

Ćwiczenie Oct

Wprowadzenie do Octave

Cześć 1. Zmienne i wyrażenia, operacje na tablicach

Polecenia w Octave możemy wpisywać bezpośrednio w oknie poleceń lub zapisywać całą ich sekwencję w pliku tekstowym, tzw. skrypcie lub m-pliku. Skrypty można tworzyć we wbudowanym w GUI edytorze (zalecane). Przykładowe polecenia Octave zamieszczono poniżej.

Polecenie	Znaczenie
% lub #	komentarz
a = 5;	nadanie zmiennej a wartości 5. Jeśli pominiemy znak ; wartość tej zmiennej wyświetli się
log(a)	wywołanie funkcji bibliotecznej z argumentem a
b = sin(pi/2)	zapamiętanie w zmiennej b wyniku działania funkcji sinus z argumentem pi/2
c = [1 2 3]	utworzenie wektora o trzech elementach
d = [1 2; 3 4; 5 6]	utworzenie macierzy d o wymiarach 3x2. Znak ; oznacza tu oddzielenie wierszy. Elementy tego samego wiersza oddziela się spacjami lub przecinkami
d(1,2)	odwołanie do elementu w 1-szym rzędzie i 2-iej kolumnie. Np. d(1,2) = 5, a = d(1,2)
d(end,end)	odwołanie do elementu w ostatnim rzędzie i ostatniej kolumnie
d(:,1)	odwołanie do wszystkich elementów 1-wszej kolumny macierzy d. Np. x = d(:,1) daje x = [1; 3; 5], a d(:,1) = [0; 0; 0] daje d = [0 2; 0 4; 0 6]
d(2,:)	odwołanie do wszystkich elementów 2-go wiersza macierzy d. Np. x = d(2,:) daje x = [3 4], a d(2,:) = [0 0] daje d = [1 2; 0 0; 5 6]. Polecenie d(1,:) = d(2,:) kopiuje 2-gi wiersz macierzy d do wiersza 1-go
d(1:2,:)	odwołanie do wszystkich elementów 1-go i 2-go wiersza macierzy d
d([1 3],:)	odwołanie do wszystkich elementów 1-go i 3-go wiersza macierzy d
d(2:3,1:2)	odwołanie do elementów z 2-go i 3-go wiersza oraz z 1-wszej i 2-giej kolumny
e = 1:10	utworzenie wektora o elementach [1 2 ... 10]
f = -5:2:5	utworzenie wektora f o pierwszym elemencie -5, każdym następnym o 2 większym od poprzedniego, a ostatnim nie większym od 5
f(1)	odwołanie do 1-go elementu wektora f (np. f(1) = 2, a = f(1))
size(d)	funkcja zwracająca rozmiary macierzy (wektora), czyli liczbę wierszy i liczbę kolumn. Polecenie size(d,1) zwraca liczbę wierszy, a size(d,2) zwraca liczbę kolumn
length(c)	funkcja zwracająca długość wektora c
s = 'Matlab'	utworzenie łańcucha znaków
[]	macierz pusta, służy do usuwania wierszy lub kolumn macierzy, np. d(1,:) = [] spowoduje usunięcie 1-go wiersza macierzy d
A = [B; C]	utworzenie macierzy A z macierzy B i C w ten sposób, że do każdej kolumny macierzy B dopisywane są odpowiednie kolumny macierzy C. Macierze B i C muszą mieć tę samą liczbę kolumn

