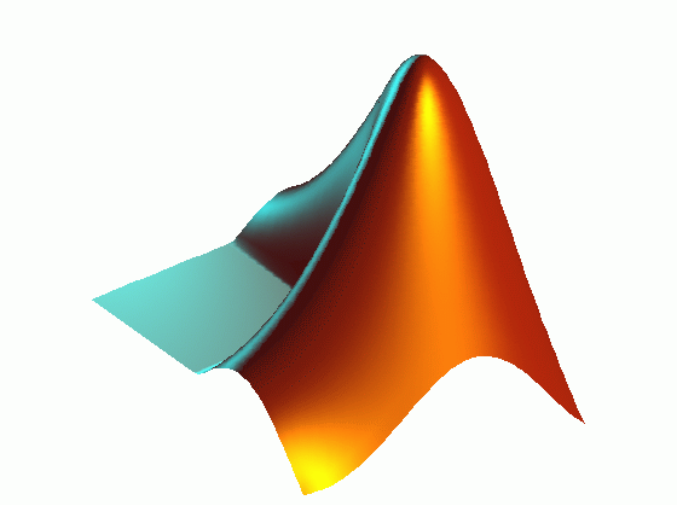


Matlab — MATrix LABoratory

Wprowadzenie

Bartosz Miller, Piotr Nazarko, Artur Borowiec

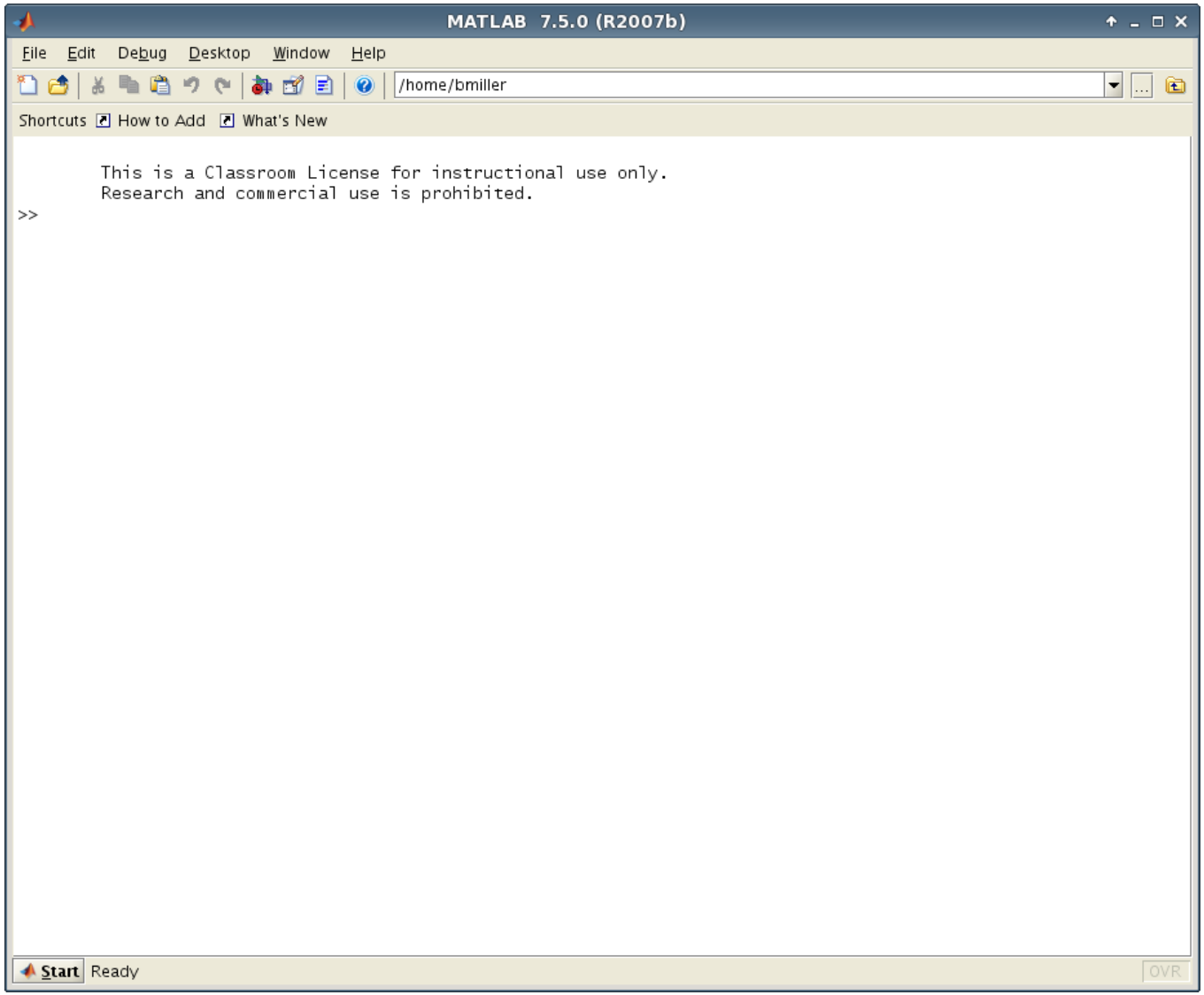
Katedra Mechaniki Konstrukcji
Politechnika Rzeszowska

18 listopada 2010

Matlab

Program komputerowy będący interaktywnym środowiskiem do wykonywania obliczeń naukowych i inżynierskich, oraz do tworzenia symulacji komputerowych. Zalety programu:

- duża liczba funkcji bibliotecznych,
- możliwość rozbudowy (tworzenie własnych funkcji),
- posiada swój język programowania, co umożliwia pisanie programów działających w środowisku Matlaba,
- rysowanie dwu i trójwymiarowych wykresów funkcji,
- wizualizacja wyników obliczeń w postaci rysunków statycznych i animacji.


