

7. Funkcje

Funkcja to szczególny przypadek skryptu. Funkcja ma określone parametry wejściowe i wyjściowe. W odróżnieniu od skryptu funkcja musi posiadać określoną strukturę. Zmienne tworzone w czasie jej działania mają charakter lokalny (są tworzone przy uruchomieniu funkcji i usuwane przy jej zakończeniu) – inaczej niż w przypadku zwykłego skryptu.

Podstawową zaletą funkcji jest to, że można ją sparametryzować – uzależnić od podawanych w trakcie wywołania parametrów. Przykładowy skrypt „sincos1.m” zawsze działa w ten sam sposób. Skrypt „sincos1.m” zostanie przerobiony na funkcję. Użytkownik podczas wywołania funkcji będzie mógł zadać przedział zmienności zmiennej x . Przewagą funkcji nad skryptem będzie więc to, że można dzięki niej uzyskać wykresy \sin i \cos w przedziałach np.: $[0, \pi]$, $[-\pi, \pi]$, itp.

7.1. Przykłady funkcji bibliotecznych MATLAB-a

W poprzednich punktach omówionych zostało już kilkanaście różnych funkcji np.:

Funkcja	Parametry wejściowe		Parametry wyjściowe		Opis
	ilość	typ	ilość	typ	
sin	1	macierz	1	macierz	wyznacza macierz, której elementy są równe sinusom odpowiednich elementów macierzy, która jest przekazana jako parametr wejściowy
zeros	1 lub 2	liczby	1	macierz	generuje macierz wypełnioną zerami, jeżeli podczas wywołania podano jeden parametr – generowana jest macierz kwadratowa, jeżeli podano dwa parametry – macierz o różnej liczbie wierszy i kolumn

Czytelnik ma już pewne doświadczenie z wykorzystaniem funkcji standardowych. Powinien więc zdawać sobie sprawę z tego, że jeżeli napisana jest jakaś funkcja (np. \sin), to twórca tej funkcji przewidział wszelkie metody jej użycia.

1. Jeżeli funkcja wymaga jednego parametru wejściowego to podczas jej wywoływania w oknie **Command Window** można podać tylko jeden parametr wejściowy (nie 0 i nie 2 czy 3 – spróbuj wywołać funkcję \sin bez parametrów, z dwoma parametrami).
2. Pewne funkcje są napisane elastycznie, przewidziano różne sposoby działania funkcji w zależności od liczby parametrów wejściowych – np.: zeros (inaczej działa dla jednego parametru wejściowego, inaczej dla dwóch).



Przykładowe funkcje biblioteczne MATLAB-a zawsze zwracały po jednym parametrze wyjściowym. Wartość zwracana przez funkcję mogła trafiać do zmiennej systemowej **ans** lub do zmiennej podanej przez użytkownika np.: $y = \sin(0)$ (wartość $0 - \sin(0) = 0$ – trafia do zmiennej **y**). Istnieją jednak funkcje, które nie zwracają żadnych wartości, istnieją też takie, które zwracają np. dwie wartości.

Funkcja	Parametry wejściowe		Parametry wyjściowe		Opis
	ilość	typ	ilość	typ	
size	1 lub 2	macierz lub macierz i liczba	1 lub 2	wektor lub dwie liczby	funkcja size informuje o rozmiarze macierzy, która jest jej parametrem wejściowym. Jeżeli funkcja wywoływana jest z dwoma parametrami, drugi parametr informuje funkcję czy należy zwrócić informację o ilości wierszy czy kolumn (wartość 1 – ilość wierszy, wartość 2 – ilość kolumn)

Przykładowe wywołania funkcji **size**:

```
>> A = [1 2 3; 4 5 6]
```

```
A =
     1     2     3
     4     5     6
```

