **Z** w pliku **fname.mat**,

save fname X /ascii zapisuje zmienną **X** w postaci 8-cyfrowych liczb ASCII w pliku tekstowym o nazwie **fname**,

save fname X /ascii /double zapisuje zmienną **X** w postaci 16-cyfrowych liczb ASCII w pliku tekstowym o nazwie **fname**.

Zamiast znaku "/" można użyć minusa „-”.

sc2dc **konwertuj współrzędne ekranu na współrzędne danych**

Instrukcja postaci **[Xd,Yd]=sc2dc(Xe,Ye)** umożliwia przeliczenie wektorów **Xe** i **Ye** opisujących położenie punktów w układzie współrzędnych ekranu na współrzędne danych **Xd**, **Yd**. W układzie współrzędnych ekranu punkt (0,0) odpowiada lewemu dolnemu, a punkt (1,1) - prawemu górnemu rogowi ekranu. W układzie współrzędnych danych (**Xmin**, **Ymin**) jest lewym dolnym, a (**Xmax**, **Ymax**) - prawym górnym rogiem prostokąta przeznaczonego na wykres. Czteroelementowy wektor **[Xmin Xmax Ymin Ymax]** opisuje aktualny stan ustawienia skali na osiach współrzędnych, tak jak to zwraca funkcja **axis**.
Patrz także **dc2sc**.

semilogx **wykreśl z logarytmiczną osią x**

Utworzenie wykresu z logarytmiczną (o podstawie 10) skalą na osi **x** i liniową skalą na osi **y**. Składnia i parametry - patrz opis instrukcji **plot**.

semilogy **wykreśl z logarytmiczną osią y**

Utworzenie wykresu z logarytmiczną (o podstawie 10) skalą na osi **y** i liniową skalą na osi **x**. Składnia i parametry - patrz opis instrukcji **plot**.

shg **wyświetl okno graficzne****sign** **znak**

sign(X) jest funkcją znaku, która dla każdego elementu macierzy **X** funkcja zwraca: 1 - jeśli jest to element większy od zera, 0 - jeśli jest to element równy zero, -1 - jeśli jest to element ujemny. Jeśli **X** jest macierzą zespoloną, wówczas **sign(X)=X ./abs(X)**

sin **sinus**

sin(X) oblicza sinus elementów macierzy **X**.

size **wymiar**

size(X) zwraca wymiary macierzy **X**. Jeśli **X** jest macierzą o wymiarach **m x n**, to funkcja spowoduje zwrot dwuelementowego wektora **[m n]**. Można również użyć funkcji w instrukcji z podwójnym podstawieniem: **[m,n]=size(X)**.

sort **sortuj**

sort(X) sortuje kolumny macierzy **X** w kierunku rosnącym. Instrukcja postaci **[Y,I]=sort(X)** zwraca, obok macierzy **Y** będącej uporządkowaną macierzą **X**, macierz **I** zawierającą indeksy używane przy sortowaniu. Jeśli **X** jest wektorem, to zachodzi zależność **Y=X(I)**. Jeśli **X** jest macierzą zespoloną, wówczas jako klucza podczas sortowania używa się **abs(X)**.

sprintf **wyprowadź znakowo**

sprintf konwertuje wartość liczbową na ciąg znaków. Wyrażenie **T=sprintf('tekst %g',x)** powoduje utworzenie zmiennej tekstowej **T**, której częścią będzie ciąg znaków ASCII, reprezentujący wartość skalara **x**, zgodnie ze wzorcem konwersji **%g**. Więcej informacji można znaleźć w opisie funkcji **sprintf**, zdefiniowanej w bibliotece **stdio** języka C. Wersja funkcji **sprintf** zaimplementowana w języku MATLAB dopuszcza jeszcze dwa dodatkowe argumenty wejściowe. Przykład: jeśli wartość zmiennej **R** wynosi 4, wówczas w wyniku wykonania instrukcji

T=sprintf('Pole koła o promieniu %g wynosi %g',R,pi*R^2) otrzymamy zmienną tekstową **T** zawierającą napis: *Pole koła o promieniu 4 wynosi 50.2655*. Patrz także: **num2str**.

sqrt **pierwiastek kwadratowy**

sqrt(X) wyznacza pierwiastek kwadratowy elementów macierzy **X**. Jeśli którykolwiek z elementów **X** jest ujemny, to wynik będzie zespolony.

startup **start programu od wykonania poleceń**

zawartych w pliku *startup.m* (plik o nazwie *startup.m*, o ile istnieje, jest wykonywany w momencie uruchamiania programu MATLAB).

string **łańcuch znakowy**

(ciąg znaków). Instrukcja **T='Hello, world'** tworzy zmienną tekstową **T**, pamiętaną jako wektor wierszowy 12-elementowy, którego elementami są kody ASCII kolejnych znaków tekstu 'Hello, world'. Długość wektora jest równa liczbie znaków tekstu. Wyrażenie **Y=abs(T)** zwraca wektor, którego elementy są liczbami rzeczywistymi równymi kodom ASCII kolejnych znaków tekstu zapisanego w zmiennej **T**. Wyrażenie **T=setstr(Y)** zamienia wektor liczbowy **Y** na zmienną tekstową **T**. Jeśli elementem tekstu ma być znak cudzysłowu **'**, to należy użyć dwóch takich znaków. Dla sprawdzenia, czy zmienna **X** jest zmienną tekstową można użyć funkcji **isstr(X)**.