MATLAB 7.5.0 (R2007b)
File Edit Debug Desktop Window Help
/home/bmiller
Shortcuts How to Add What's New
This is a Classroom License for instructional use only.
Research and commercial use is prohibited.
>>
Start Ready
OVR

```
   This is a Classroom License for instructional use only.
   Research and commercial use is prohibited.

>>
```

```
   This is a Classroom License for instructional use only.
   Research and commercial use is prohibited.

>>komenda [ENTER]
```

```
   This is a Classroom License for instructional use only.
   Research and commercial use is prohibited.

>>2+2.5 [ENTER]
```

```
   This is a Classroom License for instructional use only.
   Research and commercial use is prohibited.

>>2+2.5 [ENTER]
ans =
     4.5
>>
```

```
   This is a Classroom License for instructional use only.
   Research and commercial use is prohibited.

>>2+2.5[ENTER]
ans =
     4.5
>>2+2,5[ENTER]
```

```
   This is a Classroom License for instructional use only.
   Research and commercial use is prohibited.

>>2+2.5[ENTER]
ans =
     4.5
>>2+2,5[ENTER]
ans =
     4
ans =
     5
>>
```

Separatorem dziesiętnym jest w Matlabie wyłącznie kropka. Przecinek służy do oddzielania elementów (argumentów). W powyższym przykładzie otrzymane odpowiedzi to suma elementów przed przecinkiem i powtórzony element za przecinkiem.

Instrukcja przypisania

```
>>a=5
a =
      5
>>
```

Instrukcja przypisania

```
>>a=5
a =
      5
>>
```

Brak echa na ekranie. UWAGA: instrukcja jest wykonywana, brak tylko potwierdzenia!

```
>>b=0;
>>
```

Instrukcja przypisania

```
>>a=5
a =
      5
>>
```

Brak echa na ekranie. UWAGA: instrukcja jest wykonywana, brak tylko potwierdzenia!

```
>>b=0;
>>
```

Wyświetlenie wartości zmiennej

```
>>b
b =
      0
>>
```

Instrukcja przypisania

```
>>a=5
a =
     5
>>
```

Brak echa na ekranie. UWAGA: instrukcja jest wykonywana, brak tylko potwierdzenia!

```
>>b=0;
>>
```

Wyświetlenie wartości zmiennej

```
>>b
b =
     0
>>
```

Dzielenie

```
>>a/b
ans =
     Inf
>>
```

Podnoszenie do dowolnej potęgi

```
>>3^2
ans =
     9
>>
```

Podnoszenie do dowolnej potęgi

```
>>3^2
ans =
     9
>>9^0.5
ans =
     3
```

Pierwiastek `n`-tego stopnia uzyskujemy przez podniesienie do potęgi o wykładniku `1/n`.

Podnoszenie do dowolnej potęgi

```
>>3^2
ans =
     9
>>9^0.5
ans =
     3
```

Pierwiastek n-tego stopnia uzyskujemy przez podniesienie do potęgi o wykładniku 1/n.

Odwrotność

```
>>1/4
ans =
     0.25
>>
```

Podnoszenie do dowolnej potęgi

```
>>3^2
ans =
     9
>>9^0.5
ans =
     3
```

Pierwiastek n-tego stopnia uzyskujemy przez podniesienie do potęgi o wykładniku 1/n.

Odwrotność

```
>>1/4
ans =
     0.25
>>4^-1
ans =
     0.25
```

Odwrotność liczby można otrzymać podnosząc ją do potęgi o wykładniku -1.

Definiowanie macierzy 3x3

```
>>A = [1 2 3 ; 3 2 1 ; 0 1 2]
A =
     1   2   3
     3   2   1
     0   1   2
>>
```

Definiowanie macierzy 3x3

```
>>A = [1 2 3 ; 3 2 1 ; 0 1 2]
A =
     1   2   3
     3   2   1
     0   1   2
>>
```

Nawiasy kwadratowe [oraz] oznaczają początek i koniec macierzy, średniki ; oddzielają od siebie kolejne wiersze. Każdy wiersz musi zawierać tyle samo elementów rozdzielonych spacjami.

Definiowanie macierzy 3x3

```
>>A = [1 2 3 ; 3 2 1 ; 0 1 2]
A =
     1   2   3
     3   2   1
     0   1   2
>>
```

Nawiasy kwadratowe [oraz] oznaczają początek i koniec macierzy, średniki ; oddzielają od siebie kolejne wiersze. Każdy wiersz musi zawierać tyle samo elementów rozdzielonych spacjami.

Definiowanie macierzy 1x3

```
>>B = [2 3 4]
B =
     2   3   4
>>
```

Definiowanie macierzy 3x3

```
>>A = [1 2 3 ; 3 2 1 ; 0 1 2]
A =
     1   2   3
     3   2   1
     0   1   2
>>
```

Nawiasy kwadratowe [oraz] oznaczają początek i koniec macierzy, średniki ; oddzielają od siebie kolejne wiersze. Każdy wiersz musi zawierać tyle samo elementów rozdzielonych spacjami.

Definiowanie macierzy 1x3

```
>>B = [2 3 4]
B =
     2   3   4
>>
```

Definiowanie macierzy 3x1 (wektora 3-elementowego), średnik na końcu — brak echa

```
>>b = [2 ; 3 ; 4];
```

Program rozróżnia małe i wielkie litery, B oraz b to różne znaki.

Mnożenie macierzy

```
>>C=A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>>
```

Mnożenie macierzy

```
>>C=A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>>
```

Nie zawsze można mnożyć przez siebie macierze!

Mnożenie macierzy

```
>>C=A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>>
```

Nie zawsze można mnożyć przez siebie macierze!

Mnożenie macierzy

```
>>C=B*A
C =
     11    14    17
>>
```

Mnożenie macierzy

```
>>C=A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>>
```

Nie zawsze można mnożyć przez siebie macierze!

Mnożenie macierzy

```
>>C=B*A
C =
     11    14    17
>>
```

Jeżeli odpowiednie wymiary macierzy są zgodne mnożenie można wykonać.

Mnożenie macierzy

```
>>C=A*B
??? Error using ==> mtimes
Inner matrix dimensions must agree.
>>
```

Nie zawsze można mnożyć przez siebie macierze!

Mnożenie macierzy

```
>>C=B*A
C =
     11   14   17
>>
```

Jeżeli odpowiednie wymiary macierzy są zgodne mnożenie można wykonać.

