

Octave to bardzo złożony pakiet i *żadna ściągawka nie zastąpi szczegółowej dokumentacji* — tym bardziej, że używanie niektórych funkcji numerycznych wymaga pewnej wiedzy i ostrożności. Oficjalna dokumentacja Octave dostępna jest pod adresem:

<http://www.gnu.org/software/octave/doc>.

W tekście poniżej, nawiasy klamrowe {} oznaczają argumenty opcjonalne, a wielokropek ... zastępuje precyzyjną listę argumentów.

Rozpoczynanie pracy

<code>octave</code>	Uruchomienie standardowej sesji Octave w trybie interakcyjnym
<code>octave file</code>	Uruchomienie Octave i wykonanie poleceń zawartych w pliku <code>file</code>
<code>octave --eval code</code>	Wykonanie sekwencji poleceń <code>code</code>
<code>octave --help</code>	Wypisuje dostępne opcje uruchomienia Octave

Kończenie pracy

<code>quit</code> or <code>exit</code>	Zamknięcie sesji i wyjście z Octave
<code>INTERRUPT</code>	(<i>Np.</i> <code>Ctrl+c</code>) zakończenie wykonywania bieżącego polecenia i powrót do linii komend

Pomoc

<code>help</code>	Lista wszystkich poleceń i wbudowanych zmiennych
<code>help command</code>	Krótki opis polecenia <code>command</code>
<code>doc</code>	Przeglądanie wersji Info manuala Octave
<code>doc command</code>	Opis polecenia <code>command</code> w manualu Octave
<code>lookfor str</code>	Szuka w manualu Octave hasła <code>str</code>

Nawigacja w Info

<code>SPACE</code> albo <code>Ctrl+v</code>	Przewija jeden ekran do przodu
<code>DEL</code> albo <code>Alt+v</code>	Przewija jeden ekran do tyłu
<code>Ctrl+l</code>	Odswieża ekran

Wybór sekcji Info

<code>n</code>	Następny rozdział
<code>p</code>	Poprzedni rozdział
<code>u</code>	Rozdział 'piętro wyżej'
<code>t</code>	Początek bieżącego manuala
<code>d</code>	Spis treści Info
<code><</code>	Pierwsze hasło w bieżącym manualu
<code>></code>	Ostatnie hasło w bieżącym manualu
<code>g</code>	Próbuje odnaleźć rozdział o podanej nazwie
<code>Ctrl+x k</code>	Zamyka bieżący rozdział

Wyszukiwanie w Info

<code>s</code>	Wyszukuje podany tekst
<code>Ctrl+s</code>	Wyszukuje kolejne pojawienia się tekstu
<code>Ctrl+r</code>	To samo, ale do tyłu
<code>i</code>	Przeszukuje indeks i przechodzi do hasła
<code>,</code>	Znajduje następne pasujące hasło z komendy 'i'

Nawigacja w linii poleceń

<code>Ctrl+b</code> albo ←	Przesuń kursor o jeden znak w lewo
<code>Ctrl+f</code> albo →	Przesuń kursor o jeden znak w prawo
<code>Ctrl+a</code> albo <code>HOME</code>	Przesuń kursor na początek linii
<code>Ctrl+e</code> albo <code>END</code>	Przesuń kursor na koniec linii
<code>Alt+f</code> albo <code>Ctrl+→</code>	Przeskocz słowo w prawo
<code>Alt+b</code> albo <code>Ctrl+←</code>	Przeskocz słowo w lewo
<code>Ctrl+l</code>	Wyczyszczenie ekranu terminala

Wpisywanie tekstu

<code>Alt+TAB</code>	Wstawia znak tabulacji
<code>DEL</code>	Kasuje znak na lewo od kursora
<code>Ctrl+d</code> albo <code>BACKSPACE</code>	Kasuje znak wskazywany przez kursor
<code>Ctrl+t</code>	Zamienia ze sobą znak wskazywany przez kursor i sąsiadujący z nim (z lewej)

Wycinanie i wklejanie

<code>Ctrl+k</code>	Wycina tekst od kursora do końca linii
<code>Ctrl+y</code>	Wkleja ostatnio wycięty tekst
<code>Alt+d</code>	Wycina tekst od kursora do końca słowa

