INFORMATYKA

MPDI2

Wykład 8 MATLAB

Wprowadzenie

Instrukcja przypisania

Instrukcje wejścia/wyjścia

Obliczenia - standardowe funkcje arytmetyczne

Operacje logiczne

M-pliki

Instrukcja warunkowa

Literatura

1. Mrozek B., Mrozek Z.: Matlab 5.x – poradnik użytkownika, Wyd. PLJ, Warszawa, 1998.
2. Pratap R.: Matlab 7 dla naukowców i inżynierów, Wyd. PWN, 2007.
3. Kamińska A., Pańczyk B.: Ćwiczenia z Matlab. Przykłady i zadania, Wyd. Mikom., 2002
4. Bajorek T. :MATLAB, Podstawy użytkowania z przykładami. Materiały dydaktyczne, Wyd. PRz 2020

Wersja Matlaba w do pobrania

- R2023a i starsze (**do instalacji dla studentów wersja licencjonowana przez Politechnikę Rzeszowską**)

  matlab.prz.edu.pl

… instrukcja instalacji i aktywacji po zalogowaniu w serwisie

matlab.prz.edu.pl/instalacja-i-aktywacja-oprogramowania

Konieczna rejestracja w serwisie *mathworks.com* – konto studenckie

Można też korzystac z Matlaba w wersji ***online****:*

matlab.mathworks.com

(po zalogowaniu na konto Mathworks)

Pliki programów są w chmurze:

drive.matlab.com

Zamiast Matlaba można też wykorzystać:

FreeMat

FreeMat jest to darmowe (open source, GNU GPL) numeryczne, macierzowo zorientowane środowisko obliczeniowe i język programowania. FreeMat obsługuje wiele funkcji MATLAB'a

Strona domowa FreeMat: http://freemat.sourceforge.net

Scilab **http://www.scilab.org/**

Cytat ze strony...