Transponowanie macierzy

```
>>D=C'
D =
     11
     14
     17
>>
```

Wyświetlenie zdefiniowanych zmiennych i ich wymiarów

```
>>whos
Name Size Bytes Class Attributes
A    3x3   72    double
B    1x3   24    double
C    1x3   24    double
D    3x1   24    double
a    1x1   8     double
ans  1x1   8     double
b    1x1   8     double
```

Wyświetlenie zdefiniowanych zmiennych i ich wymiarów

```
>>whos
Name Size Bytes Class Attributes
A    3x3   72    double
B    1x3   24    double
C    1x3   24    double
D    3x1   24    double
a    1x1   8     double
ans  1x1   8     double
b    1x1   8     double
```

Wykasowanie zmiennej

```
>>clear A
>>whos
Name Size Bytes Class Attributes
B    1x3   24    double
C    1x3   24    double
D    3x1   24    double
a    1x1   8     double
ans  1x1   8     double
b    1x1   8     double
```

Wykasowanie wszystkich zmiennych

```
>>clear all
>>whos
>>
```

Wykasowanie wszystkich zmiennych

```
>>clear all
>>whos
>>
```

Definicja dwóch macierzy (bez echa na ekranie)

```
>>A = [1 2 3 ; 4 5 6 ; 7 8 9];
>>B = [11 12 13 ;14 15 16 ; 17 18 19];
```

Wykasowanie wszystkich zmiennych

```
>>clear all
>>whos
>>
```

Definicja dwóch macierzy (bez echa na ekranie)

```
>>A = [1 2 3 ; 4 5 6 ; 7 8 9];
>>B = [11 12 13 ;14 15 16 ; 17 18 19];
```

Dodawanie, odejmowanie i odwracanie macierzy (na ekranie pojawi się echo)

```
>>C=A+B
>>D=A-B
>>A^-1
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 1.541976e-18.
ans =
1.0e+16 *
-0.4504 0.9007 -0.4504
0.9007 -1.8014 0.9007
-0.4504 0.9007 -0.4504
```

Wyznacznik macierzy

```
>>A = [1 2 5; 1 0 5; 5 2 3];
>>det(A)
ans =
     44
>>
```

Wyznacznik macierzy

```
>>A = [1 2 5; 1 0 5; 5 2 3];
>>det(A)
ans =
     44
>>
```

Odwracać można wyłącznie macierze nieosobliwe, tzn. takie, których wyznacznik jest różny od zera. W przypadku macierzy A jej wyznacznik wynosi 44, więc istnieje macierz od niej odwrotna.

Odwracanie macierzy

```
>>A^-1
ans =
-0.2273 0.0909 0.2273
0.5000 -0.5000 0
0.0455 0.1818 -0.0455
>>
```

Wyznacznik macierzy

```
>>A = [1 2 5; 1 0 5; 5 2 3];
>>det(A)
ans =
     44
>>
```

Odwracać można wyłącznie macierze nieosobliwe, tzn. takie, których wyznacznik jest różny od zera. W przypadku macierzy A jej wyznacznik wynosi 44, więc istnieje macierz od niej odwrotna.

Odwracanie macierzy

```
>>A^-1
ans =
-0.2273 0.0909 0.2273
0.5000 -0.5000 0
0.0455 0.1818 -0.0455
>>
```

Z definicji A*A^-1 daje macierz jednostkową (1 na głównej przekątnej, poza nią 0)

```
>>A*A^-1
>>A^-1*A
```

Komentarz — ignorowana część komendy

```
>>Z=3+5; % wszystko, co znajduje się po znaku % jest ignorowane
>>
```

Komentarz — ignorowana część komendy

```
>>Z=3+5; % wszystko, co znajduje się po znaku % jest ignorowane
>>
```

Macierze specjalne — macierz wypełniona zerami

```
>>Z=zeros(12); % macierz kwadratowa 12 na 12
>>Z=zeros(12,10); % macierz 12 na 10, argumenty oddzielone przecinkiem
>>
```

Komentarz — ignorowana część komendy

```
>>Z=3+5; % wszystko, co znajduje się po znaku % jest ignorowane
>>
```

Macierze specjalne — macierz wypełniona zerami

```
>>Z=zeros(12); % macierz kwadratowa 12 na 12
>>Z=zeros(12,10); % macierz 12 na 10, argumenty oddzielone przecinkiem
>>
```

Macierze specjalne — macierz wypełniona jedynkami

```
>>J=ones(12); % macierz kwadratowa 12 na 12
>>J=ones(12,10); % macierz 12 na 10, argumenty oddzielone przecinkiem
>>
```

Komentarz — ignorowana część komendy

```
>>Z=3+5; % wszystko, co znajduje się po znaku % jest ignorowane
>>
```

Macierze specjalne — macierz wypełniona zerami

```
>>Z=zeros(12); % macierz kwadratowa 12 na 12
>>Z=zeros(12,10); % macierz 12 na 10, argumenty oddzielone przecinkiem
>>
```

Macierze specjalne — macierz wypełniona jedynkami

```
>>J=ones(12); % macierz kwadratowa 12 na 12
>>J=ones(12,10); % macierz 12 na 10, argumenty oddzielone przecinkiem
>>
```

Macierze specjalne — macierz jednostkowa (1 na głównej przekątnej, poza nią 0)

```
>>J=eye(12); % macierz kwadratowa 12 na 12
>>
```

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Argumentem funkcji `sum` (obliczenie sumy argumentów) są wszystkie elementy macierzy M znajdujące się w wierszach od 1 do 6 (`1:6`) oraz w kolumnie pierwszej (`1`). Numery wierszy i kolumn są rozdzielone przecinkiem.

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Argumentem funkcji sum (obliczenie sumy argumentów) są wszystkie elementy macierzy M znajdujące się w wierszach od 1 do 6 (1:6) oraz w kolumnie pierwszej (1). Numery wierszy i kolumn są rozdzielone przecinkiem.

To samo zadanie — suma elementów z pierwszej kolumny — można wykonać prościej

```
>>sum( M(1:end,1) )
```

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Argumentem funkcji sum (obliczenie sumy argumentów) są wszystkie elementy macierzy M znajdujące się w wierszach od 1 do 6 (1:6) oraz w kolumnie pierwszej (1). Numery wierszy i kolumn są rozdzielone przecinkiem.

To samo zadanie — suma elementów z pierwszej kolumny — można wykonać prościej

```
>>sum( M(1:end,1) )
>>sum( M(:,1) )
```

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Argumentem funkcji sum (obliczenie sumy argumentów) są wszystkie elementy macierzy M znajdujące się w wierszach od 1 do 6 (1:6) oraz w kolumnie pierwszej (1). Numery wierszy i kolumn są rozdzielone przecinkiem.