<code>A = [B C]</code>	utworzenie macierzy A z macierzy B i C w ten sposób, że do każdego wiersza macierzy B dopisywane są odpowiednie wiersze macierzy C. Macierze B i C muszą mieć tę samą liczbę wierszy
<code>A = rand(3, 4)</code>	utworzenie macierzy o rozmiarach 3x4 i wypełnienie jej liczbami losowymi z zakresu [0, 1], z rozkładu jednostajnego
<code>A = rand(4)</code>	utworzenie macierzy o rozmiarach 4x4 i wypełnienie jej liczbami losowymi z zakresu [0, 1], z rozkładu jednostajnego
<code>A = zeros(3, 4)</code>	utworzenie macierzy o rozmiarach 3x4 i wypełnienie jej zerami
<code>A = ones(3, 4)</code>	utworzenie macierzy o rozmiarach 3x4 i wypełnienie jej jedynkami
<code>A+B, A.*B</code>	dodanie lub pomnożenie odpowiadających sobie elementów macierzy A i B (macierze muszą mieć jednakowe rozmiary)
<code>clear</code>	usunięcie wszystkich zmiennych ze środowiska
<code>save plik.txt a -ascii</code>	zapisanie zmiennej a do pliku tekstowego plik.txt
<code>load plik.txt</code>	załadowanie zmiennej z pliku plik.txt (zmienna ma teraz nazwę plik)
<code>save plik.mat a b</code>	zapisanie zmiennych a i b do pliku
<code>load plik</code>	załadowanie zmiennych z pliku plik.mat
<code>help nazwa_m_pliku</code>	pomoc na temat funkcji bibliotecznej lub m-pliku, np. <code>help sin</code>
<code>Ctrl+C</code>	przerwanie działania uruchomionego programu

Zadania

1. Przećwicz w środowisku Octave powyższe polecenia.
2. Utwórz w dowolny sposób macierze A i B o wymiarach 4x4 (każda sekcja tworzy inne macierze).
3. Wykonaj na tych macierzach podstawowe operacje arytmetyczne: $A+B$, $A-B$, $A*B$, $A.*B$, A/B , $A./B$, $A\backslash B$, $A.\backslash B$, A^2 , $A.^2$, A' , $A.'$ (opis operacji w helpie).
4. Pobierz z macierzy A 1-wszy wiersz (nazwij go x) i z macierzy B 2-gą i 3-cią kolumnę (nazwij je y). Podaj polecenia Matlaba (wraz z wynikami), które wykonają następujące działania:
 - a. $3x * 5y$
 - b. $(4x^T * x)^2 * 2y$
 - c. odejmij od każdego elementu macierzy y liczbę π i pomnóż każdy element macierzy wynikowej przez ostatni element wektora x . Transponuj tak otrzymaną macierz i nazwij ją z . Następnie utwórz nową macierz q złożoną z macierzy z (macierz z stanowi 1 i 2 wiersz macierzy q) oraz z transponowanej macierzy y (ta macierz stanowi 3 i 4 wiersz macierzy q).

Część 2. Instrukcje i funkcje

Przykłady instrukcji

`sum(A)` – suma elementów wektora A lub, jeśli A jest macierzą, suma poszczególnych kolumn macierzy

`[wart, ind] = max(A)` – gdy A jest wektorem, zwraca element o największej wartości (wartość ta zapamiętana będzie w zmiennej `wart`, a indeks elementu o największej wartości w zmiennej `ind`),

dla macierzy – zwraca maksymalne wartości elementów w każdej kolumnie. Analogicznie działa funkcja `min()`

```
for i=1:5
    %polecenia, które wykonają się w pętli
end
    – pętla for wykonująca pięć iteracji (i=1:5). Inny przykład: for j = 2.1:-0.1:-1.5 ...
    end – pętla wykonująca się dla kolejnych wartości licznika j = 2.1, 2.0, 1.9, ..., -1.5.

while a > 0
    % polecenia, które wykonają się w pętli
end
    – pętla while wykonująca się pod warunkiem a>0

if b == 0
    %polecenia, które wykonają się, jeśli b jest równe 0 (== oznacza równość logiczną)
end

if b ~= 0
    %polecenia, które wykonają się, jeśli b nie równa się 0
else
    %polecenia, które wykonają się, jeśli b równa się 0
end
    – instrukcja warunkowa
```

`fprintf('Wartość zmiennej: %5.2f \n', x)` – funkcja wyjścia wyświetlająca komunikaty na ekranie. Ciąg wyświetlanych znaków podajemy w apostrofach. `%5.2f` to sekwencja formatująca, zamiast której podstawiona będzie wartość zmiennej `x`. `5.2` oznacza, że liczba wyświetlona będzie na 5-ciu pozycjach z dokładnością do 2-ch miejsc po przecinku. `f` oznacza liczbę zmiennopozycyjną. `\n` to znak przejścia do następnej linii.