Scilab is free and open source software

Maths & Simulation

2-D & 3-D Visualization

Optimization

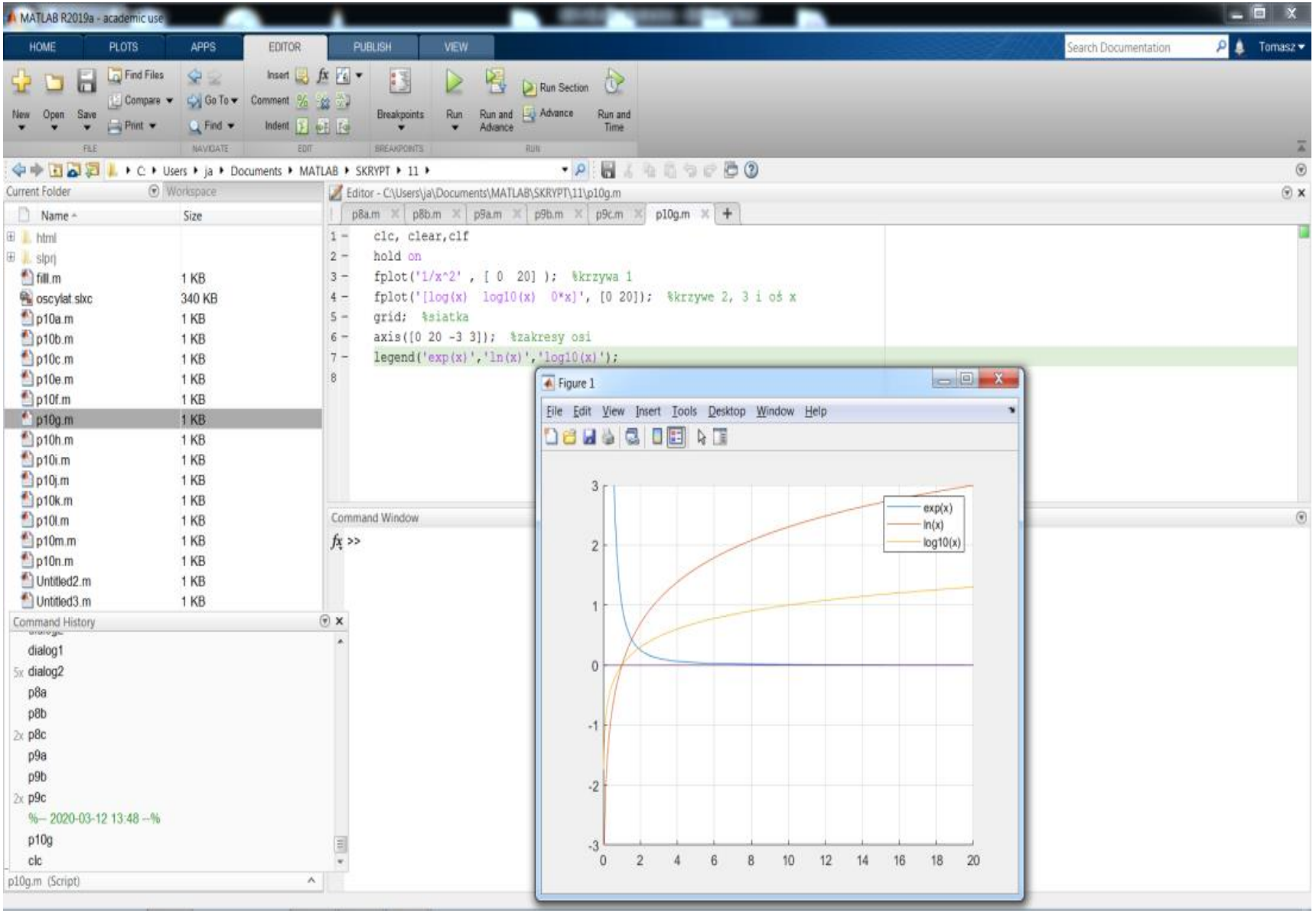
Statistics

Control System Design & Analysis

Środowisko ***Matlaba*** składa się z:

- okna ***Command Window***, w którym wpisujemy pojedyncze polecenia i widzimy ich rezultaty oraz rezultaty wykonywanego m-pliku
- okna ***Command History*** - historia poleceń,
- okna z zakładkami:
  - ***Workspace*** - obszar roboczy - lista zainicjowanych przez użytkownika zmiennych i ich wartości,
  - ***Current Folder*** - zawartość katalogu roboczego.
  - opcjonalnego okna edycji m-plik-ów

Okna aplikacji Matlab



Praca w środowisku odbywa się sposobami:

- **interakcyjnym**
- **wsadowym**

Sposób interakcyjny – interpretacja pojedynczych poleceń pisanych w oknie **Command Window,** (wpisujemy pojedyncze instrukcje obliczeniowe i na bieżąco otrzymujemy wyniki)