To samo zadanie — suma elementów z pierwszej kolumny — można wykonać prościej

```
>>sum( M(1:end,1) )
>>sum( M(:,1) )
```

Samodzielny znak dwukropka : oznacza, że zostaną wzięte pod uwagę wszystkie możliwe wartości. W przypadku macierzy M o wymiarze 6 na 6 będą to wszystkie wartości z zakresu od 1 do 6.

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Argumentem funkcji sum (obliczenie sumy argumentów) są wszystkie elementy macierzy M znajdujące się w wierszach od 1 do 6 (1:6) oraz w kolumnie pierwszej (1). Numery wierszy i kolumn są rozdzielone przecinkiem.

To samo zadanie — suma elementów z pierwszej kolumny — można wykonać prościej

```
>>sum( M(1:end,1) )
>>sum( M(:,1) )
```

Samodzielny znak dwukropka : oznacza, że zostaną wzięte pod uwagę wszystkie możliwe wartości. W przypadku macierzy M o wymiarze 6 na 6 będą to wszystkie wartości z zakresu od 1 do 6.

Suma elementów z pierwszego wiersza

Macierz *magiczna*

```
>>M=magic(6) % macierz 6 na 6 o ciekawych właściwościach
```

Suma wszystkich elementów z pierwszej kolumny

```
>>sum( M(1:6,1) )
```

Argumentem funkcji sum (obliczenie sumy argumentów) są wszystkie elementy macierzy M znajdujące się w wierszach od 1 do 6 (1:6) oraz w kolumnie pierwszej (1). Numery wierszy i kolumn są rozdzielone przecinkiem.

To samo zadanie — suma elementów z pierwszej kolumny — można wykonać prościej

```
>>sum( M(1:end,1) )
>>sum( M(:,1) )
```

Samodzielny znak dwukropka : oznacza, że zostaną wzięte pod uwagę wszystkie możliwe wartości. W przypadku macierzy M o wymiarze 6 na 6 będą to wszystkie wartości z zakresu od 1 do 6.

Suma elementów z pierwszego wiersza

```
>>sum( M(1,:) )
```

Suma elementów z trzeciego wiersza

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
```

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
>>sum( M ) % to samo co powyżej
```

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
>>sum( M ) % to samo co powyżej
```

Suma wszystkich wierszy

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
>>sum( M ) % to samo co powyżej
```

Suma wszystkich wierszy

```
>>sum( M(:,:)' ) % każdy wiersz jest sumowany oddzielnie
```

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
>>sum( M ) % to samo co powyżej
```

Suma wszystkich wierszy

```
>>sum( M(:,:)' ) % każdy wiersz jest sumowany oddzielnie
>>sum( M' ) % to samo co powyżej
```

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
>>sum( M ) % to samo co powyżej
```

Suma wszystkich wierszy

```
>>sum( M(:,:)' ) % każdy wiersz jest sumowany oddzielnie
>>sum( M' ) % to samo co powyżej
```

Suma wszystkich elementów macierzy

Suma elementów z trzeciego wiersza

```
>>sum( M(3,:) )
```

Suma elementów z piątej kolumny

```
>>sum( M(:,5) )
```

Suma wszystkich kolumn

```
>>sum( M(:,:) ) % każda kolumna jest sumowana oddzielnie
>>sum( M ) % to samo co powyżej
```

Suma wszystkich wierszy

```
>>sum( M(:,:)' ) % każdy wiersz jest sumowany oddzielnie
>>sum( M' ) % to samo co powyżej
```

Suma wszystkich elementów macierzy

```
>>sum(sum( M ))
```

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:6)
```

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:6)
```

Zostaną wyświetlone wyłącznie elementy leżące w pierwszym, drugim i trzecim wierszu 1:3 oraz jednocześnie w czwartej, piątej i szóstej kolumnie 4:6 (prawy, górny narożnik macierzy).

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:6)
```

Zostaną wyświetlone wyłącznie elementy leżące w pierwszym, drugim i trzecim wierszu 1:3 oraz jednocześnie w czwartej, piątej i szóstej kolumnie 4:6 (prawy, górny narożnik macierzy).

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:end)
```

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:6)
```

Zostaną wyświetlone wyłącznie elementy leżące w pierwszym, drugim i trzecim wierszu 1:3 oraz jednocześnie w czwartej, piątej i szóstej kolumnie 4:6 (prawy, górny narożnik macierzy).

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:end)
```

Słowo end zastępuje maksymalny numer wiersza (kolumny).

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:6)
```

Zostaną wyświetlone wyłącznie elementy leżące w pierwszym, drugim i trzecim wierszu 1:3 oraz jednocześnie w czwartej, piątej i szóstej kolumnie 4:6 (prawy, górny narożnik macierzy).

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:end)
```

Słowo end zastępuje maksymalny numer wiersza (kolumny).

Wykasowanie drugiej kolumny z macierzy M

```
>>whos M
>>M(:,2)=[]
>>whos M
```

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:6)
```

Zostaną wyświetlone wyłącznie elementy leżące w pierwszym, drugim i trzecim wierszu 1:3 oraz jednocześnie w czwartej, piątej i szóstej kolumnie 4:6 (prawy, górny narożnik macierzy).

Wyświetlenie fragmentu macierzy

```
>>M(1:3,4:end)
```

Słowo end zastępuje maksymalny numer wiersza (kolumny).

Wykasowanie drugiej kolumny z macierzy M

```
>>whos M
>>M(:,2)=[]
>>whos M
```

Funkcje max i min

```
>>max( M(1,:) ) % największy element w pierwszym wierszu
>>min( M(:,4) ) % najmniejszy element w czwartej kolumnie
>>max(max(M)) % największy element w całej macierzy
```

Podstawowym typem wykresów jest wykres liniowy. Do jego tworzenia służy komenda `plot` (patrz także `>>help plot`). Argumentem tego polecenia musi być co najmniej nazwa zmiennej, dla której wykres chcemy narysować (np. `plot(y)`), albo definicja wybranej funkcji (np. `plot(sin(x))`). Aby poprawnie narysować jej wykres, często zmuszeni będziemy podawać także wektor argumentów funkcji, np. `plot(x,y)`.

Podstawowym typem wykresów jest wykres liniowy. Do jego tworzenia służy komenda `plot` (patrz także `>>help plot`). Argumentem tego polecenia musi być co najmniej nazwa zmiennej, dla której wykres chcemy narysować (np. `plot(y)`), albo definicja wybranej funkcji (np. `plot(sin(x))`). Aby poprawnie narysować jej wykres, często zmuszeni będziemy podawać także wektor argumentów funkcji, np. `plot(x,y)`.

Rysowanie wykresów