str2num **zamień łańcuch znakowy w liczbę**

x=str2num(s) zamienia łańcuch **s** reprezentujący znakowo (w kodzie ASCII) wartość numeryczną na MATLABową wartość numeryczną; **str2num** przekształca macierz łańcuchów znakowych na macierz numeryczną, np.

```
A = [ '1 2'
      '3 4' ];
str2num(A)    %=> [1 2; 3 4]
```

Patrz również **num2str**

subplot **podwykres**

subplot(mnp) określa podział ekranu graficznego. Ekran graficzny zostaje podzielony na **m×n** części i w każdej z tych części można umieścić osobny wykres. Argumentem **mnp** instrukcji jest liczba 3-cyfrowa, której pierwsza cyfra **m** określa sposób podziału w pionie (liczbę rzędów wykresów), druga cyfra **n** - sposób podziału w poziomie (liczbę kolumn wykresów), a trzecia cyfra **p** określa część, w której pojawi się następny wykres. Na przykład, instrukcja **subplot(223)** umożliwi podział ekranu w pionie i poziomie na dwie części (łącznie na cztery pola wykresów) i wybranie lewego dolnego pola do umieszczenia tam kolejnego wykresu. Użycie instrukcji **cdg** lub **subplot(111)** powoduje powrót do pracy z pojedynczym polem wykresu na ekranie graficznym.

sum **sumuj elementy**

sum(X) wylicza sumę elementów wektora **X**. Jeżeli **X** jest macierzą, wówczas zwracany jest wektor, którego elementy są równe sumom elementów kolejnych kolumn tej macierzy; **sum(diag(X))** jest śladem macierzy **X**.

svd **dekompozycja macierzy według wartości osobliwych (dmwwo)**

svd dekomponuje macierz według jej wartości osobliwych. Wyrażenie **[U,S,V]=svd(X)** daje w wyniku macierz diagonalną **S** o tym samym wymiarze co **X** i o nieujemnych elementach diagonalnych uporządkowanych w kierunku malejącym oraz macierze unitarne **U** i **V** takie, że **X=U * S * V'**. Samo **svd(X)** zwraca wektor zawierający wartości osobliwe macierzy **X**; **[U,S,V]=svd(X,0)** daje w wyniku rozkład „ekonomiczny”. Jeśli **X** jest macierzą o wymiarze **m×n** i **m > n**, wówczas jest obliczanych tylko pierwszych **n** kolumn macierzy **U**, a macierz **S** ma wymiar **n×n**.

tan **tangens**

tan(X) oblicza tangens elementów macierzy **X**.

text **tekst**

Instrukcja **text** służy do wyświetlenia napisu na wykresie: **text(x,y,'napis')** służy do wyświetlenia tekstu podanego jako trzeci argument na ekranie graficznym, począwszy od miejsca o współrzędnych **(x,y)** (zgodnie z aktualnymi wartościami współrzędnych danych). Jeśli **x** i **y** są wektorami, to podany tekst zostanie wypisany we wszystkich miejscach określonych przez kolejne pary współrzędnych wektorów **x** i **y**. Jeśli **'napis'** jest tablicą tekstową o takiej samej długości jak wektory **x** i **y**, to kolejne napisy - wiersze tablicy tekstowej - będą umieszczane w punktach wskazywanych przez odpowiadające im współrzędne wektorów **x** i **y**. Użycie **text(x,y,'napis','se')** interpretuje liczby zawarte w zmiennych **x** i **y** jako współrzędne ekranu, tj. przyjmując, że (0,0) jest lewym dolnym, a (1,1) prawym górnym rogiem ekranu.

title **tytuł**

title('tekst') wyświetla u góry wykresu, jako jego nagłówek, tekst będący argumentem instrukcji.

triu	górną trójkątą triu(X) jest górną trójkątną częścią X ; triu(X,K) jest utworzone przez elementy na i powyżej K -tej przekątnej macierzy X . K = 0 wyznacza główną przekątną, K > 0 lokuje się powyżej głównej przekątnej, a K < 0 poniżej.
type	wypisz zawartość pliku tekstowego type fname wyświetla zawartość pliku tekstowego fname . Jeśli podana nazwa pliku nie zawiera rozszerzenia, automatycznie przyjmowane jest rozszerzenie .m .
what	wypisz pliki MATLABa Wypisuje listę m -, mat - i mex -plików. Instrukcja what , bez dodatkowych parametrów wypisuje wspomniane pliki znajdujące się tylko w aktualnym katalogu; instrukcja what dirname wypisuje wspomniane pliki znajdujące się w katalogu dirname wymienionym w MATLABPATH, przy czym wystarczy podać ostatnie komponenty ścieżki do katalogu, o ile jednoznacznie go identyfikują. Na przykład what general i what matlab/general wypisują pliki z tego samego katalogu toolbox/matlab/general . Patrz też: dir , who , which , lookfor .
who	wypisz używane zmienne who wypisuje używane zmienne przestrzeni roboczej, whos wypisuje więcej informacji o tych zmiennych (np. wymiar), who global i whos global czyni to samo dla przestrzeni globalnej.
while	dopóty Instrukcja pętli o nieokreślonej z góry liczbie powtórzeń. Składnia while cond, instr1, ..., instrn; end , gdzie cond jest w ogólnym przypadku wyrażeniem dającym w wyniku macierz. Instrukcje instr1, ..., instrn ; wewnątrz instrukcji while będą wykonywane tak długo, jak długo macierz otrzymana w wyniku obliczenia wyrażenia cond będzie miała wszystkie elementy niezerowe. Zazwyczaj wyrażenie cond jest relacją skalarną. Przykład (zakładamy, że macierz kwadratowa A została wcześniej zdefiniowana) E=O.*A; F=E + eye(size(E)); N=1; while norm(E+F-E,1) > O, E=E+F; F=A*F/N; N=N+1; end Wewnątrz pętli powinna być instrukcja zmieniająca wartość wyrażenia cond , gdyż inaczej pętla nigdy się nie zakończy.
xlabel	opis osi x xlabel('tekst') umieszcza tekst podany jako argument na aktualnym wykresie pod osią poziomą.
ylabel	opis osi y ylabel('tekst') umieszcza tekst podany jako argument na aktualnym wykresie przy osi pionowej.
zeros	zera zeros generuje macierz złożoną z samych zer: zeros(N) zwraca macierz kwadratową o wymiarze N*N , zeros(N,M) zwraca macierz o N wierszach i M kolumnach, zeros(X) , gdzie X jest macierzą, zwraca macierz o takim samym wymiarze jak X .