Instrukcje wpisywane przez użytkownika, każda instrukcja wykonywana

(i potwierdzana ENTER-em) na bieżąco

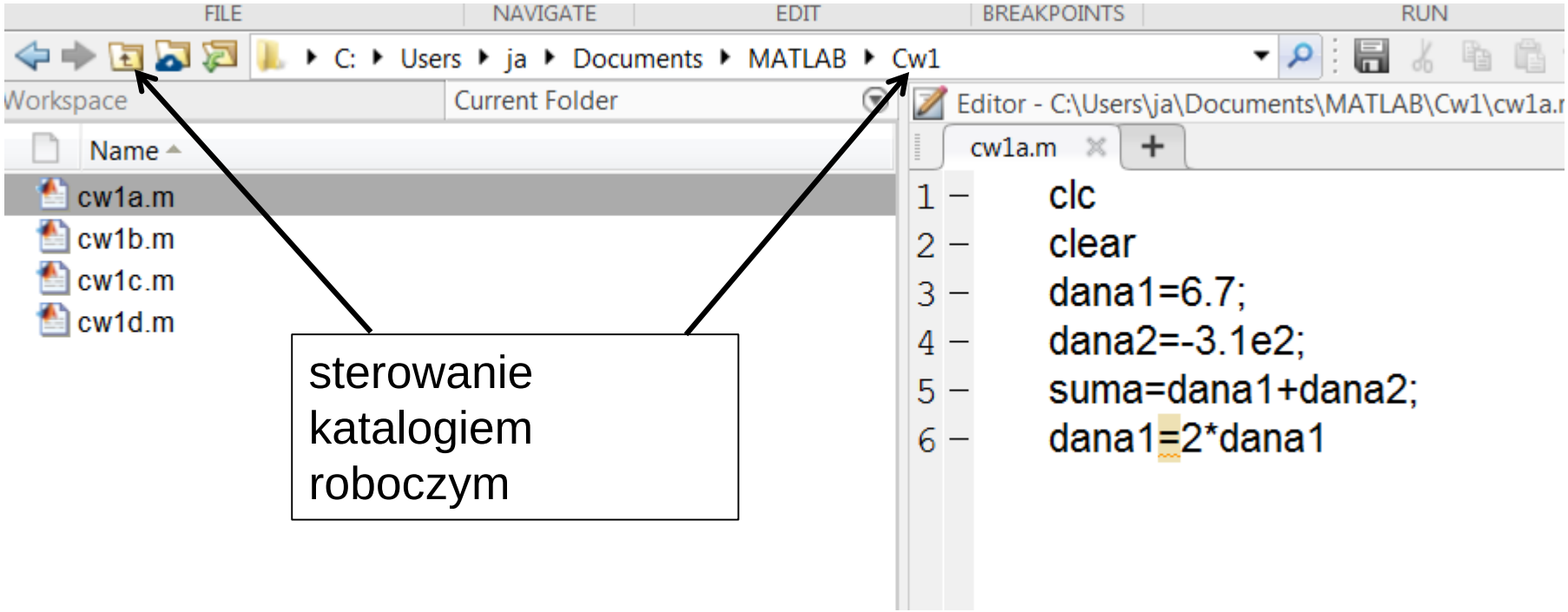
Dialog:

>> *tu wpisujemy instrukcję wykonawczą (ENTER)*

odpowiedź

Sposób wsadowy

można tworzyć pliki (tzw. m-pliki) z wieloma instrukcjami Matlab'a i je wykonywać globalnie – wykonanie sekwencyjne



Większość instrukcji to tzw. instrukcje przypisania (nadania wartości **zmiennym**)

Zmienna służy do identyfikacji wartości określonego typu (liczba, tekst, tablica itp.) - jest tworzona w pamięci w momencie nadania jej wartości

Tak jak w *JavaScript*

Nazwa (identyfikator) zmiennej musi zaczynać się od **litery**, dalszy ciąg litery i cyfry – **bez spacji**). Nie używamy polskich "ogonków" :ą ę ś ć itd.

W nazwach zmiennych (funkcji) **istotne duże i małe litery**!

Zmienna a i A to dwie różne
zmienne, mogą mieć różne wartości

Wartości (stałe, wyrażenia obliczeniowe) można przypisywać zmiennym, nadawać im wartości

Ogólna postać instrukcji przypisania (inicjacja zmiennej i nadawanie wartości)

zmienna = wyrażenie

oblicz wyrażenie i wynik wyrażenia przechowaj w zmiennej

wyrażenie

wynik wyrażenia przechowany w domyślnej zmiennej ans

Zainicjowane zmienne i ich wartości widoczne są w oknie ***Workspace***

Oprócz tego efektem instrukcji przypisania jest echo ekranowe (wartość zmiennej jest wypisywana w oknie ***Command Window***), chyba, że instrukcję przypisania zakończymy średnikiem (**;**) – wówczas nie ma echa ekranowego