```
>>x = 0:pi/100:2*pi; % definicja argumentów funkcji (od 0 do 2π)
>>y = sin(x); % definicja wartości funkcji
>>plot(x,y); % polecenie rysuje wykres funkcji y(x)
>>xlim([0 2*pi]); % zakres wartości wyświetlanych na osi poziomej
```

Podstawowym typem wykresów jest wykres liniowy. Do jego tworzenia służy komenda `plot` (patrz także `>>help plot`). Argumentem tego polecenia musi być co najmniej nazwa zmiennej, dla której wykres chcemy narysować (np. `plot(y)`), albo definicja wybranej funkcji (np. `plot(sin(x))`). Aby poprawnie narysować jej wykres, często zmuszeni będziemy podawać także wektor argumentów funkcji, np. `plot(x,y)`.

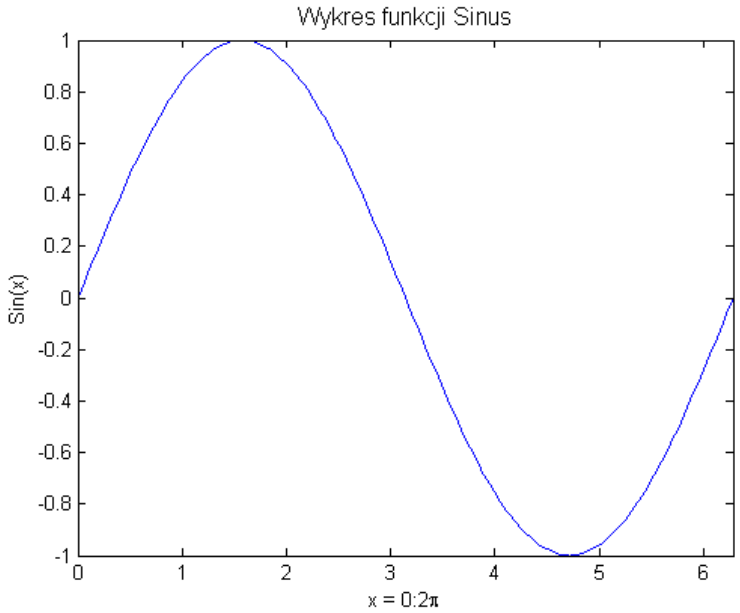
Rysowanie wykresów

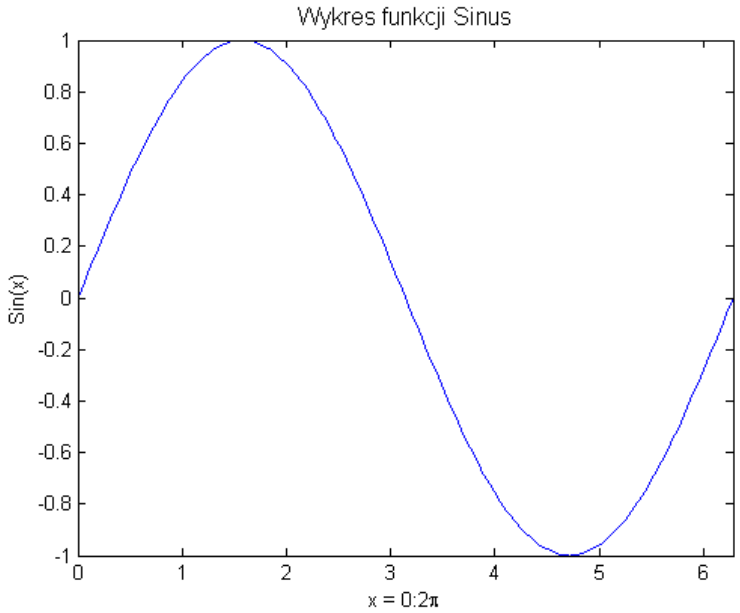
```
>>x = 0:pi/100:2*pi; % definicja argumentów funkcji (od 0 do 2π)
>>y = sin(x); % definicja wartości funkcji
>>plot(x,y); % polecenie rysuje wykres funkcji y(x)
>>xlim([0 2*pi]); % zakres wartości wyświetlanych na osi poziomej
```

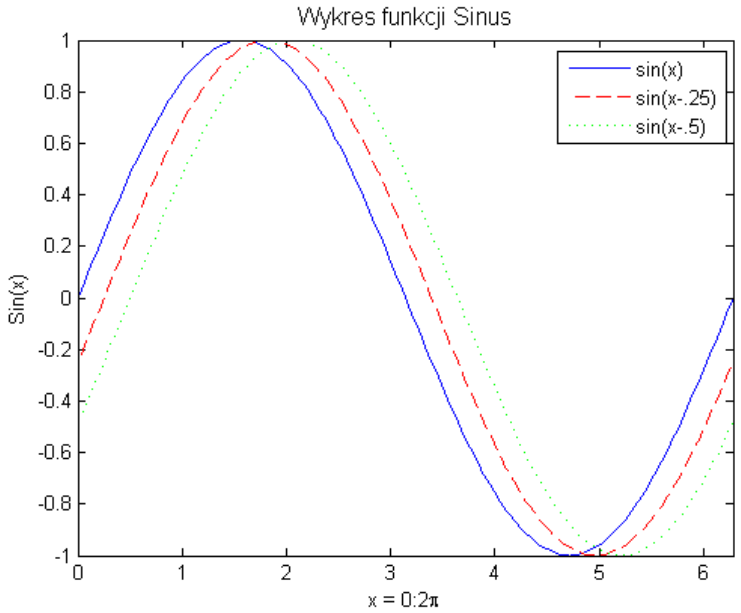
Wynik działania powyższych poleceń wyświetli się na kolejnym slajdzie, wcześniej jednak dodajmy do wykresu odpowiednie opisy.

Dodawanie opisów