Wprowadzanie komend i historia

<code>TAB</code>	Uzupełnienie wpisywanej komendy lub nazwy
<code>ENTER</code>	Wprowadzenie bieżącego polecenia
<code>Ctrl+p</code> albo ↑	Poprzednio wydana komenda
<code>Ctrl+n</code> albo ↓	Następnie wydana komenda
<code>Alt+<</code>	Pierwsza komenda na liście
<code>Alt+></code>	Ostatnia komenda na liście
<code>Ctrl+r</code>	Szuka komendy wstecz na liście
<code>history {-q}</code>	Wypisz <i>N</i> ostatnich komend, <i>-q</i> pomija numery komend
<code>history -w {file}</code>	Zapisuje historię do <code>file</code> (do <code>~/octave_hist</code> gdy nie podano argumentu <code>file</code>)
<code>history -r {file}</code>	Wczytuje historię z <code>file</code> (z <code>~/octave_hist</code> jeśli nie podano <code>file</code>)
<code>run_history {beg} {end}</code>	Wykonuje polecenia zapisane w historii. <code>{beg}</code> <code>{end}</code> określają numer pierwszej i ostatniej komendy. Bez argumentów wykonuje ostatnio wydaną komendę.

Polecenia systemowe

<code>cd dir</code>	Zmienia bieżący katalog na <code>dir</code>
<code>pwd</code>	Wypisuje nazwę bieżącego katalogu
<code>ls {...}</code>	Wyświetla zawartość bieżącego katalogu
<code>getenv (string)</code>	Zwraca wartość danej zmiennej środowiskowej
<code>system (cmd)</code>	Wykonuje zadaną sekwencję komend systemowych <code>cmd</code>

Macierze

<code>[x, y, ...]</code>	Wektor wierszowy
<code>[x; y; ...]</code>	Wektor kolumnowy
<code>[w, x; y, z]</code>	Macierz 2×2 postaci $\begin{pmatrix} w & x \\ y & z \end{pmatrix}$

Zawartość macierzy wpisujemy wewnątrz nawiasów kwadratowych. Przecinki oddzielają kolejne elementy w wierszu. Średnik oddziela wiersze. Zamiast przecinka można użyć spacji, a zamiast średnika — przejść do nowego wiersza. Elementami macierzy mogą być dowolne wyrażenia dobrane tak, by zgadzały się rozmiary.

Macierze *N*-wymiarowe

<code>squeeze (arr)</code>	Usuwa singletonowe wymiary z macierzy <code>arr</code>
<code>ndims (arr)</code>	Liczba wymiarów macierzy
<code>permute (arr, p)</code>	Permutuje wymiary macierzy
<code>ipermute (arr, p)</code>	Odwrotna permutacja wymiarów
<code>shiftdim (arr, s)</code>	Rotacja wymiarów macierzy
<code>circshift (arr, s)</code>	Rotacja elementów macierzy

Macierze *N*-wymiarowe można tworzyć z macierzy niższego wymiaru, korzystając z poleceń `cat` lub `reshape`.

Macierze rzadkie

<code>sparse (...)</code>	Tworzy macierz rzadką
<code>speye (n)</code>	Tworzy macierz identyczności w formacie macierzy rzadkiej
<code>sprand (n, m, d)</code>	Rzadka macierz $n \times m$ o wypełnieniu <i>d</i>
<code>spdiags (...)</code>	Uogólnienie <code>diag</code> na macierze rzadkie
<code>nnz (s)</code>	Liczba niezerowych elementów macierzy

Zakresy indeksów

<code>base : {incr :}</code>	Generuje zestaw kolejnych liczb od <code>base</code> , z odstępem <code>incr</code> , do <code>limit</code> . Jeśli <code>incr</code> pominiemy, odstęp wyniesie 1. Dozwolone są ujemne i ułamkowe odstęp.
------------------------------	--