Zmienną o wcześniej zdefiniowanej wartości można użyć w **wyrażeniu** przypisywanym nowej zmiennej – nie wolno używać zmiennej niezdefiniowanej

```
>> x=2*c

??? Undefined function or variable 'c'.
```

Wartość zmiennej można przedefiniować:

```
>> x=4.5

        x= 4.5000

>> x = 5.5

        x= 5.5000                          Nowa wartość x

>> x= 2*x

        x=11.0000                          Podwojenie wartości x
```

Typy zmiennych

Utworzone programem zmienne i ich typy obserwujemy w oknie **Workspace**

Większość zmiennych w programach obliczeniowych będzie typu **liczbowego** (typ **double**)

x1 = 12

x2 = -1.78

x3 = 1.3e-8

x4 = sin(pi/7)

Typ zmiennej zależy od typu przypisanego tej zmiennej **wyrażenia**

Istnieją też zmienne innych typów:

napis = "Politechnika" typ *string*

nazwisko = 'Kowalski' typ *char* (tablica znaków)

czy = 56>12 typ *logical*

M = [1.5 2.1 3.4] tablica (typu *double*)

W = { 1 "Nowak" [2 12 2000] } typ *cell*

i inne

UWAGI:

- **Wyrażenia budujemy podobnie jak w Excelu i JavaScript - stałe, zmienne, operatory działań, funkcje, nawiasy okrągłe ()**
- **Separator liczb dziesiętnych – kropka!!!!!**
- **Przywołanie poprzednich instrukcji w celu ich ponownego wykonania bądź edycji (strzałki kursora – góra, dół) lub kliknięcie w oknie Command history**
- **Długa instrukcja – kontynuacja trzy kropki (...) i kontynuacja w nowym wierszu**

Operatory działań arytmetycznych podobnie jak w Excel-u.

^	potęgowanie !! (też pierwiastki)
-	zmiana znaku (przed liczbą lub zmienną)
* /	mnożenie, dzielenie
+ -	dodawanie, odejmowanie

Uwaga: Priorytet operatorów: potęgowanie wcześniejsze do zmiany znaku (w Excelu było odwrotnie)

Sprawdzić:

-2^2	w Matlabie	-4
=-2^2	w Excelu	4

Zmiana priorytetu operatorów: nawiasy okrągłe!

Nie ma operatorów składania ++ -- += -= itd. jak JavaScript

Podstawowe funkcje matematyczne:

pi – stała wbudowana

sin(w) cos(w) tan(w) cot(w) *(funkcje trygonometryczne–dla kąta w radianach!)*

sind(w) cosd(w) tand(w) cotd(w) *(funkcje trygonometryczne -dla kąta w stopniach)*

log(w)	***logarytm naturalny,***
log10(w)	***logarytm dziesiętny***
exp(w)	***funkcja wykładnicza*** e^x
sqrt(w)	***pierwiastek kwadratowy***
abs(w)	***wartość bezwzględna***
fix(w)	***zaokrąglenie do całkowitej (w kierunku zera)***
floor(w)	***zaokrąglenie do całkowitej w kierunku -∞***
ceil(w)	***zaokrąglenie do całkowitej w kierunku +∞***
round(w)	***zaokrąglenie do najbliższej całkowitej***
rand	***bezargumentowa funkcja - generator liczby losowej dziesiętnej z przedziału (0, 1)***
rem(a,b)	***reszta z dzielenia a/b***
power(a,b)	a^b **alternatywnie do operatora ^**

Przykłady:

Jeśli znane a, b c, x , y to wyrażenia zapisujemy:

Matematyka	Matlab
abc	a*b*c
$a+\frac{b}{c}$	a+b/c
$\frac{a+b}{c}$	(a+b)/c
$\frac{sin30^o}{a+b}$	sind(30)/(a+b)
$\frac{e^{x+2}+y}{1+\ln c}$	(exp(x+2)+y)/(1+log(c))
$\frac{\sqrt[3]{sin^2x+cos^2x}}{log_{10}y}$	power(sin(x)^2+cos(x)^2,1/3)/log10(y)