```
>>xlabel('x = 0:2\pi'); % opis osi poziomej
>>ylabel('Sin(x)'); % opis osi pionowej
>>title('Wykres funkcji Sinus','FontSize',12); % tytuł wykresu
z podaniem rozmiaru czcionki 12 pt
```


Wykres funkcji Sinus
Sin(x)
x = 0:2π
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6





W pojedynczym oknie znajdować się może także więcej wykresów. Aby do istniejącego okna (z lewej) dodać dwa nowe wykresy (z prawej), wykonać należy poniższe polecenia.

Dodawanie serii danych

```
>>y2 = sin(x-.25); % definiujemy funkcję y2
>>y3 = sin(x-.5); % definiujemy funkcję y3
>>hold on; % wstrzymujemy nadpisywanie wykresu
>>plot(x,y2,'r--',x,y3,'g-.'); % rysujemy jednocześnie dwie funkcje,
definiujemy kolory (r=red, g=green) i style linii
>>legend('sin(x)','sin(x-.25)','sin(x-.5)'); % wstawiamy legendę
```

Polecenia dodatkowe

```
>>grid on; % wyświetla na wykresie linie siatki
>>grid off; % wyłącza linie siatki
>>figure; % użyte bez parametru otwiera nowe okno wykresu
>>close all; % zamyka wszystkie okna wykresów
```

Polecenia dodatkowe

```
>>grid on; % wyświetla na wykresie linie siatki
>>grid off; % wyłącza linie siatki
>>figure; % użyte bez parametru otwiera nowe okno wykresu
>>close all; % zamyka wszystkie okna wykresów
```

Przydatne

```
subplot(m,n,p); % dzieli okno graficzne na macierz o rozmiarze m-na-n
małych wykresów oraz wybiera p-ty region jako aktywny podwykres
text(x0,y0,'przykładowy tekst'); % wstawia tekst w położeniu x0,y0
axis([xmin xmax ymin ymax]); % zakresy osi poziomej i pionowej
plot(x,y,'r-o','LineWidth',2); % rysuje wykres y(x) czerwoną linią
ciągłą grubości 2pt z kółkami w miejscu znaczników punktów
```

Polecenia dodatkowe

```
>>grid on; % wyświetla na wykresie linie siatki
>>grid off; % wyłącza linie siatki
>>figure; % użyte bez parametru otwiera nowe okno wykresu
>>close all; % zamyka wszystkie okna wykresów
```

Przydatne

```
subplot(m,n,p); % dzieli okno graficzne na macierz o rozmiarze m-na-n
małych wykresów oraz wybiera p-ty region jako aktywny podwykres
text(x0,y0,'przykładowy tekst'); % wstawia tekst w położeniu x0,y0
axis([xmin xmax ymin ymax]); % zakresy osi poziomej i pionowej
plot(x,y,'r-o','LineWidth',2); % rysuje wykres y(x) czerwoną linią
ciągłą grubości 2pt z kółkami w miejscu znaczników punktów
```

Zapisywanie wykresów

Aby umieścić wykres w dowolnym dokumencie (np. edytorze tekstu) możemy posłużyć się schowkiem systemowym lub zwyczajnie zapisać go do pliku graficznego.
W pierwszym przypadku z menu **Edit** wybieramy polecenie **Copy Figure**, a następnie używamy polecenia **Wklej** (CTRL+V). Chcąc natomiast zapisać wykres do pliku, z menu **File** wybieramy polecenie **Save As...**, wskazujemy **lokalizację** pliku, jego **nazwę** oraz **typ** (np. *.png).

Skrypt to plik tekstowy zawierający w kolejnych wierszach polecenia do wykonania w środowisku Matlaba. Nazwa podstawowa skryptu musi rozpoczynać się od litery, niedozwolone są znaki oznaczające działania arytmetyczne (np. + czy -), rozszerzeniem nazwy musi być pojedyncza litera m. Przykładowy skrypt, służący do narysowania wykresu funkcji sinus, może wyglądać następująco:

wykres.m

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y);
xlim([0 2*pi]);
xlabel('x = 0:2\pi');
ylabel('Sin(x)');
title('Wykres funkcji Sinus','FontSize',12);
```

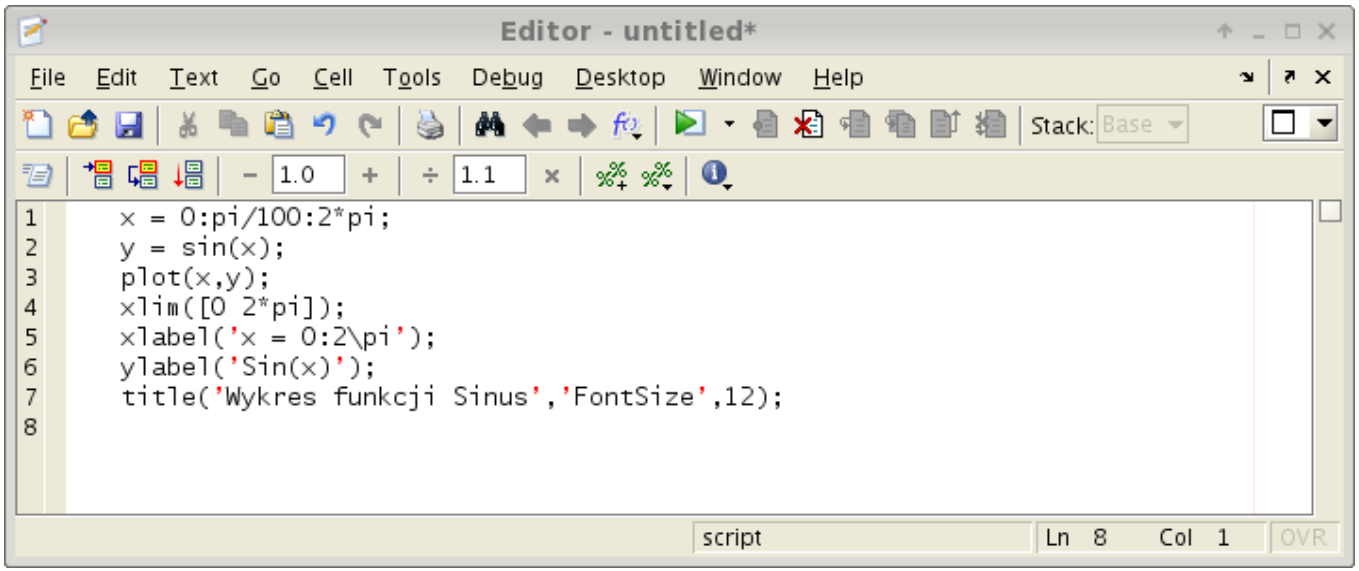
Skrypt to plik tekstowy zawierający w kolejnych wierszach polecenia do wykonania w środowisku Matlaba. Nazwa podstawowa skryptu musi rozpoczynać się od litery, niedozwolone są znaki oznaczające działania arytmetyczne (np. `+` czy `-`), rozszerzeniem nazwy musi być pojedyncza litera `m`. Przykładowy skrypt, służący do narysowania wykresu funkcji `sinus`, może wyglądać następująco:

wykres.m

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y);
xlim([0 2*pi]);
xlabel('x = 0:2\pi');
ylabel('Sin(x)');
title('Wykres funkcji Sinus','FontSize',12);
```

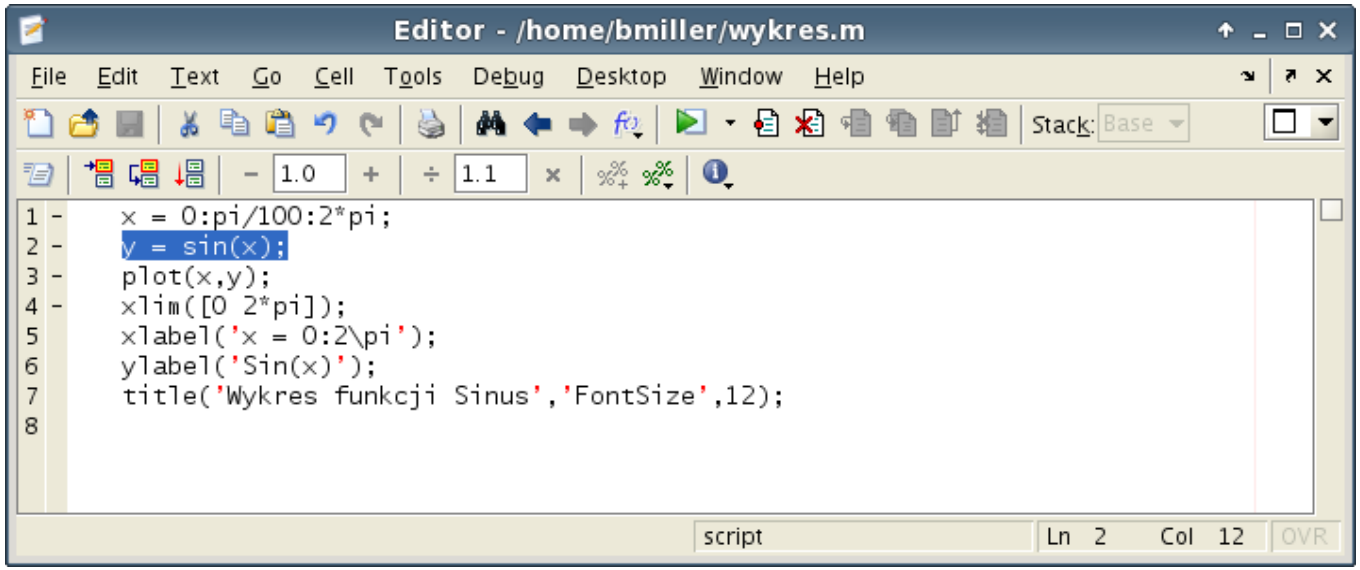
Skrypty można edytować w dowolnym edytorze tekstowym, wygodniej jednak jest skorzystać z wbudowanego edytora Matlaba. Edytor ten wyróżnia kolorami polecenia, ich argumenty i komentarze co znacznie przyspiesza pracę.

Nowy skrypt można utworzyć wybierając z górnego menu Matlaba funkcję **File**, następnie **New** i **M-File**. Po przekopiowaniu do pustego skryptu komend z poprzedniej strony można zapisać skrypt (z górnego menu **File** a następnie **Save As...**, w oknie dialogowym należy określić **położenie** pliku oraz jego **nazwę**). Po zapisaniu skryptu w linii tytułu edytora tekst **untitled** zostanie zastąpiony nazwą skryptu.



Aby wykonać (uruchomić) skrypt należy z górnego menu edytora wybrać polecenie **Debug** a następnie **Run** lub w oknie poleceń Matlaba wpisać nazwę skryptu (bez rozszerzenia) i zatwierdzić klawiszem ENTER. Ten drugi sposób jest możliwy wyłącznie wówczas, gdy skrypt jest zapisany w katalogu bieżącym Matlaba.

Aby zmienić (wyedytować) istniejący plik skryptowy należy go otworzyć wybierając z górnego menu polecenie **File** a następnie **Open...**. Należy wskazać skrypt do edycji i zatwierdzić wybór. Aby zmodyfikować skrypt z poprzedniej strony tak, aby rysował wykres funkcji `cosinus`, należy go otworzyć i zmienić linię numer 2 (właściwa definicja funkcji, na rysunku zaznaczona) oraz linie numer 6 i 7 (opis osi i tytuł wykresu).



Po wprowadzeniu zmian należy skrypt zapisać pod nową nazwą (np. `wykres2.m`) i uruchomić.

Inny przykład edycji skryptu wraz z wynikami przed i po zmianach.

Przykładowy skrypt

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 17; 20 0 5; 4 9 11];
C =a1*B^-1
```

Inny przykład edycji skryptu wraz z wynikami przed i po zmianach.