Selekcja elementów macierzy

<code>var (idx)</code>	Wybiera z wektora elementy o zadanych indeksach
<code>var (idx1, idx2)</code>	Wybiera z macierzy elementy o zadanych indeksach
<code>scalar</code>	Możliwe postaci indeksów elementów:
<code>vector</code>	Indeks o numerze <code>scalar</code>
<code>range</code>	Numery indeksów w wektorze <code>vector</code>
<code>:</code>	Numery indeksów w postaci zakresu <code>range</code>
<code>:</code>	Wszystkie dopuszczalne elementy

Napisy i znaki specjalne

<code>\\</code>	Znak „backslash”
<code>\"</code>	Cudzysłów podwójny
<code>\'</code>	Cudzysłów pojedynczy
<code>\n</code>	Znak nowej linii, ASCII code 10
<code>\t</code>	Znak tabulacji, ASCII code 9

Napis to sekwencja znaków otoczona cudzysłowami. Użycie podwójnych cudzysłów umożliwia stosowanie powyższych znaków specjalnych wewnątrz napisu.

Zmienne globalne i statyczne

<code>global var1 ...</code>	Deklaruje zmienną jako globalną
<code>global var1 = val</code>	Deklaruje zmienną jako globalną i nadaje jej wartość
<code>persistent var1</code>	Deklaruje zmienną w funkcji jako statyczną
<code>persistent var1 = val</code>	Deklaruje zmienną w funkcji jako statyczną i nadaje jej wartość początkową

Aby zmienna globalna była widoczna we wnętrzu funkcji, musi wcześniej być zadeklarowana jako globalna zarówno w funkcji, jak i w sesji Octave.

Wybrane „stałe”

<code>Inf, NaN</code>	Nieskończoność i not-a-number zgodnie z IEEE 754
<code>NA</code>	Brak danych
<code>ans</code>	Zmienna, której domyślnie przypisywany jest wynik wyrażenia wprowadzonego z linii komend
<code>eps</code>	Epsilon maszynowy
<code>pi</code>	Liczba π
<code>1i</code>	Jednostka urojona, $\sqrt{-1}$
<code>realmax</code>	Największa reprezentowalna liczba zmiennoprzecinkowa
<code>realmin</code>	Najmniejsza reprezentowalna (dodatnia) liczba zmiennoprzecinkowa

W rzeczywistości są to funkcje.

Przypisanie

<code>var = expr</code>	Wartość wyrażenia <code>expr</code> przypisuje zmiennej <code>var</code>
<code>var (idx) = expr</code>	Wartość wyrażenia przypisuje elementom wektora o zadanych indeksach
<code>var (idx) = []</code>	Usuwa elementy o zadanych indeksach
<code>var {idx} = expr</code>	Wartość wyrażenia przypisuje zadanym elementom zmiennej tablicowej

Struktury i tablice

<code>var.field = ...</code>	Nadaje wartość polu <code>field</code> struktury <code>var</code>
<code>var{idx} = ...</code>	Nadaje wartość elementowi zmiennej tablicowej
<code>cellfun(f, c)</code>	Przykłada funkcję do elementów zmiennej tablicowej
<code>fieldnames(s)</code>	Zwraca nazwy pól struktury

Operatory arytmetyczne

<code>x + y</code>	Dodawanie
<code>x - y</code>	Odejmowanie
<code>x * y</code>	Mnożenie (macierzowe)
<code>x .* y</code>	Mnożenie elementów
<code>x / y</code>	Prawostronny operator rozwiązywania układu równań liniowych, <i>formalnie</i> równoważny obliczeniu $(\text{inv}(y') * x')$
<code>x ./ y</code>	Prawostronne dzielenie elementów
<code>x \ y</code>	Lewostronny operator rozwiązywania układu równań liniowych, <i>formalnie</i> równoważny obliczeniu $\text{inv}(x) * y$
<code>x .\ y</code>	Lewostronne dzielenie elementów
<code>x ^ y</code>	Potęgowanie (macierzowe)
<code>x .^ y</code>	Potęgowanie elementów
<code>-x</code>	Znak -
<code>+x</code>	Znak +
<code>x '</code>	Sprzężenie zespolone
<code>x .'</code>	Transpozycja
<code>++x (--x)</code>	Inkrement (dekrement), zwraca <i>przyjętą</i> wartość
<code>x++ (x--)</code>	Inkrement (dekrement), zwraca <i>starą</i> wartość