Przykład

Matematyka

$$\frac{3sin^3x - 2ln\,x}{3 - \sqrt[3]{x^3 + 3}}$$

Zapis w *Matlabie*:

```
x= pi/6 %radiany
y= (3*sin(x)^3-2*log(x))/(3-(x^3+3)^(1/3))
```

lub wykorzystanie funkcji power:

```
y= (power(3*sin(x),3)-2*log(x))/(3-power(x*x*x+3,1/3))
```

Przydatne polecenia

help - pomoc globalna

help elfun - pomoc – spis funkcji elementarnych

help *rem* - pomoc na temat wybranej funkcji (tu: *rem*)

format long - wyświetlanie liczb z 15-ma miejscami dziesiętnymi

format short - liczby wyświetlane z 4-ma miejscami dziesiętnymi

clc - czyszczenie ekranu Command Window

clear zmienna - usunięcie zmiennej z obszaru roboczego (*Workspace*)

clear - usunięcie wszystkich zmiennych z obszaru roboczego

Algorytm zaokrąglania z dokładnością do N miejsc dziesiętnych

```
clc
format long
x=pi
N=5
y= round(x*10^N)/10^N
format short
```

albo

```
clc
format long
x=pi
N=5
y= round(pi, N)
format short
```

Generator losowy rand

Jeśli bezargumentowa funkcja - losuje liczbę rzeczywistą z przedziału (0, 1)

```
x= rand()

x= rand
```

Zmiana przedziału losowania: rand*szerokość+przesunięcie

```
x=round(100*rand-50)
```

Losowanie liczby całkowitej z przedziału (-50, 50)

Uwaga: jeśli podamy argument (lub argumenty) funkcji **rand**

x=rand(5) - losowanie tablicy kwadratowej 5x5

x=rand(5,10) - losowanie tablicy 5x10

Zmienne zespolone

Zmienna zespolona ma postać:

$$a + b\,i$$

gdzie:

a – liczba będąca tzw. częścią rzeczywistą (łac. pars realis),

b – liczba będąca tzw. częścią urojoną (łac. pars imaginaria),

i – jednostka urojona wg definicji: $i^2 = -1$

Wyniki zespolone otrzymamy dla przykładowych działań:

```
>> a= sqrt(-4)
a =
      0 + 2.0000i
>>b= log(-3)
b =
      1.0986 + 3.1416i
```

Zobaczymy potem, że rozwiązania niektórych równań mogą dać wartości zespolone, np. równania kwadratowego o ujemnej wartości ***delta***

Prezentacja wyników w Matlabie - instrukcje wyjścia

Mamy poniższe możliwości:

- jeśli instrukcja przypisania nie kończy się średnikiem) – pojawia się echo instrukcji wypisujące nadaną zmiennej wartość

zmienna - *proste wypisanie wartości*

disp(*zmienna*) - *funkcja wypisania wartości*

disp ('jakiś tekst') - *wypisanie tekstu na ekranie*

NOWOŚĆ!!!