`NaN` – *not a number*; symbol oznaczający wartość nienumeryczną, otrzymaną np. w wyniku działania `0.0/0.0`

`Inf` – symbol oznaczający nieskończoność

`plot(x,y)` – tworzenie wykresu; `x` – wektor współrzędnych `x` punktów, `y` – wektor współrzędnych `y` punktów. Np. `x=-pi:0.1:pi`; `plot(x,sin(x))`. Jeśli `x` i `y` to macierze, tworzonych jest tyle wykresów ile jest kolumn.

`figure(n)` – tworzenie nowego okna do wykresu; `x` oznacza numer okna

`hold on` – powoduje, że kolejne wykresy rysowane są bez kasowania wykresów poprzednio narysowanych.
`hold off` powoduje skasowanie wcześniejszych wykresów.

W Octave możemy tworzyć własne funkcje zapisując je w oddzielnych m-plikach. Np. poniżej podano definicję funkcji obliczającej pierwiastek równania liniowego. Definicja zaczyna się słowem `function`, dalej jest nazwa zmiennej zwracanej przez funkcję (jeśli zmiennych jest więcej wymieniamy je w nawiasach kwadratowych, np. `[x, y, z]`) i nazwa funkcji. W nawiasach wymieniamy argumenty funkcji (tu – współczynniki równania prostej). W kolejnych liniach piszemy instrukcje (tzw. ciało funkcji). **Uwaga!** plik z funkcją należy nazwać tak samo jak funkcję; w tym wypadku `rown1.m`.

```
function x = rown1(a, b)
    if a == 0
        if b ~= 0
            x = []; %jeśli a=0 i b<>0 nie zwracaj niczego (brak rozwiązania)
        else
            % ...
        end
    end
end
```

```
x = NaN; %jeśli a=0 i b=0 istnieje nieskończenie wiele rozwiązań
end;
else
    x = -b / a; %jeśli a<>0 można wyznaczyć pierwiastek
end;
```

Przykładowe wywołania funkcji: `p = rown1(3,9); v = rown1(a1, a2);`

Jeśli funkcja zwraca więcej niż jedną wartość wywołanie ma postać: `[a, b, c] = nazwa_funkcji(d, e, f, g);`

Po wykonaniu instrukcji (wpisanych w oknie poleceń lub zawartych w wykonywanym skrypcie) utworzone w ich wyniku zmienne dostępne są cały czas w środowisku (do zamknięcia programu). Możemy podejrzeć ich wartości wpisując ich nazwy. Zmienne występujące w ciele funkcji traktowane są jak zmienne lokalne; po wykonaniu funkcji znikają.

Skrypt uruchamiamy wpisując w oknie poleceń jego nazwę bez rozszerzenia.

Zadania

5. Przećwicz w środowisku Octave powyższe polecenia.
6. Zapoznaj się z operatorami relacji (`>`, `=`, `~=`, ...) i z operatorami logicznymi (`&&`, `||`). Podaj przykłady użycia wszystkich tych operatorów.
7. Podaj sensowne przykłady użycia pętli `for` i `while` oraz instrukcji warunkowej.
8. Napisz funkcję, która oblicza pierwiastki równania kwadratowego i rysuje parabolę w zakresie od `x1` do `x2` (`x1` i `x2` to pierwiastki). Podaj przykład jej użycia (wywołanie i wyniki).

Zawartość sprawozdania

Sprawozdania powinny być sporządzone według wzoru zamieszczonego na stronie i zawierać:

- A) Cel ćwiczenia.
- B) Treści i wyniki zadań 2, 3, 4, 6, 7 i 8.
- C) Wnioski.

Przykładowe zagadnienia i pytania zaliczeniowe

- Opisane powyżej instrukcje
- Treść sprawozdania

Do przygotowania na następne zajęcia

1. Zapoznać się z instrukcją do kolejnego ćwiczenia.
2. Zapoznać się z częścią teoretyczną do kolejnego ćwiczenia.
3. Wykonać zadania pomocnicze do kolejnego ćwiczenia.