D.2. BIBLIOTEKA PROCEDUR DO CYFROWEGO PRZETWARZANIA SYGNAŁÓW (SIGNAL PROCESSING TOOLBOX)

D.2.1. Funkcje okien

bartlett(N)	N -punktowe okno Bartlett'a,
blackman(N)	N -punktowe okno Blackmana,
boxcar(N)	N -punktowe okno prostokątne,
chebwin(N,R)	N -punktowe okno Czebyszewa, o R decybelowych zafalowaniach. N musi być nieparzyste. Jeżeli N jest parzyste, zwracane jest nieparzyste okno o długości N+1 ,
hamming(N)	N -punktowe okno Hamminga,
hanning(N)	N -punktowe okno Hanny,
kaiser(N,beta)	N -punktowe okno Kaisera o parametrze beta ,
triang(N)	N -punktowe okno trójkątne.

D.2.2. Transformaty, procedury filtracji, funkcje korelacji**conv** **splot i mnożenie wielomianów**

Wywołanie **C = conv(A,B)** splata wektory **A** i **B**. Wynikowy wektor ma długość **length(A)+length(B)-1**. Jeśli **A** i **B** są wektorami współczynników wielomianu, ich splatanie jest równoważne mnożeniu dwóch wielomianów. Patrz też **conv2** i **deconv**.

conv2 **splot dwuwymiarowy**

W wyniku wywołania **C=conv2(A, B)** otrzymamy 2-wymiarowy splot macierzy **A** i **B**. Jeżeli macierz **A** ma wymiar $m_a \times n_a$, a macierz **B** - wymiar $m_b \times n_b$, wówczas macierz **C** będzie miała wymiar $(m_a+m_b-1) \times (n_a+n_b-1)$.

corrcoef **współczynniki korelacji**

Wywołanie **corrcoef(X)** zwraca macierz współczynników korelacji utworzoną z macierzy **X**, w której wiersze odpowiadają kolejnym obserwacjom, a kolumny – zmiennym; **corrcoef(X,Y)** jest równoważne **corrcoef([X Y])**. Jeżeli **C** jest macierzą kowariancji, **C = cov(X)**, wówczas **corrcoef(X)** jest macierzą, której element (i,j) jest równy $C(i,j)/\sqrt{C(i,i)*C(j,j)}$.

cov **macierz kowariancji**

Wywołanie **cov(X)** gdzie **X** jest wektorem, zwraca wariancję tego wektora. Dla macierzy, gdzie wiersze odpowiadają kolejnym obserwacjom, a kolumny kolejnym zmiennym, **cov(X)** jest macierzą kowariancji, **diag(cov(X))** jest wektorem wariancji kolejnych kolumn, a **sqrt(diag(cov(X)))** jest wektorem odchyłeń standardowych. Nadto **cov(X, Y)** jest równoważne **cov([X Y])**. Patrz też **xcov**.

deconv **rozplot**

Wywołanie **[Q, R] = deconv(B, A)** „rozplata” wektor **A** z wektora **B**. Wynik jest zwracany w wektorze **Q**, a reszta - wektorze **R** takim, że **B = conv(Q,A)+R**. Jeśli **A** i **B** są wektorami współczynników wielomianu, rozplot jest równoważny dzieleniu wielomianów. Wynikiem dzielenia **B** przez **A** jest iloraz **Q** i reszta **R**. Patrz też **conv**.

fft **dyskretna transformata Fouriera**

Funkcja wbudowana, opisana w ramach prezentacji instrukcji MATLABa

fft2 **dwuwymiarowe FFT.**

fft2(X) jest dwuwymiarowym FFT macierzy **X**. Wiersze lub kolumny, których liczba elementów nie jest potęgą dwójki, są dopełniane zerami.

fftfilt **filtracja z zastosowaniem FFT (Overlap-add method).**

Y=fftfilt(B,X) - filtruje sygnał **X** za pomocą filtru SOI **B**.