Przykładowy skrypt

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 17; 20 0 5; 4 9 11];
C =a1*B^-1
```

Wynik działania

```
C =

-0.1117 0.2707 0.0497
-0.4966 -0.1415 0.8318
0.4470 0.0174 -0.1986
```

Inny przykład edycji skryptu wraz z wynikami przed i po zmianach.

Przykładowy skrypt

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 17; 20 0 5; 4 9 11];
C =a1*B^-1
```

Wynik działania

```
C =

-0.1117 0.2707 0.0497
-0.4966 -0.1415 0.8318
0.4470 0.0174 -0.1986
```

Skrypt po zmianach

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 22; 20 23 5; 54 9 11];
C =a1*B^-1
```

Inny przykład edycji skryptu wraz z wynikami przed i po zmianach.

Przykładowy skrypt

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 17; 20 0 5; 4 9 11];
C =a1*B^-1
```

Wynik działania

```
C =

-0.1117 0.2707 0.0497
-0.4966 -0.1415 0.8318
0.4470 0.0174 -0.1986
```

Skrypt po zmianach

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 22; 20 23 5; 54 9 11];
C =a1*B^-1
```

UWAGI:

Przed ponownym wykonaniem skryptu należy zapisać plik na dysku (CTRL+S).

Inny przykład edycji skryptu wraz z wynikami przed i po zmianach.

Przykładowy skrypt

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 17; 20 0 5; 4 9 11];
C =a1*B^-1
```

Wynik działania

```
C =

-0.1117 0.2707 0.0497
-0.4966 -0.1415 0.8318
0.4470 0.0174 -0.1986
```

Skrypt po zmianach

```
% Plik skryptowy Matlaba
a1 = 5.5; % definicja zmiennej a1
% definicja macierzy B o wymiarze 3x3
B = [1 4 22; 20 23 5; 54 9 11];
C =a1*B^-1
```

Wynik działania po zmianach

```
C =

-0.0498 -0.0369 0.1164
-0.0120 0.2820 -0.1042
0.2544 -0.0496 0.0137
```

UWAGI:

Przed ponownym wykonaniem skryptu należy zapisać plik na dysku (CTRL+S).

Praca ze skryptami pozwala na wykorzystanie struktur programowania w środowisku Matlaba np. pętli i instrukcji warunkowych.

Pętla w Matlabie

```
% Skrypt oblicza koleje potęgi liczby 2
for n=1:10          % n - indeks pętli
    A(1,n) = 2^n; % działanie wykonywane w pętli
end                 % koniec pętli
A                   % wyświetlenie wyników
```

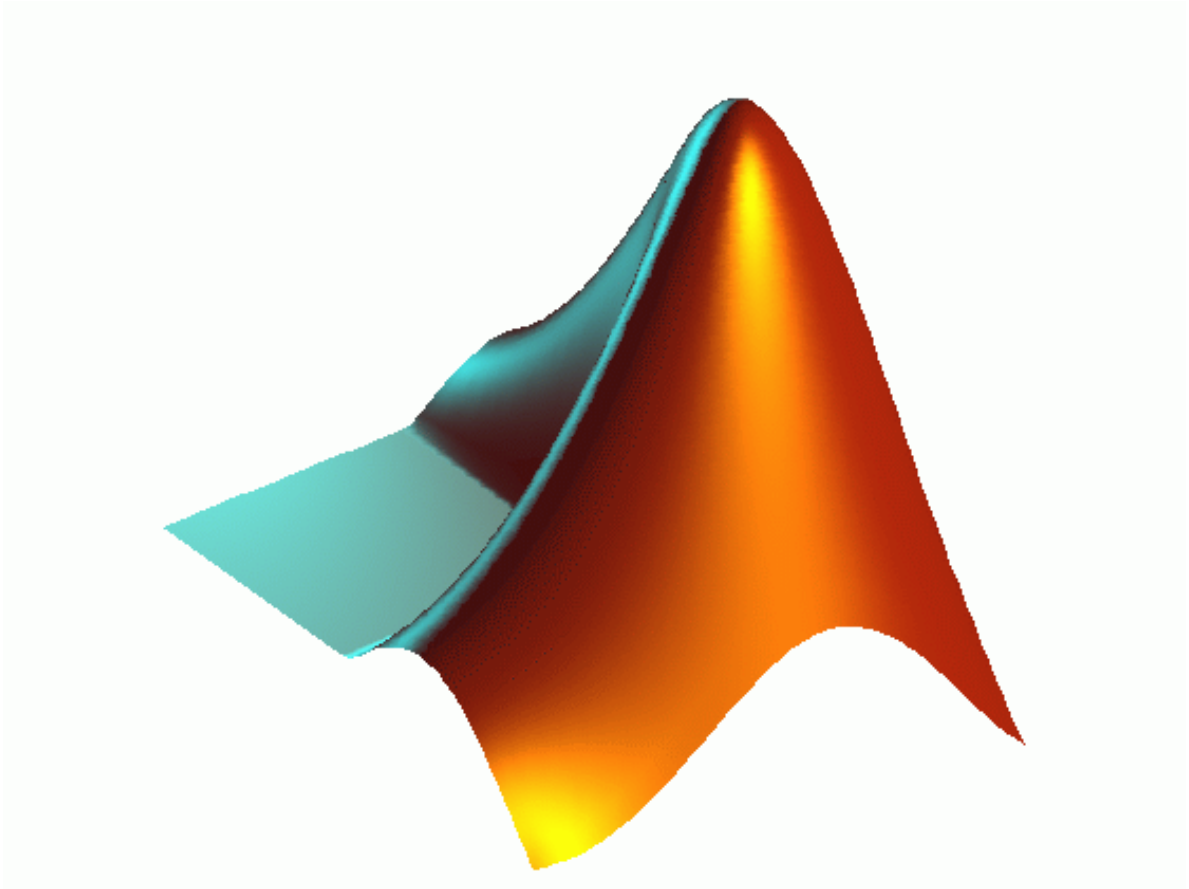
Praca ze skryptami pozwala na wykorzystanie struktur programowania w środowisku Matlaba np. pętli i instrukcji warunkowych.

Pętla w Matlabie

```
% Skrypt oblicza koleje potęgi liczby 2
for n=1:10          % n - indeks pętli
    A(1,n) = 2^n; % działanie wykonywane w pętli
end                 % koniec pętli
A                   % wyświetlenie wyników
```

Instrukcja warunkowa w Matlabie

```
% Skrypt oblicza wartości y zdefiniowanego w dwóch przedziałach
for x = 1:50;           % indeks pętli (jednocześnie argumenty funkcji)
   if x <= 20           % warunek logiczny
      y(x) = 2*x+5;     % działanie jeżeli warunek jest spełniony
   else
      y(x) = -x+65;     % działanie jeżeli warunek nie jest spełniony
   end                  % koniec instrukcji warunkowej
end                     % koniec pętli
plot(y);                % wyniki na wykresie
```



»exit [ENTER]