Wywołanie	Komentarz
<pre>>> w = size(A) w = 2 3</pre>	<p>wywołanie z:</p> <p>1 parametrem wejściowym i 1 parametrem wyjściowym</p> <p>Parametrem wyjściowym jest wektor, którego pierwszy element zawiera informację o ilości wierszy macierzy A, a drugi element zawiera informację o ilości kolumn</p>
<pre>>> w = size(A, 1) w = 2</pre>	<p>wywołanie z:</p> <p>2 parametrami wejściowymi i 1 parametrem wyjściowym</p> <p>Drugi z parametrów wejściowych – liczba 1 – informuje funkcję o tym, że użytkownik oczekuje tylko informacji o ilości wierszy macierzy A</p> <p>Parametrem wyjściowym jest liczba odpowiadająca ilości wierszy macierzy A</p>
<pre>>> w = size(A, 2)</pre>	<p>wywołanie z:</p>



<pre>w = 3</pre>	<p>2 parametrami wejściowymi i 1 parametrem wyjściowym</p> <p>Drugi z parametrów wejściowych – liczba 2 – informuje funkcję o tym, że użytkownik oczekuje tylko informacji o ilości kolumn macierzy A</p> <p>Parametrem wyjściowym jest liczba odpowiadająca ilości kolumn macierzy A</p>
<pre>>> [w k] = size(A) w = 2 k = 3</pre>	<p>wywołanie z:</p> <p>1 parametrem wejściowym i 2 parametrami wyjściowymi</p> <p>Pierwszy parametr wyjściowym (w) zawiera informację o ilości wierszy macierzy A, a drugi parametr (k) zawiera informację o ilości kolumn</p>

Powyższe przykłady funkcji MATLAB-a zostały przedstawione, aby zachęcić czytelnika do pisania własnych funkcji w sposób elastyczny. Funkcję **size** omówiono, aby przedstawić sposób wywołania funkcji z kilkoma parametrami wyjściowymi.

7.2. Tworzenie funkcji

W celu wygenerowania funkcji należy utworzyć nowy M-plik (podobnie jak w przypadku definiowania skryptu).

Funkcja musi posiadać odpowiednią strukturę. Poniżej przedstawiony został szkielet definicji funkcji:

```
function [pwy1, pwy2, ..., pwyN] = nazwa_funkcji (pwe1, pwe2, ... , pweM)
```

```
    komentarz dokumentujący (widoczny po wywołaniu: help nazwa_funkcji)
```

```
    ciąg instrukcji
```

Opis:

pwy1, pwy2, ..., pwyN – to wymyślone przez projektanta nazwy parametrów wyjściowych

pwe1, pwe2, ... , pweM – to wymyślone przez projektanta nazwy parametrów wejściowych

nazwa_funkcji – wymyślona nazwa funkcji – **nazwa ta powinna odpowiadać nazwie M-pliku.**

Uwaga!

Nadając nazwy parametrom czy samej funkcji należy kierować się zasadami:

- w nazwach mogą występować tylko litery, cyfry i znak „_”,
- nazwa musi być ciągła (nie może składać się z dwóch członów oddzielonych spacją),



- w nazwie nie mogą wystąpić polskie litery,
- nazwa musi rozpoczynać się od litery.

Występujące w przedstawionej powyżej definicji funkcji parametry wejściowe i wyjściowe są opcjonalne. Oznacza to, że można napisać funkcję bez parametrów wejściowych i wyjściowych jak i funkcję z wieloma parametrami wejściowymi i wyjściowymi.

Funkcja bez parametrów wejściowych i wyjściowych spełnia podobną rolę do skryptu, a jedyną różnicą pomiędzy taką funkcją a skrypcem jest to, że zmienne tworzone w czasie działania funkcji są usuwane przy jej zakończeniu, a zmienne utworzone w czasie działania skryptu są dostępne po jego zakończeniu.

Przykład 1.

Funkcja wyświetlająca wykresy sin i cos w przedziale $[0, 2\pi]$

Definicja funkcji	Wywołania
(funkcja została zapisana w pliku „ sincos2.m ”) function SinCos2 % Najprostsza funkcja % Generuje wykresy funkcji sin i cos % w przedziale [0, 2 pi] x=[0:pi/10:2*pi]; y1=sin(x); y2=cos(x); plot(x,y1,x,y2);	aby uruchomić funkcję należy wpisać polecenie: >> sincos2 aby wywołać pomoc związaną z funkcją należy wpisać polecenie: >> help sincos2 Najprostsza funkcja Generuje wykresy funkcji sin i cos w przedziale [0, 2 pi]

Przykład 2.