Y= fftfilt(B, X, M) - do filtracji stosowana jest **M**-punktowa FFT.

fftshift **przesunięte FFT**

Jeżeli **X** jest wektorem, wówczas **fftshift(X)** zwraca wektor, którego lewa i prawa połowa zostały zamienione. Jeżeli **X** jest macierzą, **fftshift(X)** zamienia miejscami pierwszą i trzecią oraz drugą i czwartą ćwiartkę tej macierzy. **fftshift** jest użyteczne przy przetwarzaniu FFT do przesuwania zera na osi do środka układu współrzędnych.

hilbert **transformata Hilberta.**

Wywołanie **Y=hilbert(X)** zwraca wektor **Y**, którego część rzeczywista jest kopią oryginalnych danych, a część urojona - transformatą Hilberta części rzeczywistej wektora **X**.

ifft **odwrotna dyskretna transformata Fouriera**

ifft(X) jest odwrotną dyskretną transformatą Fouriera wektora **X**.

ifft(X,N) jest **N**-punktową **ifft**, gdzie **N** musi być potęgą dwójki.

ifft2 **odwrotne dwuwymiarowe przekształcenie Fouriera.**

ifft2(Y) jest odwrotnym dwuwymiarowym FFT macierzy **Y**. Wiersze lub kolumny, których liczba elementów nie jest potęgą dwójki, są uzupełniane zerami.

xcorr **estymata funkcji korelacji wzajemnej**

Wywołanie **R = xcorr(A,B)** gdzie **A** i **B** są wektorami o długości **M**, zwraca wartości funkcji korelacji wzajemnej w wektorze kolumnowym **R** o długości **2M-1**; **xcorr(A)**, gdzie **A** jest wektorem, zwraca funkcję autokorelacji; **xcorr(A)**, gdzie **A** jest macierzą $M \times N$, zwraca dużą macierz o **2M-1** wierszach, której **N** z kolumn zawiera wartości funkcji korelacji wzajemnej dla wszystkich kombinacji kolumn **A**. Wartość obli-

czonej funkcji korelacji dla zerowego argumentu odpowiada środkowemu elementowi wektora \mathbf{R} , tj. elementowi lub wierszowi wektora lub macierzy \mathbf{M} . Standardowo, **xcorr** wyznacza korelację bez stosowania normalizacji. **xcorr(A, 'biased')** lub **xcorr(A, B, 'biased')** zwraca „obciążoną” estymatę funkcji korelacji wzajemnej. Estymata obciążona jest estymatą nieznormalizowaną, przeskalowaną przez czynnik $1/M$. **xcorr(..., 'unbiased')** zwraca „nieobciążoną” estymatę funkcji korelacji wzajemnej. Estymata nieobciążona jest estymatą nieznormalizowaną, przeskalowaną przez czynnik $1/(M-\text{abs}(\mathbf{k}))$, gdzie \mathbf{k} jest indeksem do wyniku. **xcorr(..., 'coeff')** normalizuje wektor funkcji korelacji wzajemnej w taki sposób, że wartości funkcji korelacji odpowiadające zerowemu przesunięciu są równe 1.0.

xcorr2 dwuwymiarowa funkcja korelacji wzajemnej

Wywołanie **R=xcorr2(A,B)** zwraca funkcję korelacji wzajemnej macierzy \mathbf{A} i \mathbf{B} . **xcorr2(A)** jest funkcją autokorelacji.

xcov estymata funkcji kowariancji wzajemnej

Kowariancja wzajemna jest funkcją korelacji wzajemnej dwóch ciągów po odjęciu od nich wartości średnich. Wywołanie **R = xcov(A,B)** gdzie \mathbf{A} i \mathbf{B} są wektorami o długości M , zwraca wartości funkcji kowariancji wzajemnej w wektorze kolumnowym \mathbf{R} o długości $2M-1$; **xcov(A)**, gdzie \mathbf{A} jest wektorem, zwraca wartości funkcji autokowariancji; **xcov(A)**, gdzie \mathbf{A} jest macierzą $M \times N$, zwraca dużą macierz o $2M-1$ wierszach, której NZ kolumn zawiera kowariancje wzajemne dla wszystkich kombinacji kolumn \mathbf{A} . Wartość obliczonej funkcji kowariancji dla zerowego argumentu odpowiada środkowemu elementowi wektora \mathbf{R} , tj. elementowi lub wierszowi wektora lub macierzy \mathbf{M} . Standardowo, **xcov** wyznacza kowariancję bez stosowania normalizacji. Wywołanie **xcov(A, 'biased')** lub **xcov(A, B, 'biased')** zwraca „obciążoną” estymatę funkcji kowariancji wzajemnej. Estymata obciążona jest estymatą nieznormalizowaną, przeskalowaną przez czynnik $1/M$. Wywołanie: **xcov(..., 'unbiased')** zwraca „nieobciążoną” estymatę funkcji kowariancji wzajemnej. Estymata nieobciążona jest estymatą nieznormalizowaną, przeskalowaną przez czynnik $1/(M-\text{abs}(\mathbf{k}))$, gdzie \mathbf{k} jest indeksem do wyniku. Wywołanie: **xcov(..., 'coeff')** normalizuje wektor funkcji kowariancji wzajemnej w taki sposób, że wartości funkcji kowariancji odpowiadające zerowemu przesunięciu są równe 1.0. Patrz też **cov**.

D.2.3. Funkcje pomocnicze

angle kąt fazowy

Wywołanie **angle(H)** zwraca kąty fazowe (w radianach) elementów macierzy zespolonej \mathbf{H} . Patrz też **unwrap**.

bilinear transformacja biliniowa

(z opcjonalnym wstępnym odkształceniem częstotliwości). Wywołanie

[Zd, Pd, Kd]=bilinear(Z, P, K, Fs) przekształca transmitancję w dziedzinie zmiennej s , określoną przez wektory kolumnowe zer \mathbf{Z} , biegunów \mathbf{P} i stałą wzmocnienia \mathbf{K} , na jej odpowiednik w dziedzinie z , otrzymany w wyniku zastosowania transformacji biliniowej

$$H(z) = H(s) \Big|_{s=2Fs \cdot \frac{z-1}{z+1}}$$

gdzie \mathbf{Fs} jest częstotliwością próbkowania w Hz.