Porównania i operatory logiczne

<code>x < y</code>	<code>true</code> , jeśli <code>x</code> mniejsze od <code>y</code>
<code>x <= y</code>	<code>true</code> , jeśli <code>x</code> mniejsze lub równe <code>y</code>
<code>x == y</code>	<code>true</code> , jeśli <code>x</code> równe <code>y</code>
<code>x >= y</code>	<code>true</code> , jeśli <code>x</code> większe lub równe <code>y</code>
<code>x > y</code>	<code>true</code> , jeśli <code>x</code> większe od <code>y</code>
<code>x != y</code>	<code>true</code> , jeśli <code>x</code> różne od <code>y</code>
<code>x & y</code>	<code>true</code> , jeśli jednocześnie <code>x</code> i <code>y</code> są <code>true</code> (logiczne „AND”)
<code>x y</code>	<code>true</code> , jeśli przynajmniej jedno: <code>x</code> lub <code>y</code> jest <code>true</code> (logiczne „OR”)
<code>! bool</code>	<code>true</code> , jeśli <code>bool</code> jest <code>false</code> (logiczne „NOT”)

Działają na elementach macierzy. Oba argumenty są wyznaczone przed zwróceniem wyniku.

Skrócone operacje logiczne

<code>x && y</code>	<code>true</code> , jeśli jednocześnie <code>x</code> i <code>y</code> są <code>true</code>
<code>x y</code>	<code>true</code> , jeśli przynajmniej jedno: <code>x</code> lub <code>y</code> jest <code>true</code>

Wartość operatora jest wyznaczana od lewej do prawej, a wartości operandów są obliczane tylko do momentu, gdy wartość całego wyrażenia może zostać prawidłowo określona. Operandy są uprzednio przekształcane do skalarów funkcją `all`.

Priorytet operatorów

<code>;</code>	Separatory poleceń
<code>=</code>	Przypisanie, grupuje się od lewej do prawej
<code> &&</code>	Logiczne „OR” oraz „AND”
<code> &</code>	Skrócone logiczne „OR” oraz „AND”
<code>< <= == >= ></code>	Porównania
<code>!=</code>	
<code>:</code>	Indeksy
<code>+ -</code>	Dodawanie i odejmowanie
<code>* / \ .* ./</code>	Mnożenie i dzielenie
<code>.\</code>	
<code>' .'</code>	Sprzężenie, transpozycja
<code>+ - ++ --</code>	Znak plus/minus, inkrement/dekrement, logiczne „NOT”
<code>! ^ .^</code>	Potęgowanie

Uporządkowane według rosnącego priorytetu.

Ścieżki i pakiety

<code>path</code>	Wyświetla listę katalogów, w których Octave szuka funkcji
<code>pathdef</code>	Wyświetla domyślną wartość <code>path</code>
<code>addpath(dir)</code>	Dodaje katalog do <code>path</code>
<code>EXEC_PATH</code>	Manipulacja listą katalogów w których Octave szuka zewnętrznych programów wykonywalnych
<code>pkg list</code>	Lista zainstalowanych pakietów rozszerzeń
<code>pkg describe {-verbose} pack</code>	Opis {pogłębiony} wskazanego pakietu
<code>pkg load pack</code>	Ładuje wskazany pakiet

