Politechnika Wrocławska

Pakiety obliczeniowe

INP001029WL

rok akademicki 2021/22

semestr zimowy

Wykład 3

Karol Tarnowski

karol.tarnowski@pwr.edu.pl

L-1 p. 220

Politechnika Wrocławska

Plan prezentacji (1)

- Operatory relacji
- Operatory logiczne

- Instrukcja warunkowa
- Pętle

Politechnika Wrocławska

Plan prezentacji (2)

- Wektoryzacja kodu

- Funkcje
- Funkcje anonimowe

Politechnika Wrocławska

Operatory relacji

- Operatory relacji służą do ilościowego porównania operandów (argumentów)
- Wynikiem działania operatora relacji jest tablica wartości logicznych wskazująca, gdzie dana relacja jest spełniona

https://www.mathworks.com/help/matlab/matlab_prog/array-comparison-with-relational-operators.html

Politechnika Wrocławska

Operatory relacji

Operator	Odpowiadająca funkcja	Opis
<	lt	mniejsze (less than)
<=	le	mniejsze/równe (less than or equal to)
>	gt	większe (greater than)
>=	ge	większe/równe (greater than or equal to)
==	eq	równe (equal to)
~=	ne	różne (not equal to)

Politechnika Wrocławska

Operatory relacji

- Operatory `>`, `<`, `>=`, oraz `<=` używają części rzeczywistych liczb zespolonych
- Operatory `==` oraz `~=` używaj części rzeczywistych oraz urojonych liczb zespolonych
- Wartości **`Inf`** są sobie równe
- Wartości **`NaN`** są różne od wszystkich liczb (włączając w to wartości **`NaN`**)

Politechnika Wrocławska

Operatory logiczne

operator/funkcja	działanie
`&`	koniunkcja
`~`	negacja
`|`	alternatywa
`xor`	alternatywa rozłączna
`&&, ||`	operatory logiczne ze skróconym wartościowaniem
`all`	czy wszystkie wartości są niezerowe
`any`	czy jakiś element jest niezerowy
`false`	fałsz (także tablica)
`find`	odnalezienie wartości niezerowych (także indeksów)
`islogical`	sprawdzenie typu danych
`logical`	konwersja wartości liczbowych na logiczne
`true`	prawda (także tablica)

Operatory relacji i logiczne

Przykłady

```
% na podstawie
% https://www.mathworks.com/help/matlab/matlab_prog/find-array-elements-that-meet-a-condition.html
clear all

rng default      % reset generatora liczb losowych
A = randi(21,4)  % utworzenie tablicy przykładowych danych
B = randi(21,4)  % utworzenie tablicy przykładowych danych

C = A > 11       % utworzenie tablicy logicznej
% z wykorzystaniem operatora relacji

D = A > B        % utworzenie tablicy logicznej
% z wykorzystaniem operatora relacji

% indeksowanie z wykorzystaniem wartości logicznych
A(C)
B(D)
```

Operatory relacji i logiczne

Przykłady

```
%%
% bardziej złożone wyrażenie logiczne
% wykorzystanie operatora logicznego
A(A<11 & A>2)

% to nie zadziała
A(2<A<11)
```

```
%%
% użycie innych operatorów logicznych
xor(C,D)
all(C)
any(C)
```

Operatory relacji i logiczne

Przykłady

```
%%
% wymiary macierzy muszą być odpowiednie
y = randi(5,5,1)
x = [0:2:6]

y < x
```

```
%%
% wykorzystanie funkcji find do zamiany indeksowania
% logicznego na indeksowanie liniowe
find(D)
[r, c] = find(D)
[r, c, v] = find(D)
```

```
%%
% konwersja danych liczbowych na logiczne
logical(A)
```

Politechnika Wrocławska

Instrukcja warunkowa

- Ogólna postać instrukcji warunkowej

```
if wyrażenie
    polecenia
elseif wyrażenie
    polecenia
else
    polecenia
end
```

Politechnika Wrocławska

Instrukcja warunkowa

- ***`wyrażenie`*** jest uznawane za prawdziwe, gdy jest niepuste, i zawiera tylko niezerowe elementy (logiczne lub liczby rzeczywiste)
- bloki **`else`**, **`elseif`** są opcjonalne

Politechnika Wrocławska

Instrukcja warunkowa

Przykłady

```
%% przykłady użycia instrukcji warunkowych
% na podstawie
% https://www.mathworks.com/help/matlab/ref/if.html

%% proste porównanie
if 1 > 0
    disp('1 > 0')
end

%% porównanie wektora liczb

limit = 0.75;
A = rand(5,1)

if any(A > limit)
    disp('Co najmniej jedna wartość przekracza granicę.')
else
    disp('Żadna wartość nie przekracza granicy.')
end
```

Instrukcja warunkowa

Przykłady

```
%% warunek złożony

x = 10;
minVal = 2;
maxVal = 6;

if (x >= minVal) && (x <= maxVal)
    disp('Wartość mieści się w zakresie.')
elseif (x > maxVal)
    disp('Wartość przekracza górną granicę.')
else
    disp('Wartość jest poniżej dolnej granicy.')
end
```