[NUMd, DENd]=bilinear(NUM, DEN, Fs) (gdzie \mathbf{NUM} i \mathbf{DEN} są wektorami wierszowymi zawierającymi współczynniki licznika i mianownika transmitancji $\mathbf{NUM}(s)/\mathbf{DEN}(s)$, w kolejności malejących potęg s , daje w wyniku transformatę $\mathbf{NUMd}(z)/\mathbf{DENd}(z)$ w dziedzinie z ;

[Ad, Bd, Cd, Dd]=bilinear(A, B, C, D, Fs) jest wersją transformacji w dziedzinie przestrzeni stanu. Każda z powyższych trzech postaci funkcji **bilinear** akceptuje dodatkowy opcjonalny parametr wejściowy, który określa wstępne „odkształcenie” częstotliwości. Na przykład, **[Zd, Pd, Kd]=bilinear(Z, P, K, Fs, Fp)** wykonuje odkształcenie przed transformacją biliniową tak, że charakterystyki amplitudowe przed i po przekształceniu są zgodne dokładnie dla wartości częstotliwości \mathbf{Fp} (punkt zgodności \mathbf{Fp} jest określony w Hz).

decimate "decymuj" próbk

czyli obniżyć częstotliwość próbkowania sygnału (po filtracji dolnoprzepustowej). Wywołanie

Y=decimate(X, R) próbkuje dane zawarte w wektorze \mathbf{X} z częstotliwością $1/R$ razy mniejszą niż początkowa częstotliwość próbkowania. Wynikowy wektor \mathbf{Y} jest \mathbf{R} razy krótszy, tzn. **length(Y)=length(X)/R**. Przed powtórny próbkowaniem **decimate** filtruje dane z wektora \mathbf{X} przez dolnoprzepustowy filtr Czebyszewa typu I, ósmego rzędu, o częstotliwości odcięcia $0,8 \cdot (\mathbf{Fs}/2)/R$; **Y=decimate(X, R, M)** wykorzystuje do tego celu filtr Czebyszewa rzędu \mathbf{M} , a **Y=decimate(X, R, 'FIR')** wykorzystuje do filtracji 30-punktowy filtr SOI, utworzony przez **fir1(30, 1/R)**, natomiast **Y=decimate(X, R, N, 'FIR')** wykorzystuje \mathbf{N} -punktowy filtr SOI. Patrz też **interp**.

detrend usunięcie liniowego trendu z wektora

(zazwyczaj przed FFT). Wywołanie **Y=detrend(X)** wyznacza trend danych na podstawie wektora **X** i usuwa go z tego wektora; wynikiem jest wektor **Y**. Jeśli **X** jest macierzą, wówczas **detrend** usuwa trend z każdej kolumny tej macierzy; **Y=detrend(X,0)** usuwa jedynie wartość średnią z wektora **X** lub, jeśli **X** jest macierzą, wartość średnią z każdej kolumny

freqs charakterystyka amplitudowo-fazowa filtru analogowego

Wywołanie **H=freqs(B,A,W)** zwraca w wektorze **H** wartości zespolone charakterystyki amplitudowo-fazowej filtru dla wartości pulsacji zawartych w wektorze **W**. Filtr określony jest transmitancją

$$\frac{B(s)}{A(s)} = \frac{b(1)s^{nb-1} + b(2)s^{nb-2} + \dots + b(nb)}{a(1)s^{na-1} + a(2)s^{na-2} + \dots + a(na)}$$

gdzie współczynniki licznika i mianownika są dane w wektorach **B** i **A**. Charakterystyki: amplitudowa i fazowa mogą być wykreślone za pomocą

```
w=logspace(d1,d2); h=freqs(b,a,w);
mag=abs(h); phase=angle(h)*180/pi;
loglog(w,mag), semilogx(w,phase)
```

freqz charakterystyka amplitudowo-fazowa filtru cyfrowego

Jeżeli **N** jest liczbą całkowitą, to za pomocą wywołania **[H,W]=freqz(B,A,N)** zwracany jest **N**-punktowy wektor częstotliwości **W** i **N**-punktowa zespolona charakterystyka częstotliwościowa **H** filtru opisanego transmitancją

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1)+b(2)z^{-1} + \dots + b(nb+1)z^{-nb}}{a(1)+a(2)z^{-1} + \dots + a(na+1)z^{-na}}$$

dla $z=e^{j\omega}$, gdzie ω jest pulsacją unormowaną względem częstotliwości próbkowania. Współczynniki licznika i mianownika transmitancji są dane w wektorach **B** i **A**. Charakterystyka częstotliwościowa jest obliczana w **N** punktach równomiernie rozłożonych na górnej połowie okręgu jednostkowego. Charakterystyki amplitudowa i fazowa mogą być wykreślone za pomocą

```
[h,w]=freqz(b,a,n); mag = abs(h); phase = angle(h);
semilogy(w, mag), plot(w, phase)
```

freqz(B,A,N,'whole') oblicza charakterystykę częstotliwościową w **N** punktach na całym okręgu jednostkowym, **freqz(B,A,W)** zwraca charakterystykę częstotliwościową obliczoną w punktach częstotliwości określonych w wektorze **W**, zazwyczaj między 0 i π .