Instrukcje sterujące

<code>for identifier</code>	Wykonuje <i>stmt-list</i> w pętli po kolumnach <i>expr</i> .
<code>= expr stmt-list</code>	W każdym obrocie pętli zmienna <i>identifier</i> przyjmuje wartość bieżącej kolumny.
<code>endfor</code>	
<code>while (condition) stmt-list</code>	Powtarza instrukcje <i>stmt-list</i> dopóki <i>condition</i> ma wartość <code>true</code> .
<code>endwhile</code>	
<code>break</code>	Wychodzi z najbardziej zagnieżdżonej pętli
<code>continue</code>	Przechodzi na początek najbardziej zagnieżdżonej pętli
<code>return</code>	Kończy działanie bieżącej i wraca do wywołującej funkcji
<code>if (condition) if-body {else else-body} endif</code>	Wykonuje instrukcje <i>if-body</i> pod warunkiem, że <i>condition</i> ma wartość <code>true</code> , w przeciwnym razie wykonują się instrukcje <i>else-body</i> .
<code>if (condition) if-body {elseif (condition) elseif-body} {else else-body} endif</code>	Wykonuje instrukcje <i>if-body</i> pod warunkiem, że <i>condition</i> ma wartość <code>true</code> , w przeciwnym razie wykonuje instrukcje z pierwszego <i>elseif-body</i> dla którego warunek <i>elseif</i> jest spełniony; gdy żaden z warunków nie jest spełniony, wykonuje <i>else-body</i> . W instrukcji <i>if</i> może znaleźć się „dowolna” liczba rozgałęzień <i>elseif</i> .
<code>switch (expr) case label case-body {case label case-body} {otherwise otherwise-body} endswitch</code>	Wykonuje ten zestaw instrukcji <i>case-body</i> , dla którego <i>label</i> jest równa wartości <i>expr</i> . W przeciwnym razie wykonuje <i>otherwise-body</i> . Działa także wtedy, gdy <i>label</i> jest tekstem. W instrukcji <i>switch</i> może znaleźć się „dowolna” liczba rozgałęzień <i>case</i> .
<code>unwind_protect body</code>	Wykonuje <i>body</i> . Następnie wykonuje <i>cleanup</i> bez względu na to, czym zakończyło się wykonanie <i>body</i> .
<code>unwind_protect_cleanup</code>	czyszczenie <i>body</i> .
<code>cleanup end</code>	
<code>try body catch cleanup end</code>	Wykonuje <i>body</i> . Następnie wykonuje <i>cleanup</i> , o ile wykonanie <i>body</i> zakończy się przedwcześnie (np. błędem).

Funkcje tekstowe

<code>strcmp (s, t)</code>	Porównuje teksty
<code>strcat (s, t, ...)</code>	Łączy teksty
<code>regexp (str, pat)</code>	Wyszukiwanie według wyrażenia regularnego
<code>regexprep (str, pat, rep)</code>	Wyszukiwanie i zastępowanie według wyrażenia regularnego

Definiowanie funkcji

<code>function {ret-list} function-name { (arg-list) } function-body endfunction</code>	<i>ret-list</i> może być pojedynczym identyfikatorem lub otoczoną nawiasami kwadratowymi listą oddzielonych przecinkami identyfikatorów. <i>arg-list</i> jest listą oddzielonych przecinkami argumentów; może być pusta.
---	--

Wskaźniki do funkcji

<code>@func</code>	Wyluskuje wskaźnik do <i>func</i> .
<code>@(var1, ...)</code>	Definiuje funkcję anonimową i zwraca wskaźnik do niej.
<code>str2func (str)</code>	Tworzy wskaźnik na podstawie nazwy funkcji.
<code>functions (handle)</code>	Zwraca informacje o konkretnym wskaźniku.
<code>func2str (handle)</code>	Zwraca nazwę funkcji o danym wskaźniku.
<code>handle (arg1, ...)</code>	Dla podanych argumentów wyznacza wartość funkcji o wskaźniku <i>handle</i> .
<code>feval (func, arg1, ...)</code>	Dla podanych argumentów wyznacza wartość funkcji o wskaźniku lub nazwie <i>func</i> .

Funkcje anonimowe używają kopii zmiennych z przestrzeni roboczej sesji Octave, o początkowych wartościach takich, jak w chwili definiowania funkcji.

Różne użyteczne funkcje

<code>eval (str)</code>	Wykonuje <i>str</i> jako polecenie Octave
<code>error (msg)</code>	Wyświetla komunikat o błędzie <i>msg</i> i powraca do sesji Octave
<code>warning (msg)</code>	Wyświetla ostrzeżenie <i>msg</i> i działa dalej
<code>clear pattern</code>	Usuwa zmienne odpowiadające wzorcowi <i>pattern</i>
<code>exist (str)</code>	Sprawdza, czy istnieje zmienna lub funkcja o nazwie <i>str</i>
<code>who, whos</code>	Podaje informacje o zdefiniowanych obiektach
<code>whos var</code>	Szczegółowe informacje o zmiennej <i>var</i>