Funkcja wyświetlająca wykresy funkcji sin i cos w dowolnym przedziale – określanym w momencie wywołania funkcji.

Funkcja będzie miała dwa parametry wejściowe oznaczające początek i koniec przedziału w którym generowany będzie wykres. Parametry tym w poniższym przykładzie nadano nazwy **a** i **b**.



Definicja funkcji	Wywołania
<p>(funkcja została zapisana w pliku „sincos3.m”)</p> <pre> function SinCos3(a, b) % Funkcja generuje wykresy funkcji % sin i cos w podanym przedziale % Przykładowe wywołanie: % SinCos3(-pi, pi) x=[a : pi/10 : b]; y1=sin(x); y2=cos(x); plot(x,y1,x,y2); </pre>	<p>Funkcji, która została zdefiniowana jako funkcja z dwoma parametrami nie można wywołać z inną liczbą parametrów.</p> <p>Nie można więc wywołać funkcji:</p> <pre>>> sincos3</pre> <p>Aby wywołać funkcję należy podać początek i koniec przedziału np.: od -2π do 2π:</p> <pre>>> sincos3(-2*pi, 2*pi)</pre>

Przykład 3.

Funkcja obliczająca wartość podanego elementu ciągu geometrycznego.

Aby obliczyć wartość n -tego elementu ciągu geometrycznego należy znać wartość pierwszego elementu tego ciągu (a_1) oraz iloraz ciągu (q). Zależność pozwalająca wyznaczyć wartość n -tego elementu wyraża się wzorem: $a_n = a_1 q^{n-1}$.

Przykładowa funkcja powinna więc mieć trzy parametry wejściowe (pierwszy element a_1 , iloraz ciągu q , numer elementu ciągu, który ma być wyznaczony n) oraz jeden parametr wyjściowy – wartość n -tego elementu a_n .

Definicja funkcji	Wywołania
<p>(funkcja została zapisana w pliku „element.m”)</p> <pre> function [an] = element(a1, q, n) % Oblicza wartość n-tego elementu % ciągu geometrycznego. Należy ją wywołać % podając 3 parametry: % 1. – pierwszy element % 2. – iloraz ciągu % 3. – nr wyznaczanego elementu an = a1*q^(n-1); </pre>	<p>Aby wywołać funkcję należy podać trzy parametry.</p> <p>Poniższe wywołanie wyznaczy wartość czwartego elementu ciągu o parametrach: $a_1 = 1$, $q = 3$</p> <pre>>> element(1, 3, 4)</pre> <pre>ans =</pre> <pre> 27</pre>



Uwaga !

Funkcja, która zwraca obliczoną wartość poprzez parametr wyjściowy powinna zawierać linię: parametr wyjściowy = obliczona wartość (w przykładzie: $a_n = a_1^{(n-1)}$). Jeżeli zabraknie takiego przypisania wartości dla któregoś z parametrów, parametr ten będzie miał wartość nieokreśloną.

Przykład 4.

Funkcja obliczająca pierwiastki równania kwadratowego (wer.1).

Aby obliczyć wartości pierwiastków równania kwadratowego ($ax^2 + bx + c = 0$) należy znać wartości współczynników tego równania (tzn.: a, b oraz c). Równanie kwadratowe ma zawsze dwa pierwiastki zespolone (jeżeli $\Delta \geq 0$ pierwiastki są rzeczywiste, jeżeli $\Delta < 0$ pierwiastki są liczbami zespolonymi).

Przykładowa funkcja powinna więc mieć trzy parametry wejściowe – współczynniki równania (**a, b, c**) oraz dwa parametry wyjściowe – pierwiastki (**x1, x2**).

Definicja funkcji	Wywołania
<p>(funkcja została zapisana w pliku „Kwadratowe1.m”)</p> <pre> function [x1, x2] = kwadratowe1 (a, b, c) % Funkcja oblicza pierwiastki równania % kwadratowego. % Należy ją wywołać podając 3 parametry: % współczynniki równania delta = b^2 - 4*a*c; x1=(-b - delta ^ 0.5)/(2*a); x2=(-b + delta ^ 0.5)/(2*a); </pre>	<p>Aby wywołać funkcję należy podać trzy parametry.</p> <p>W wywołaniu należy również uwzględnić fakt, że funkcja zwraca dwa parametry wyjściowe.</p> <p>Poniższe wywołanie obliczy wartości pierwiastków równania kwadratowego $x^2 + x - 2 = 0$ i umieści je w zmiennych pierw1 i pierw2.</p> <pre> >> [pierw1, pierw2] = kwadratowe1(1, 1, -2) pierw1 = -2 pierw2 = 1 </pre>

Przykład 5.