grpdelay opóźnienie grupowe filtru cyfrowego

Wywołanie **[Gd,W]=grpdelay(B,A,N)** zwraca wektory (o długości **N**) **Gd** i **W**, zawierające opóźnienie grupowe i częstotliwości, dla których zostało ono wyznaczone. Opóźnienie grupowe jest zdefiniowane jako *minus pochodna charakterystyki fazowej względem pulsacji*. Charakterystyka opóźnieniowa jest wyznaczana w **N** punktach równomiernie rozłożonych na górnej połowie okręgu jednostkowego. Gdy **N** jest potęgą dwójki, obliczenia są szybsze, gdyż wykorzystuje się FFT. Odpowiednio **grpdelay(B,A,N,'whole')** oblicza charakterystykę częstotliwościową w **N** punktach na całym okręgu jednostkowym.

interp interpoluj próbki

czyli zwiększ częstotliwość próbkowania sygnału (poprzez interpolację dolnoprzepustową). Wywołanie **Y=interp(X,R)** przeprobkowuje dane zawarte w wektorze **X** z częstotliwością **R** razy większą niż początkowa częstotliwość próbkowania. Wynikowy wektor **Y** jest **R** razy dłuższy, tzn. **length(Y)=R*length(X)**. W procesie zmiany częstotliwości próbkowania symetryczny filtr **B** przepuszcza oryginalne dane bez zmian i interpoluje wartości między nimi w taki sposób, aby błąd średniokwadratowy między nimi a ich wartościami idealnymi był minimalny. **Y=interp(X,R,L,ALPHA)** pozwala na określenie argumentów **L** i **ALPHA**, które w przeciwnym przypadku przyjmują wartości standardowe odpowiednio 4 i 0.5; **2L** jest liczbą próbek używanych do interpolacji. Warto, by **L** było ≤ 10 . Długością filtru **B** jest **2*L*R+1**. Przyjmuje się, że sygnał ma ograniczone pasmo przy częstotliwości odcięcia $0 < \text{ALPHA} \leq 1.0$; **[Y, B]=interp(X,R,L,ALPHA)** zwraca dodatkowo współczynniki filtru interpolującego **B**. Patrz też **decimate**.

mean wartość średnia

Dla wektorów, **mean(X)** jest wartością średnią elementów wektora **X**. Dla macierzy, **mean(X)** jest wektorem wierszowym zawierającym wartości średnie każdej kolumny. Patrz też **std**.

polystab stabilizacja wielomianu

Wywołanie **polystab(A)**, gdzie **A** jest wektorem współczynników wielomianu, stabilizuje wielomian względem okręgu jednostkowego: pierwiastki, których moduły są większe niż 1.0 są odbite do wnętrza okręgu jednostkowego.

specplot **wydruk wyników funkcji spectrum**

Wywołanie **specplot(P,Fs)**, gdzie **P** jest macierzą otrzymaną w wyniku działania funkcji **spectrum**, a **Fs** jest częstotliwością próbkowania, spowoduje wyświetlenie następujących wykresów: **Pxx** - widmo gęstości mocy ciągu **X** i przedział ufnosci, **Pyy** - widmo gęstości mocy ciągu **Y** i przedział ufnosci, **abs(Txy)** - charakterystyka amplitudowa, **angle(Txy)** - charakterystyka fazowa, **Cxy** - funkcja koherencji. 95% przedziały ufnosci są wyświetlane wraz z krzywymi widmowej gęstości mocy; **specplot(P)** wykorzystuje znormalizowaną częstotliwość, **Fs = 2** tak, że 1.0 na osi częstotliwości jest połową częstotliwości próbkowania (częstotliwością Nyquista).

spectrum **estymata widma mocy**

jednego lub dwóch szeregów czasowych. Wywołanie **P=spectrum(X,Y,M)** dokonuje analizy FFT dwóch ciągów **X** i **Y** przy użyciu metody Welch estymacji widma mocy. **N**-punktowe ciągi **X** i **Y** są dzielone na **K** sekcji, długości **M** punktów każda (**M** musi być potęgą dwójki). Kolejne sekcje są przepuszczane przez okno Hanna, wyznaczane są ich **M**-punktowe FFT, które następnie są akumulowane; **spectrum** zwraca macierz o wymiarze **M/2+1** na **8**, czyli. **P=[Pxx Pyy Pxy Txy Cxy Pxxc Pyyc Pxye]** gdzie: **Pxx** - widmowa gęstość mocy wektora **X**, **Pyy** - widmowa gęstość mocy wektora **Y**, **Pxy** - wzajemna widmowa gęstość mocy, **Txy** - charakterystyka amplitudowo-fazowa (przenoszenie od **X** do **Y**; należy użyć **abs** i **angle** dla otrzymania amplitudy i fazy), **Cxy** - funkcja koherencji pomiędzy **X** i **Y**, **Pxxc**, **Pyyc**, **Pxye** - przedział ufnosci (95 procent). Patrz **specplot** - jak wyświetlić wykresy podanych wyników. **P=spectrum(X,Y,M, NOVERLAP)** powoduje, że **M**-punktowe sekcje będą zachodziły na siebie w **NOVERLAP** punktach. Wywołania **spectrum(X,M)** i **spectrum(X,M,NOVERLAP)** zwracają pojedyncze widmo mocy **Pxx** i przedział ufnosci **Pxxc**. Przeczytaj też **help** na temat **spectrum**.