Podstawowe funkcje macierzowe

<code>rows (a)</code>	Liczba wierszy <i>a</i>
<code>columns (a)</code>	Liczba kolumn <i>a</i>
<code>all (a)</code>	Czy wszystkie elementy <i>a</i> są niezerowe
<code>any (a)</code>	Czy jakikolwiek element <i>a</i> jest niezerowy
<code>find (a)</code>	Zwraca współrzędne niezerowych elementów <i>a</i>
<code>sort (a)</code>	Sortuje elementy każdej kolumny <i>a</i>
<code>sum (a)</code>	Sumuje wartości w każdej kolumnie <i>a</i>
<code>prod (a)</code>	Mnoży wartości w każdej kolumnie <i>a</i>
<code>min (...)</code>	Minimum
<code>max (...)</code>	Maksimum
<code>rem (x, y)</code>	Reszta z dzielenia <i>x/y</i>
<code>reshape (a, m, n)</code>	Zmienia rozmiar <i>a</i> na $m \times n$
<code>diag (v, k)</code>	Tworzy macierz diagonalną
<code>linspace (b, l, n)</code>	Tworzy zestaw równoodległych węzłów
<code>logspace (b, l, n)</code>	Tworzy zestaw węzłów równoodległych w skali logarytmicznej
<code>eye (n, m)</code>	Tworzy macierz identyczności $n \times m$
<code>ones (n, m)</code>	Tworzy macierz jedynkową $n \times m$
<code>zeros (n, m)</code>	Tworzy macierz zerową $n \times m$
<code>rand (n, m)</code>	Tworzy macierz losową $n \times m$

Wybrane elementarne funkcje matematyczne

<code>log (a)</code>	Funkcje logarytmiczne; <code>log</code> to logarytm naturalny
<code>log10 (a)</code>	
<code>log2 (a)</code>	
<code>exp (a), sqrt (a), pow (a, b)</code>	e^a, \sqrt{a}, a^b
<code>sin (a) cos (a)</code>	Funkcje trygonometryczne
<code>tan (a) asin (a)</code>	
<code>acos (a)</code>	
<code>atan (a)</code>	
<code>sinh (a)</code>	
<code>cosh (a)</code>	
<code>tanh (a)</code>	
<code>asinh (a)</code>	
<code>acosh (a)</code>	
<code>atanh (a)</code>	

Algebra liniowa

<code>chol (A)</code>	Rozkład Cholesky'ego–Banachiewicza
<code>det (A)</code>	wyznacznik
<code>eig (A)</code>	Wartości i wektory własne
<code>eigs (A)</code>	Wybrane wartości i wektory własne
<code>expm (A)</code>	Macierzowa funkcja wykładnicza, e^A
<code>lu (A)</code>	Rozkład LU
<code>inv (A)</code>	Odwrotność <i>A</i>
<code>norm (A, p)</code>	<i>p</i> -ta norma macierzowa
<code>pinv (A)</code>	Pseudoodwrotność <i>A</i>
<code>pcg (A, b)</code>	Rozwiązanie układu $Ax = b$ metodą gradientów sprzężonych
<code>qr (A)</code>	Rozkład QR
<code>rank (A)</code>	Rząd macierzy
<code>sprank (A)</code>	Rząd strukturalny macierzy rzadkiej
<code>schur (A)</code>	Rozkład Schura
<code>svd (A)</code>	Rozkład SVD
<code>syl (A, B, C)</code>	Rozwiązuje równanie Sylwestera

Układ równań $Ax = b$ należy standardowo rozwiązywać przez $x=A \backslash b$.

Równania nieliniowe, różniczkowe, różniczkowo-algebraiczne. Całkowanie

<code>fzero (...)</code>	Rozwiązuje skalarne równanie nieliniowe
<code>fsolve (...)</code>	Rozwiązuje układ równań nieliniowych
<code>lsode (...)</code>	Rozwiązuje układ równań różniczkowych zwyczajnych
<code>dassl (...)</code>	Rozwiązuje układ równań różniczkowo-algebraicznych
<code>quad (...)</code>	Oblicza całkę funkcji jednej zmiennej
<code>perror (nm, code)</code>	Jeśli funkcja zwraca numeryczny kod błędu, wyświetla odpowiadający mu komunikat dla użytkownika

W manualu znajdują się szczegółowe informacje o parametrach i sposobie wywołania tych funkcji. Należy pamiętać, że wyznaczają one jedynie pewne przybliżenie rozwiązania.