Funkcja fprintf - tekst i dane

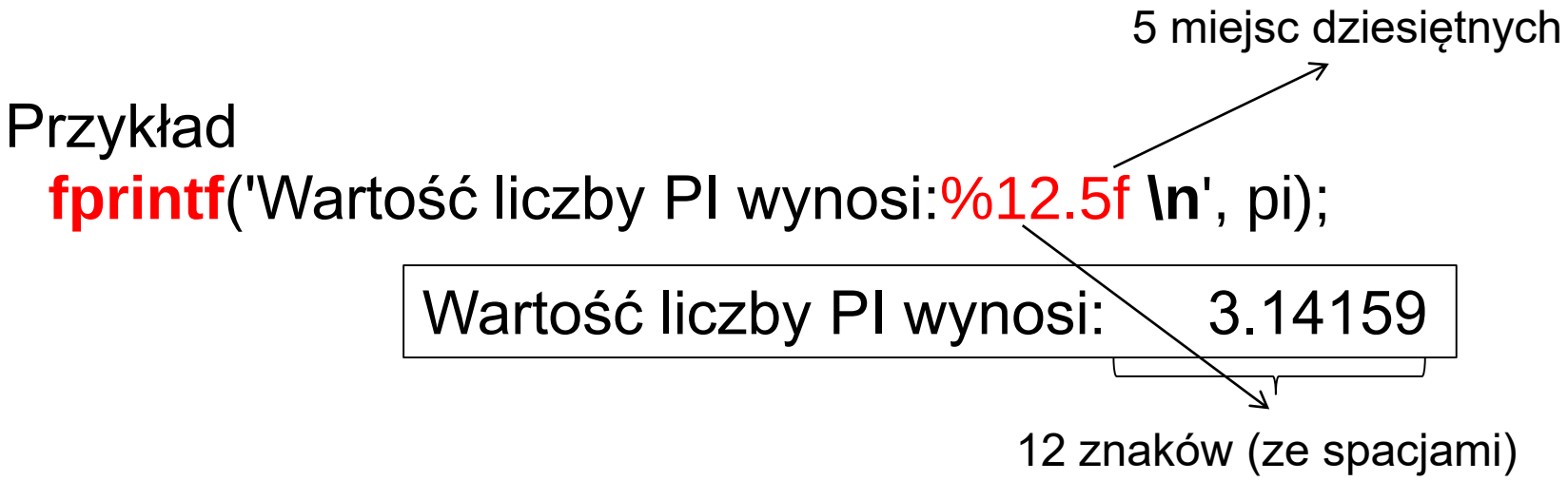
fprintf('Wartość zmiennej a wynosi %x **\n**', a);

gdzie:

Ciąg znaków %x	**Sposób wyświetlania**
%**d**	liczba całkowita
%**e**	liczba w zapisie zmiennoprzecinkowym, np. 3.1415e+02
%**f**	zapis stałoprzecinkowy

Uwaga: **\n** przejście do nowego wiersza

Można też ustalać szerokość pola dla liczby i liczbę miejsc dziesiętnych




```
fprintf('Wartość liczby PI wynosi:%0.6f \n', pi);
```

```
Wartość liczby PI wynosi:3.141593
```

M-pliki

W *Matlab*-ie można zapisać tekst ciągu instrukcji w pliku tekstowym ASCII o rozszerzeniu **m**