Funkcja obliczająca pierwiastki równania kwadratowego (wer.2).

W drugiej wersji funkcji założono, że pierwiastki równania będą zwracane nie w dwóch parametrach – ale w jednym parametrze. Parametr wyjściowy w przypadku tej funkcji będzie więc wektorem, w którego pierwszym elemencie umieszczony zostanie pierwszy pierwiastek, a w drugim drugi.



Definicja funkcji	Wywołania
<p>(funkcja została zapisana w pliku „Kwadratowe2.m”)</p> <pre> function [x] = kwadratowe2 (a, b, c) % Funkcja oblicza pierwiastki równania % kwadratowego. % Należy ją wywołać podając 3 parametry: % współczynniki równania delta = b^2 - 4*a*c; x(1)=(-b - delta ^ 0.5)/(2*a); x(2)=(-b + delta ^ 0.5)/(2*a); </pre>	<p>Przykładowe wywołanie będzie rozwiązywało równanie kwadratowe o współczynnikach jak w przykładzie poprzednim.</p> <pre> >> pierw = kwadratowe2(1, 1, -2) pierw = -2 1 </pre>

Uwaga!

Parametr x do momentu wystąpienia w funkcji linii: $x(2) = (-b + \text{delta} ^ 0.5) / (2 * a)$ jest macierzą o wymiarach 1×1 . Odwołanie $x(1)$ jest odwołaniem do pierwszego elementu tej macierzy. Odwołanie $x(2)$ nie jest interpretowane jako błąd (mimo, że x składa się tylko z jednego elementu) lecz jako chęć powiększenia x do wymiarów 1×2 . Gdyby w funkcji pojawiło się odwołanie $x(10)$ – parametr x rozszerzyłby się do wymiarów 1×10 . Gdyby pojawiło się odwołanie np. $x(2, 1)$ wymusiłoby to dodanie nowego wiersza do macierzy x tzn. x rozszerzyłby się do wymiarów 1×10 itd.

Nadużywanie automatycznego powiększania macierzy nie jest zalecane. Powiększenie macierzy może wymagać przekopiowania wszystkich jej elementów do nowego – większego obszaru pamięci – jest to operacja czasochłonna. Aby tego uniknąć lepiej jest na wstępie nadać macierzy odpowiednie wymiary tzn. odwołać się do elementu leżącego w ostatnim wierszu i ostatniej kolumnie macierzy. Np.: można wykonać polecenie $x(2,10)=0$ tym samym definiując x jako macierz o rozmiarach 2×10 .

Powyższa uwaga nie ma większego znaczenia dla funkcji „Kwadratowe2”, gdyż operacja powiększania i tak odbywa się tu tylko jeden raz.

7.3. Parametry funkcyjne

Często istnieje potrzeba przesłania w postaci parametru do funkcji – innej funkcji, np. do funkcji rysującej wykres można by przysyłać funkcję, która powinna być narysowana (tak właśnie jest w przypadku funkcji **fplot**).



Spróbujmy napisać własną wersję funkcji **fplot**. Funkcja powinna dostawać trzy parametry wejściowe:

- nazwę funkcji, która powinna być narysowana,
- początek przedziału,
- koniec przedziału.

Pierwsza linia definicji takiej funkcji może więc wyglądać następująco:

```
function Wykres(funkcja, a, b)
```

Opis:

parametr **funkcja** – to nazwa funkcji, która zostanie narysowana,

parametr **a** – to początek przedziału

parametr **b** – to koniec przedziału

Wykres – to nazwa definiowanej funkcji

Kolejną rzeczą, która powinna znaleźć się w definicji funkcji jest wygenerowanie odpowiedniego wektora reprezentującego dziedzinę rysowanej funkcji:

```
x = a : 0.1 : b;
```

Następnie należy wygenerować wektor, którego elementy będą odpowiadały wartościom rysowanej funkcji w kolejnych punktach wektora **x**. Gdyby z góry było wiadomo, że rysowaną funkcją jest funkcja np. **sin** to wektor wartości mógłby być wygenerowany:

```
y = sin(x);
```

W przypadku, kiedy rysowana funkcja jest przekazywana w jednym z parametrów wejściowych należy do obliczenia jej wartości wykorzystać funkcję **feval**.