Politechnika Wrocławska

Pętla for

- Ogólna postać pętli for

```
for element = wartości
    polecenia
end
```

Politechnika Wrocławska

Pętla `for`

- Najczęściej ***`wartości`*** są wektorem zapisanym następująco:

`wartość_początkowa : wartość_końcowa`

- Może to być również wektor elementów lub tablica

Pętla for

Przykłady

```
%% przykłady użycia pętli for
% na podstawie
% www.mathworks.com/help/matlab/ref/for.html

%% zagnieżdżone pętle for po prostych wektorach
s = 6;
H = zeros(s);

for c = 1:s
    for r = 1:s
        H(r,c) = 1/(r+c-1);
    end
end

disp(H)
```

Pętla for

Przykłady

```
%% tworzenie macierzy z wykorzystaniem
% instrukcji warunkowej
ncols = 5
nrows = 6
for c = 1:ncols
    for r = 1:nrows
        if r == c
            A(r,c) = 2;
        elseif abs(r-c) == 1
            A(r,c) = -1;
        else
            A(r,c) = 0;
        end
    end
end
disp(A)
```

Politechnika Wrocławska

Pętla `for`

Przykłady

```
%% wektor elementów o niestandardowym kroku

for v = 1.0:-0.2:0.0
   disp(v)
end
```

```
%% wektor dowolnych elementów
values = rand(1,5)

for x = values
    disp(x)
end
```

```
%% tablica dowolnych elementów
values = rand(2,5)

for x = values
    disp('kolumna z macierzy values')
    disp(x)
end
```

Politechnika Wrocławska

Pętla `while`

- Ogólna postać pętli **`while`**

```
while wyrażenie
    polecenia
end
```

Pętla `while`

Przykład

```
%% przykład użycia pętli while
% na podstawie
% https://www.mathworks.com/help/matlab/ref/while.html

%% obliczanie wartości silni z wykorzystaniem
% pętli while

n = 10;
f = n;
while n > 1
    n = n-1;
    f = f*n;
end
disp(['n! = ' num2str(f)])

%% istnieje również funkcja factorial
disp(['n! = ' num2str(factorial(10))])
```

Politechnika Wrocławska

Polecenia `break, continue`

- Polecenie **`break`** służy do przerywania pętli **`while`** oraz **`for`**
- Polecenie **`continue`** służy do omijania iteracji w pętlach **`while`** oraz **`for`**

Polecenia `break`, `continue`

Przykłady

```
%% przykłady użycia poleceń break oraz continue
% na podstawie
% https://www.mathworks.com/help/matlab/ref/break.html
% https://www.mathworks.com/help/matlab/ref/continue.html

%% polecenie break przerywające pętle while
limit = 0.8;
s = 0;

while 1
    tmp = rand() % tmp = rand;
    if tmp > limit
        break
    end
    s = s + tmp
end
```

Polecenia `break`, `continue`

Przykłady

```
%% polecenie continue wykorzystane w pętli for
for n = 1:50
    if mod(n,7)
        continue
    end
    disp(['Podzielne przez 7: ' num2str(n)])
end
```

Politechnika Wrocławska

Wektoryzacja kodu

- Wektoryzacją kodu nazywamy proces zapisywania algorytmów z wykorzystaniem operacji wektorowych i macierzowych
- Zalety kodu wektorowego:
  - czytelność kodu
  - mniejsza podatność na błędy
  - szybkość działania (na ogół kod wektorowy jest szybszy)

Wektoryzacja kodu

Przykłady

```
%% porównanie kodu wykorzystującego pętle i kodu wektorowego
% na podstawie
% https://www.mathworks.com/help/matlab/matlab_prog/vectorization.html

%% wersja z pętlą
i = 0;
for t = 0:.01:10
    i = i + 1;
    y(i) = sin(t);
end

% inny sposób z pętlą
t = 0:.01:10
for i = 1:length(t)
    y(i) = sin(t(i));
end

% równoważny kod w wersji wektorowej
t = 0:.01:10;
y = sin(t);
```

Wektoryzacja kodu

Przykłady

```
%% obliczanie sum częściowych na co piątym elemencie
% wersja z pętlą
x = 1:10000;
tic
ylength = (length(x) - mod(length(x),5))/5;
y1(1:ylength) = 0;
for n= 5:5:length(x)
    y1(n/5) = sum(x(1:n));
end
toc

% wersja wektorowa
x = 1:10000;
tic
xsums = cumsum(x);
y2 = xsums(5:5:length(x));
toc

% sprawdzenie, że wyniki są tożsame
all(y1 == y2)
```

Wektoryzacja kodu

Przykłady

```
%% do wektoryzacji kodu przydatne są operatory macierzowe

% obliczenie objętości stożka
H = 1 % wysokość
D = 1 % średnica podstawy
V = 1/12*pi*(D^2)*H;

% pętla dla 10000 stożków
N = 10000
H = rand(1,N)
D = rand(1,N)

for n = 1:N
   V1(n) = 1/12*pi*(D(n)^2)*H(n);
end

% powyższa pętla zastąpiona zapisem wektorowym
% wykorzystującym operator macierzowy potęgowania .^
V2 = 1/12*pi*(D.^2).*H;

% sprawdzenie, że wyniki są tożsame
all(V1 == V2)
```