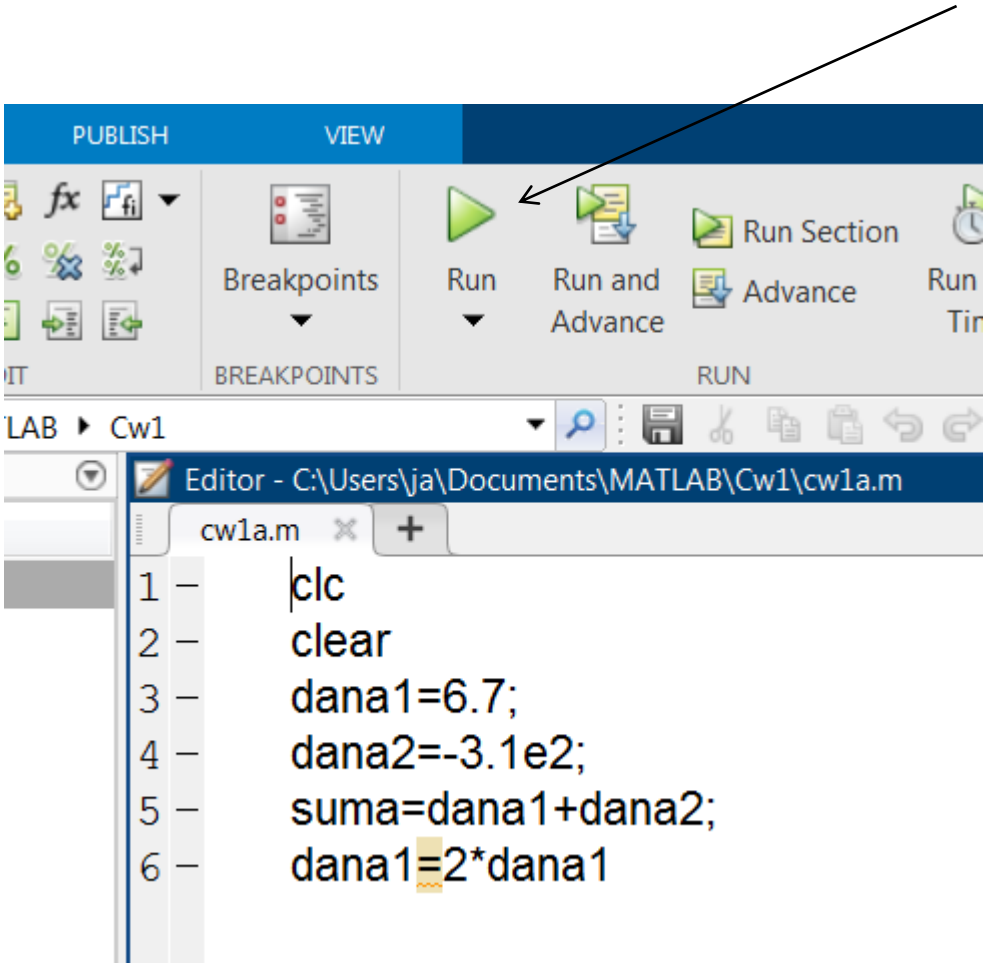
nazwa.m (tzw. m-pliki),

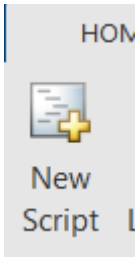
a następnie wykonać plik - **instrukcje wykonywane są kolejno, jedna po drugiej od pierwszej do ostatniej (jeśli program jest bez rozwidleń)**.

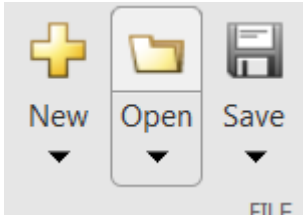
Matlab zawiera własny edytor ASCII.

Okno edytora m-plików

wykonanie m-pliku (lub klawisz F5)







Uwagi praktyczne

Efekty wykonania m-pliku widoczne *Command Window*

Mogą być wykonywane tylko m-pliki z **katalogu roboczego** (widoczne w oknie *Current Folder*) — chociaż istnieje też możliwość dodania wielu katalogów do listy katalogów roboczych

Polecenia w m-pliku piszemy dla czytelności w osobnych wierszach, choć można w jednym wierszu, oddzielając je przecinkiem lub średnikiem

Jeśli **średnik** umieścimy na końcu polecenia to **brak echa** instrukcji na ekranie *Command Window*

Po znaku **%** piszemy komentarze – ignorowane przez *Matlaba*

Uwaga na błędy instrukcji!

Rola średnika (brak echa)

m-plik

```
clc, clear
c=4
b=5;
```

Command Window

```
c =
     4
```

Przykładowy tekst w m-pliku:

```
clc, clear
a=4;
b= -3.4;
c = 2.45;
delta= b*b-4*a*c;
disp('Współczynniki równania kwadratowego:');
fprintf('a: %f b: %f c: %f \n', a, b, c);
fprintf('Wartość delta wynosi %f \n', delta);
```

Efekt wykonania m-pliku w *Command Window*

```
Współczynniki równania kwadratowego:

a: 4.000000 b: -3.400000 c: 2.450000

Wartość delta wynosi -27.640000

>>
```

Interakcja z użytkownikiem (instrukcja wejścia – możliwość wprowadzania danych w trakcie biegu programu)

```
zmienna = input('tekst zachęty');
```