square **generacja fali prostokątnej**

Wywołanie **square(T)** generuje próbki fali prostokątnej o okresie **2*pi**, określone w chwilach będących elementami wektora **T**. Wywołanie **square(T)** jest podobne do wywołania **sin(T)**, z tą różnicą, że tworzy falę prostokątną o wartościach +1 lub -1 zamiast fali sinusoidalnej. Wywołanie **square(T, DUTY)** generuje falę prostokątną o określonym współczynniku wypełnienia. Współczynnik wypełnienia **DUTY** jest procentem okresu, w którym sygnał jest dodatni. Na przykład, aby wygenerować falę prostokątną o częstotliwości 30 Hz, należy użyć instrukcji

t = 0 : 0.001 : 2.5; y = SQUARE(2*pi*30*t); plot(t, y)

std **odchylenie standardowe**

Dla wektorów, **std(X)** jest odchyleniem standardowym elementów wektora **X**. Dla macierzy, **std(X)** jest wektorem wierszowym zawierającym odchylenia standardowe każdej kolumny. Patrz też **mean**.

unwrap **"rozwijanie" fazy**

Próba „uciąglenia” fazy przez dodanie do niej **2*pi** tak, aby zamiast przeskoków od **-pi** do **pi** występowały gładkie przejścia w punktach zmiany ćwiartki układu współrzędnych. **Y=unwrap(X)** koryguje fazę w wektorze **X**. Gdy **X** jest macierzą, kąty fazowe są korygowane wzdłuż każdej kolumny. Faza MUSI być podana w radianach. Patrz też **angle**.

D.3. GRAFIKA W MATLABIE**D.3.1. Obiekty graficzne**

Obiekt graficzny	Podstawowa instrukcja wywołania
Figure	figure
Axes	axes
Line	line(X,Y) lub line(X,Y,Z)
Patch	patch(X,Y,C) lub patch(X,Y,Z,C)
Surface	surface(X,Y,Z,C) , surface(X,Y,Z) , surface(Z,C) , surface(Z)
Image	image(C) , image(X,Y,C)
Text	text(X,Y, text_string) , text(X,Y,Z, text_string)
User Interface Control	uicontrol
User Interface Menu	uimenu

figure **utwórz rysunek**

Tworzy rysunek (okno graficzne). Polecenie **figure** bez argumentu otwiera nowy rysunek (okno graficzne) i nadaje mu identyfikujący go tzw. uchwyt (handle); jest nim pierwszy wolny z numerów naturalnych (1,2, itd.). Uchwyt 0 już został nadany podstawowemu obiektowi graficznemu jakim jest monitor (to na jego

ekranie są tworzone rysunki, a na rysunkach dalsze obiekty graficzne). Wywołanie **figure(h)** powoduje, że **h**-ty rysunek stanie się rysunkiem bieżącym dla przyszłych instrukcji kreślenia. Jeżeli rysunek **h** nie istnieje, to zostanie utworzony nowy z pierwszym dostępnym uchwytem. Wywołanie **gcf** (get current figure) oddaje uchwyt bieżącego rysunku, zaś **get(h)** oddaje listę właściwości i ich aktualnych wartości dla rysunku o uchwycie **h**. Wreszcie **set(h)** oddaje listę właściwości i ich legalnych wartości dla rysunku o uchwycie **h**.

axes**kreśl osie**

Tworzy osie (układ współrzędnych dla przyszłych wykresów): **axes** bez argumentu tworzy osie standardowej wielkości (pełne okno) i nadaje im uchwyt **h**; **axes(h)** powoduje, że osie z uchwycem **h** stają się bieżące dla przyszłych instrukcji kreślenia; **axes('position', [left, bottom, width, height])** tworzy osie o zadanej lokalizacji na rysunku. Współrzędne **left, bottom** zadają położenie lewego, dolnego rogu osi, a współrzędne **width, height** zadają szerokość i wysokość współrzędnych. W znormalizowanych jednostkach współrzędne **(0,0)** odpowiadają lewemu dolnemu, a współrzędne **(1, 1)** prawemu górnemu rogowi całego rysunku, na którym umieszcza się osie. Wywołanie **get(h)** oddaje listę właściwości i ich aktualnych wartości dla osi o uchwycie **h**, zaś **set(h)** oddaje listę właściwości i ich legalnych wartości tychże osi.

line**kreśl linię**

Instrukcja **line(X,Y)** dodaje wykres linii zadanej wektorami **X,Y** do bieżących osi. Jeżeli **X** i **Y** są macierzami o tych samych wymiarach, to jest dodawana jedna linia na jedną kolumnę; **line(X,Y,Z)** z tworzy linie we współrzędnych trzywymiarowych 3-D, zaś samo **line** daje wektor kolumnowy uchwytów wszystkich obiektów będących liniami (jeden uchwyt na linię). Linie są obiektami potomnymi (dziećmi) w stosunku do osi (nie mogą istnieć bez nich). Wywołanie **get(h)** oddaje listę właściwości i ich aktualnych wartości dla linii **h**, zaś **set(h)** oddaje listę właściwości i ich legalnych wartości dla tej linii.