Funkcja **feval(nazwa_funkcji, x1, ..., xn)** oblicza wartość funkcji o nazwie podanej w parametrze **nazwa_funkcji** dla argumentów **x1, ..., xn**, np.: **y = feval('sin', [0 : 0.1 : pi])**.

Stąd w definiowanej funkcji należałoby umieścić linię:

```
y = feval(funkcja, x);
```

Po wygenerowaniu wektora **x** i wektora **y**, można już wykorzystać funkcję **plot** do narysowania wykresu:

```
plot(x, y);
```



Ostatecznie, pełna definicja funkcji Wykres może wyglądać następująco:

Definicja funkcji	Wywołania
<p>(funkcja została zapisana w pliku „Wykres.m”)</p> <pre> function Wykres (funkcja, a, b) % Rysuje wykres dowolnej funkcji % w podanym przedziale % Należy ją wywołać podając 3 parametry: % 1. – nazwa rysowanej funkcji % 2. – początek przedziału % 3. – koniec przedziału x = a ; 0.1 : b; y = feval(funkcja, x); plot(x, y); </pre>	<p>Należy narysować funkcję abs w przedziale [-10, 10].</p> <pre> >> wykres('abs', -10, 10) </pre>

Ćwiczenia:

1. Stworzyć w folderze roboczym **Work** katalog na własne skrypty i funkcje np. o nazwie **Gr301A**. Wszystkie skrypty i funkcje należy zapisywać wewnątrz utworzonego folderu.
2. Napisz funkcję „DoTrzeciej” obliczającą trzecią potęgę elementów wektora wejściowego (parametr wejściowy – wektor np. **x**, parametr wyjściowy wektor np. **y** zawierający elementy **x³**).
3. Napisz funkcję „SumaCiagu” obliczającą sumę ciągu arytmetycznego od **a** do **b** ze skokiem **c** (**a**, **b**, **c** powinny być parametrami wejściowymi funkcji).

Wskazówka: wygeneruj wektor od **a** do **b** ze skokiem **c** i wykorzystaj funkcję MATLAB-a **sum**.

4. Napisz funkcję „Kolo” obliczającą pole i obwód koła (parametrem wejściowym funkcji powinien być promień, wyjściowym wektor zawierający pole i obwód).
5. Napisz funkcję „Prostopadloscian” obliczającą objętość, pole powierzchni ścian i sumę długości krawędzi prostopadłościanu o podstawie prostokąta. (parametry wejściowe: długości krawędzi podstawy i wysokość).
6. Napisz funkcję „NpoK” obliczającą ilość **k**-elementowych kombinacji bez powtórzeń z **n** elementów:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (n, k \text{ powinny być parametrami wejściowymi funkcji}).$$



Wskazówka: W celu wyliczenia silni z n wygeneruj wektor $[1, 2, 3, \dots, n]$ a następnie oblicz iloczyn jego elementów przy pomocy funkcji MATLAB-a **prod**.

7. Napisz funkcję „UkladRownan” rozwiązującą układ równań liniowych:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \quad \text{Parametrami wejściowymi funkcji są: macierz}$$

$$\dots \dots \dots \dots \dots \dots$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \quad \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \text{ oraz wektor } [b_1 \ b_2 \ \dots \ b_n].$$

Parametrem wyjściowym funkcji powinno być uzyskane rozwiązanie \mathbf{x} oraz wartość wyznacznika macierzy współczynników.

Wskazówka: Układ równań może być zapisany w postaci macierzowej: $\mathbf{A} \mathbf{x} = \mathbf{B}$. Aby wyznaczyć rozwiązanie należy skorzystać z faktu, że: $\mathbf{x} = \mathbf{A}^{-1} \mathbf{B}$.