Działanie: zatrzymanie programu i oczekiwanie na podanie wartości dla zmiennej przez użytkownika

```
a = input('Podaj a:');
b = input('Podaj b:');
c = input('Podaj c:');
```

Stałe (wykorzystywane w wyrażeniach)

Liczbowe, np.: 3 -3.4 2.3e6

Tekstowe (napisy):

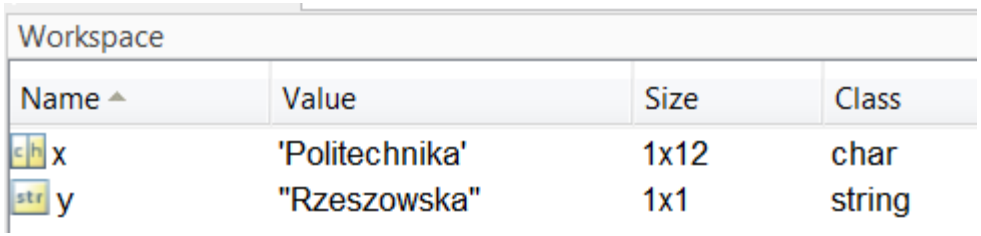
'Politechnika' - to jest **wektor znaków** - array of char (ang. *character* – znak) – tu wektor ma 12 komórek

równoważny zapis: x=['P', 'o', 'l', 'i', 't', 'e', 'c', 'h', 'n', 'i', 'k', 'a']

"Rzeszowska" - to jest zmienna typu string (łańcuch znaków)

x='Politechnika'

y="Rzeszowska"



Logiczne: 0 1

Instrukcja warunkowa

Instrukcja służy do **sprawdzenia warunków** i alternatywnego wykonywania różnych grup instrukcji gdy dany warunek będzie prawdziwy (true)

Postać ogólna

```
if warunek1
  instrukcje  (wykonywane gdy jest spełniony warunek1)
elseif warunek2
  instrukcje (wykonywane gdy jest spełniony warunek2)
elseif warunek3
 instrukcje (wykonywane gdy jest spełniony warunek3)
…itd
else
  instrukcje (wykonywane gdy niespełnione oba warunki)
end
```

Uwaga1: Bloki else i elseif mogą zostać pominięte.

Uwaga2: Gdy kolejny warunek jest prawdziwy, pozostałe warunki nie są już sprawdzane

Uwaga3: Instrukcja zawsze kończy się słowem kluczowym ***end***

Warunek (relacja) to połączenie dwóch wyrażeń arytmetycznych **operatorami relacji**:

> < >= <= == (równe)

~= (nierówne – inaczej jak w JavaScript)

Przykłady warunków:

a == 0 (równe UWAGA: dwa znaki =)
b<c
2*a >= 5
x ~= 5 (różne od 5)

Warunki przyjmują wartość typu logicznego:

0 – fałsz

1 – prawda

Operatory logiczne

Dwa wartości logiczne można związać operatorami logicznymi:

&	koniunkcja warunków
|	alternatywa warunków
	lub jak w *JavaScript* && i ||

lub zanegować wartość logiczną:

~	negacja (inaczej jak w JavaScript)

Przykład

```
x=6
czy=x>0&x<10
```

```
czy
        1
```

Przykład 1

```
a = 1;
b = 6;
c = 3;
delta =  b^2-4*a*c;
if delta<0
    disp ('delta jest ujemne')        % wyświetlenie tekstu1
elseif delta==0
    disp('delta równe 0')              % wyświetlenie tekstu2
else
   disp('delta większe od 0')         % wyświetlenie tekstu3
end;
```

Przykład

Test zawierania się liczby w określonym przedziale:

```
a = input('Podaj liczbę:');
if (a>6) && (a<10)
        disp('a w przedziale (6, 10)')
else
        disp('a poza przedziałem (6, 10)')

end
```