Zbiory

<code>create_set (a, b)</code>	Tworzy wektor wierszowy unikalnych wartości
<code>complement (a, b)</code>	Elementy b , których nie ma w a
<code>intersection (a, b)</code>	Przecięcie zbiorów a i b
<code>union (a, b)</code>	Suma (teoriomnościowa) zbiorów a i b

Przetwarzanie sygnałów

<code>fft (a)</code>	Dyskretna transformacja Fouriera z wykorzystaniem FFTW
<code>ifft (a)</code>	Transformacja odwrotna do FFT
<code>freqz (...)</code>	Filtr częstotliwości
<code>filter (a, b, x)</code>	Rozwiązuje proste równanie różnicowe
<code>conv (a, b)</code>	Splot dwóch wektorów

W manualu znajdują się szczegółowe informacje o parametrach i sposobie wywołania tych funkcji.

Przetwarzanie obrazów

<code>colormap (map)</code>	Ustawia bieżącą gamę kolorów
<code>gray2ind (i, n)</code>	Zamienia obraz w skali szarości na (indeksowany) obraz Octave
<code>image (img, zoom)</code>	Wyświetla obraz Octave
<code>imagesc (img, zoom)</code>	Wyświetla macierz jako wyskalowany obraz
<code>imshow (img, map)</code>	Wyświetla obraz Octave
<code>imshow (i, n)</code>	Wyświetla obraz Octave w skali szarości
<code>imshow (r, g, b)</code>	Wyświetla obraz RGB
<code>ind2gray (img, map)</code>	Zamienia (indeksowany) obraz Octave na obraz w skali szarości
<code>ind2rgb (img, map)</code>	Zamienia (indeksowany) obraz Octave na RGB
<code>loadimage (file)</code>	Wczytuje z pliku obraz Octave
<code>rgb2ind (r, g, b)</code>	Zamienia RGB na (indeksowany) obraz Octave
<code>saveimage (file, img, fmt, map)</code>	Zapisuje macierz img do pliku $file$ w jednym z kilku formatów
<code>print (file, fmt)</code>	Zapisuje obecnie wyświetlany obraz do pliku $file$ w jednym z wielu formatów

Funkcje wejścia–wyjścia

<code>save {options} filename var ...</code>	Zapisuje wskazane zmienne do pliku o nazwie $filename$
<code>load filename {var} ...</code>	Wczytuje {wskazane} zmienne z pliku o nazwie $filename$
<code>disp (var)</code>	Wyświetla na ekranie bieżącą wartość zmiennej var

Wejście–wyjście w stylu języka C

<code>fopen (filename, mode)</code>	Otwiera plik o nazwie $filename$
<code>fclose (file)</code>	Zamyka plik $file$
<code>printf (fmt, ...)</code>	Formatowane wyjście na <code>stdout</code>
<code>fprintf (file, fmt, ...)</code>	Formatowane wyjście do pliku $file$
<code>sprintf (fmt, ...)</code>	Formatowane wyjście do stringa
<code>scanf (fmt)</code>	Formatowane wejście z <code>stdin</code>
<code>fscanf (file, fmt)</code>	Formatowane wejście z pliku $file$
<code>sscanf (str, fmt)</code>	Formatowane wejście z napisu str
<code>fgets (file, len)</code>	Wczytuje len znaków z pliku $file$
<code>fflush (file)</code>	Oczyszcza bufor pliku $file$
<code>ftell (file)</code>	Bieżąca pozycja w pliku
<code>frewind (file)</code>	Przewija na początek pliku
<code>freport</code>	Informacja o otwartych plikach
<code>fread (file, size, prec)</code>	Wczytywanie danych z pliku w formacie binarnym
<code>fwrite (file, size, prec)</code>	Zapisywanie danych do pliku w formacie binarnym
<code>feof (file)</code>	Sprawdza, czy jest na końcu pliku

Do pliku można odwoływać się albo przez nazwę, albo przez deskryptor zwracany przez `fopen`. Po uruchomieniu, Octave otwiera trzy pliki: `stdin`, `stdout`, oraz `stderr`.