Politechnika Wrocławska

Wektoryzacja kodu

Przykłady

```
%% operatory macierzowe sprawdzają się dla danych w różnych wymiarach
s = 6;
H = zeros(s);

for c = 1:s
    for r = 1:s
        H(r,c) = 1/(r+c-1);
    end
end
disp(H)

c = [1:s]
r = [1:s]'
H(r,c) = 1./(r+c-1)
disp(H)
```

```
%% inny przykład
x = (-2:0.2:2)'; % 21 na 1 (kolumna)
y = -1.5:0.2:1.5; % 1 na 16 (wektor)
F = x.*exp(-x.^2-y.^2); % 21 na 16 (macierz)
```

Politechnika Wrocławska

Wektoryzacja kodu

Przykłady

```
%% obliczanie silni z wykorzystaniem pętli
% oraz operacji wektorowych

n = 10;
f = n;
while n > 1
    n = n-1;
    f = f*n;
end
disp(['n! = ' num2str(f)])

n = 10
prod(1:n)    % iloczyn liczb od 1 do n
cumprod(1:n) % ciąg iloczynów częściowych liczb od 1 do n
```

Politechnika Wrocławska

Skrypty

- Najprostszym typem programu Matlaba jest skrypt
- Skrypt zawiera polecenia i wywołania funkcji wykonywane przy uruchomieniu program
- Przy uruchomieniu skryptu dostępne są zmienne zapisane w przestrzeni roboczej
- Skrypt może modyfikować zawartość przestrzeni roboczej

https://www.mathworks.com/help/matlab/learn_matlab/scripts.html

Politechnika Wrocławska

Funkcje

- Deklaracja funkcji ma postać:

```
function [y1,...,yN] = myfun(x1,...,xM)
```

- ***myfun*** jest nazwą funkcji
- ***x1,...,xM*** to argumenty funkcji
- ***y1,...,yN*** wartości zwracane przez funkcję

https://www.mathworks.com/help/matlab/ref/function.html

Politechnika Wrocławska

Funkcje

- Funkcja może być umieszczona w pliku zawierającym tylko definicje funkcji (nazwa pliku zgodna z nazwą pierwszej funkcji)
- Funkcja może być umieszczona na końcu skryptu (od wersji R2016b, skrypt nie może się nazywać tak jak funkcja)
- Definicja funkcji najczęściej kończy się słowem kluczowym **`end`**

https://www.mathworks.com/help/matlab/ref/function.html

Funkcje

Przykłady

average.m | runnig_function.m | stat.m | rur

```
%% przykład definicji funkcji w pliku
function ave = average(x)
    ave = sum(x(:))/numel(x);
end
```

average.m | stat.m | runnig_function.m | runnig_fur

```
%% przykład funkcji zwracającej dwie wartości
function [m,s] = stat(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

verage.m | stat.m | runnig_function.m | runnig_functi

```
%% wywołanie funkcji zdefiniowanej w pliku

z = 1:99;
ave = average(z)
[m, s] = stat(z)
```

https://www.mathworks.com/help/matlab/ref/function.html

Funkcje

Przykłady

stat.m | runnig_function.m | runnig_function_2.m

```
%% wywołanie funkcji zdefiniowanych w skrypcie
z = 1:99;
ave = average2(z)
[m, s] = stat2(z)
```

```
%% przykład definicji funkcji w skrypcie
function ave = average2(x)
    ave = sum(x(:))/numel(x);
end

function [m,s] = stat2(x)
    n = length(x);
    m = sum(x)/n;
    s = sqrt(sum((x-m).^2/n));
end
```

https://www.mathworks.com/help/matlab/ref/function.html

Funkcje anonimowe

- Funkcja anonimowa nie jest przechowywana w pliku programu
- Jest przypisana do zmiennej typu **`function_handle`**
- Funkcje anonimowe mogą przyjmować wiele argumentów, a zwracają jedno wyjście
- Mogą zawierać tylko pojedyncze polecenia (ale można je zagnieżdżać)

https://www.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html

Politechnika Wrocławska

Funkcje anonimowe

- Przykładowa definicja funkcji anonimowej

```
sqr = @(x) x.^2
```

- `@` jest operatorem uchwytu do funkcji
- `(x)` nawiasy zawierają listę argumentów
- `x.^2` jest poleceniem
- funkcja jest przypisywana do zmiennej `sqr`

https://www.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html

Politechnika Wrocławska

Funkcje anonimowe

- Istnieje wiele funkcji Matlaba, do których można przekazać funkcję jako argument
  - `fplot`
  - `fzero`
  - `...`

https://www.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html

Politechnika Wrocławska

Podsumowanie

- Operatory relacji i logiczne
- Instrukcja warunkowa
- Pętle
- Wektoryzacja kodu
- Funkcje i funkcje anonimowe