patch**kreśl łate**

Polecenie **patch(X,Y,C)** dodaje do bieżących osi łate o postaci wypełnionego, dwuwymiarowego wieloboku zdefiniowanego wektorami **X,Y**. Parametr **C** specyfikuje indeks koloru w mapie kolorów. Jeżeli jest to skalar, to po prostu specyfikuje on kolor wieloboku („płaskie” kolorowanie). Jeżeli jest to wektor o tej samej długości co **X** i **Y**, to jest specyfikowany kolor każdego wierzchołka wieloboku, a między wierzchołkami kolory są interpolowane biliniowo („interpolowane” kolorowanie). Jeżeli **C** jest łańcuchem 'r' - red, 'g' - green, 'b' - blue, 'c' - cyan, 'm' - magenta, 'y' - yellow, 'w' - white lub 'k' - black, to wielobok jest wypełniony danym kolorem. Jeżeli **X, Y, C** są macierzami o tych samych wymiarach, to do osi będzie dodanych tyle łate, ile jest kolumn. Wywołanie **patch(X,Y,Z,C)** tworzy łaty w przestrzeni trzywymiarowej. Łaty są obiektami potomnymi (dziećmi) w stosunku do osi. Poprzez wywołanie **patch** uzyskujemy wektor kolumnowy uchwytów wszystkich obiektów będących łatami, natomiast **get(h)** oddaje listę właściwości i ich aktualnych wartości dla łaty **h**, zaś **set(h)** oddaje listę właściwości i ich legalnych wartości dla tejże łaty.

surface**kreśl powierzchnię**

Polecenie **surface(X,Y,Z,C)** dodaje do bieżących osi kolorowe powierzchnie zdefiniowane parametrycznie współrzędnymi **X(j), Y(i), Z(i,j)**; **surface(X,Y,Z)** przyjmuje **C=Z** i kolor jest proporcjonalny do wysokości powierzchni; **surface(Z,C)** i **surface(Z)** przyjmują **X=1:n** i **Y=1:m** kreśląc jednowartościową funkcję rozpiętą nad geometrycznie prostokątną siatką. Powierzchnie są obiektami potomnymi (dziećmi) w stosunku do osi. Wywołanie **surface** oddaje uchwyt powierzchni, natomiast **get(h)** oddaje listę właściwości i ich aktualnych wartości dla powierzchni **h**, zaś **set(h)** oddaje listę właściwości i ich legalnych wartości dla tejże powierzchni.

image**wyświetl obraz**

Polecenie **image(C)** odtwarza macierz **C** jako obraz na ekranie monitora. Każdy element macierzy **C** specyfikuje kolor obrazu. Elementy **C** są wskaźnikami dla **COLORMAP**. Jeżeli **[M N]=size(C)**, to granice osi są **[0.5N+0.5 0.5M+0.5]**, tak że każdy element w **C** jest wycentrowany w całkowitych współrzędnych zmieniających się między **1** i **M** lub **N**. Wywołanie **image(X,Y,C)**, gdzie **X** i **Y** są wektorami, specyfikuje położenie środków pikseli od **C(1,1)** do **C(M,N)**. Element **C(1,1)** jest scentrowany w punkcie **(X(1), Y(1))**, element **C(M,N)** jest scentrowany w punkcie **(X(end), Y(end))**, a centra pozostałych pikseli są rozłożone równomiernie między tymi skrajnymi punktami. Obrazy są obiektami potomnymi (dziećmi) w stosunku do osi. Wywołanie **image** oddaje uchwyt obrazu, **get(h)** daje listę właściwości i ich aktualnych wartości dla obrazu **h**, zaś **set(h)** daje listę właściwości i ich legalnych wartości dla tegoż obrazu.

text**tekst**

Polecenie **text(X,Y,'łańcuch')** dodaje tekst o treści jak w łańcuchu, ulokowany w bieżących osiach w miejscu **(X,Y)**, gdzie **(X,Y)** jest w jednostkach z bieżącego wykresu. Jeżeli **X** i **Y** są wektorami, to tekst jest wpisywany we wszystkich wskazanych miejscach. Jeżeli 'łańcuch' ma tyle wierszy jaki jest wymiar **X** i **Y**, to na kolejnych pozycjach jest wpisywany tekst z kolejnych wierszy. Wywołanie **text(X,Y,Z,'łańcuch')**

dodaje tekst w przestrzeni trzywymiarowej. Tekst jest obiektem potomnym (dzieckiem) w stosunku do osi. Pisząc **text** otrzymamy w odpowiedzi wektor kolumnowy uchwytów tekstów, natomiast **get(h)** oddaje listę właściwości i ich aktualnych wartości dla tekstu **h**, a **set(h)** oddaje listę właściwości i ich legalnych wartości dla tegoż tekstu.

uicontrol **utwórz sterownik interfejsu użytkownika**

Polecenie **uicontrol('PropertyName1', value1, 'PropertyName2', value2, ...)** tworzy sterownik interfejsu użytkownika w oknie bieżącego rysunku i nadaje mu uchwyt **h**. Właściwości **uicontrol** mogą być ustawione w czasie jego tworzenia poprzez podanie **value1**, **value2**,... lub zmienione instrukcją **set**. Ponadto **get(h)** oddaje listę właściwości i ich aktualnych wartości dla sterownika o uchwycie **h**, zaś **set(h)** oddaje listę właściwości i ich legalnych wartości dla tego sterownika.

uimenu **utwórz menu interfejsu użytkownika**

Polecenie **uimenu('PropertyName1', value1, 'PropertyName2', value2,)** tworzy menu na pasku na górze okna bieżącego rysunku i nadaje mu uchwyt **h**. Wywołaniem **uimenu(h,...)** stworzymy nowe menu potomne (dziecko) względem starego menu o uchwycie **h**. Właściwości **uimenu** mogą być ustawione w czasie jego tworzenia poprzez podanie **value1**, **value2**,... lub zmienione później instrukcją **set**, natomiast **set(h)** oddaje listę właściwości i ich aktualnych wartości dla menu o uchwycie **h**, zaś **set(h)** oddaje listę właściwości i ich legalnych wartości dla menu **h**.