Wielomiany

<code>compan (p)</code>	Macierz, której wartości własne są równe pierwiastkom wielomianu p
<code>conv (a, b)</code>	Współczynniki iloczynu wielomianów a i b
<code>deconv (a, b)</code>	Dzielenie wielomianów
<code>poly (a)</code>	Wielomian charakterystyczny macierzy a
<code>polyderiv (p)</code>	Pochodna wielomianu
<code>polyint (p)</code>	Całka nieoznaczona z wielomianu
<code>polyval (p, x)</code>	Wyznacza wartość wielomianu w węzłach x
<code>polyvalm (p, X)</code>	Wyznacza w sensie macierzowym wartość wielomianu na macierzy kwadratowej X
<code>roots (p)</code>	Wyznacza wszystkie pierwiastki wielomianu
<code>residue (a, b)</code>	Wyznacza rozkład funkcji wymiernej a/b na ułamki proste

Statystyka

<code>corrcoef (x, y)</code>	Współczynnik korelacji
<code>cov (x, y)</code>	Macierz kowariancji
<code>mean (a)</code>	Średnia
<code>median (a)</code>	Mediana
<code>std (a)</code>	Odchylenie standardowe
<code>var (a)</code>	Wariancja

Wykresy

<code>plot (...)</code>	Wykres krzywej 2D w skali liniowej
<code>plot3 (...)</code>	Wykres krzywej 3D w skali liniowej
<code>line (...)</code>	Linia w 2D lub 3D
<code>patch (...)</code>	Wielokąt w 2D
<code>semilogx (...)</code>	Wykres krzywej 2D w skali logarytmicznej na osi OX
<code>semilogy (...)</code>	Wykres krzywej 2D w skali logarytmicznej na osi OY
<code>loglog (...)</code>	Wykres krzywej 2D w skali logarytmicznej na obu osiach
<code>bar (...)</code>	Wykres słupkowy
<code>stairs (x, y)</code>	Wykres schodkowy
<code>stem (x, y)</code>	Wykres drążkowy
<code>hist (y, x)</code>	Histogram
<code>contour (x, y, z)</code>	Wykres poziomicowy
<code>title (string)</code>	Nadaje wykresowi tytuł
<code>axis (limits)</code>	Parametry osi wykresu
<code>xlabel (string)</code>	Nadaje nazwę osi OX
<code>ylabel (string)</code>	Nadaje nazwę osi OY
<code>zlabel (string)</code>	Nadaje nazwę osi OZ
<code>text (x, y, str)</code>	Do wykresu dodaje tekst str w pozycji (x, y)
<code>legend (string)</code>	Dodaje do wykresu legendę
<code>grid {on off}</code>	Przełącza wyświetlanie siatki na wykresie
<code>hold {on off}</code>	Przełącza tryb dorysowywania do bieżącego wykresu
<code>ishold</code>	Sprawdza, czy jesteśmy w trybie dorysowywania
<code>mesh (x, y, z)</code>	Wykres powierzchni 3D
<code>meshgrid (x, y)</code>	Tworzy macierze współrzędnych
<code>print (file, fmt)</code>	Zapisuje obecnie wyświetlany obraz do pliku $file$ w jednym z wielu formatów

Wydanie polskie 1.0 dla Octave 3.0 na podstawie wydania angielskiego 2.0 dla Octave 3.0.0. Copyright 1996, 2007, John W. Eaton (jwe@octave.org). Tłumaczenie, opracowanie i makra L^AT_EX'owe: Piotr Krzyżanowski (piotr.krzyzanowski@mimuw.edu.pl). Możesz do woli rozpowszechnić ten dokument na zasadach GNU General Public License.

Ani John W. Eaton, ani Piotr Krzyżanowski, nie ponoszą żadnej odpowiedzialności za błędy znajdujące się w tym dokumencie.

Octave to wolne oprogramowanie; możesz do woli je rozpowszechnić na zasadach GNU General Public License. Nie ma żadnych gwarancji związanych z Octave.