

ĆWICZENIA 02

OPERACJE MACIERZOWE I TABLICOWE, FUNKCJE.

Cel zajęć

Implementacja funkcji rozwiązujących proste problemy inżynierskie.

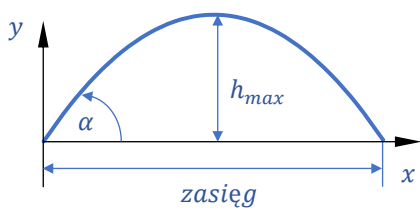
Materiały do przygotowania

Materiały umieszczone na stronie przedmiotu:

- Wykład 01

Zadania do wykonania

Rzut ukośny



położenie:	$x(t) = v_0 t \cos \alpha$
	$y(t) = v_0 t \sin \alpha - \frac{g t^2}{2}$
zasięg	$z = \frac{v_0^2 \sin 2\alpha}{g}$
wysokość	$h_{max} = \frac{(v_0 \sin \alpha)^2}{2g}$
czas lotu	$t_l = \frac{2 v_0 \sin \alpha}{g}$

1. Funkcja `rzut1` wyznacza współrzędne toru lotu w rzucie ukośnym. Parametrami wejściowymi funkcji są

- v_0 – prędkość początkowa ciała,
- α – kąt (w stopniach) pod jakim ciało zostało rzucone,
- t – chwila dla której należy wyznaczyć położenie ciała,

parametrami wyjściowymi:

- x i y – współrzędne opisujące położenie ciała.

Sprawdź działanie funkcji i wyznacz położenie ciała w chwili $t = 1s$ przyjmując, że ciało zostało wyrzucone z prędkością $v_0 = 20 \frac{m}{s}$ pod kątem $\alpha = 30^\circ$.

Uwaga! Kąty wprowadzane do funkcji trygonometrycznych muszą być podawane w radianach. Zamianę stopni na radiany umożliwia np. funkcja `deg2rad`¹.

2. Funkcja `rzut2` jest zmodyfikowaną wersją funkcji `rzut1`. Wprowadzona modyfikacja powinna pozwalać na wyznaczenie współrzędnych toru lotu dla wielu chwil zapisanych w parametrze t (parametr powinien być wektorem).

Sprawdź działanie funkcji i wyznacz położenie ciała w czasie od $0s$ do $10s$ z odstępem $0,5s$. Pozostałe parametry przyjmij jak w zadaniu poprzednim.

3. Funkcja `rzut3` wyznacza zasięg, wysokość maksymalną oraz czasu lotu rzutu ukośnego. Parametrami wejściowymi funkcji są

- v_0 – prędkość początkowa ciała,
- α – kąt (w stopniach) pod jakim ciało zostało rzucone,

parametrami wyjściowymi:

- z – zasięg,
- h_{max} – wysokość maksymalna,
- t_l – czas lotu.

Sprawdź działanie funkcji dla parametrów: $v_0 = 20 \frac{m}{s}$ i $\alpha = 30^\circ$ wyznaczając wszystkie oraz tylko wybrane parametry wyjściowe.

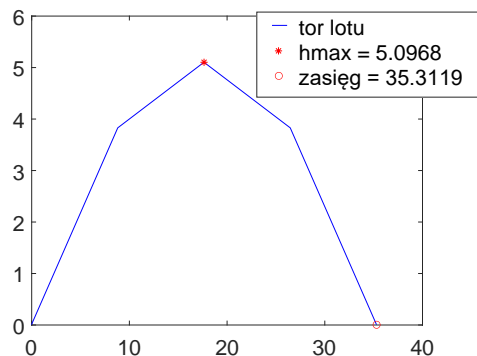
Uwaga! Do pomijania parametrów wyjściowych służy operator \sim (tylda)²:

4. Funkcja `rzut4` wyświetla wykres przedstawiający tor lotu obiektu w rzucie ukośnym. Parametrami wejściowymi funkcji są

- v_0 – prędkość początkowa ciała,
- α – kąt (w stopniach) pod jakim ciało zostało rzucone,
- n – liczba punktów dla których wyznaczone zostaną współrzędne toru.

Funkcja nie ma żadnych parametrów wyjściowych.

Uwaga!



- W funkcji, dla podanych parametrów rzutu, należy najpierw wyznaczyć czas lotu t_l . Następnie należy przygotować wektor zawierający n chwil w przedziale $[0, t_l]$ (należy skorzystać z funkcji `linspace`³) i w oparciu o przygotowany wektor należy wyznaczyć współrzędne toru.
- Na wykresie, oprócz toru, należy również wykreślić 2 punkty charakterystyczne pokazujące maksymalną wysokość oraz zasięg.
- Wykres powinien mieć legendę opisaną w sposób pokazany na rysunku powyżej. Rysunek można uzupełnić o legendę posługując się funkcją `legend`⁴. Łączone opisy w legendzie zbudowane z tekstów i wartości liczbowych można przygotować zamieniając liczby na teksty posługując się funkcją `string`⁵ a fragmenty tekstów można łączyć posługując się operatorem `+`⁶.

Analiza potrzeb i kosztów

5. Funkcja `analiza1` wyznacza ilość surowców niezbędnych do wyprodukowania wybranego stopu aluminium. Parametrami wejściowymi funkcji są:

- `sklad` – wektor opisujący procentową ilość domieszek *Cu*, *Mg*, *Mn*, *Si* oraz *Zn* w stopie,
- `produkcja` – planowana wielkość produkcji (w tonach),

parametrem wyjściowym jest wektor *ilosc* zawierający ilość domieszek i samego aluminium potrzebnych do produkcji stopu.

Sprawdź działanie funkcji i wyznacz ilość składników potrzebnych do wyprodukowania 1000 ton aluminium 6061 (*Cu* – 0%, *Mg* – 1%, *Mn* – 0%, *Si* – 0,6% oraz *Zn* – 0%).

Uwaga! Obliczenia należy przeprowadzić jedną operacją mnożenia (wektor opisujący skład powinien zostać uzupełniony o zawartość procentową aluminium w stopie).

6. Funkcja *analiza2* wyznacza oprócz ilości składników również koszty związane z ich zakupem oraz całkowity koszt związany z produkcją stopu. Parametrami wejściowymi funkcji są:

- *sklad* – wektor opisujący procentową ilość domieszek *Cu*, *Mg*, *Mn*, *Si* oraz *Zn* w stopie,
- *produkcja* – planowana wielkość produkcji (w tonach),
- *cena* – wektor kolumnowy zawierający ceny składników stopu.

Parametrami wyjściowymi funkcji są:

- *ilosc* – wektor z ilościami składników,
- *koszty* – wektor z kosztami składników,
- *koszt* – całkowity koszt składników.

Uwaga! Funkcja *analiza2* powinna wykorzystywać funkcję *analiza1*.

7. Funkcja *analiza3* jest zmodyfikowaną wersją funkcji *analiza2*. Wprowadzona modyfikacja powinna pozwalać na wykonanie analizy dla kilku stopów aluminium jednocześnie – należy przyjąć, że parametr *sklad* jest macierzą, której kolejne wiersze opisują procentową ilość domieszek w kolejnych stopach a parametry *produkcja* oraz *cena* są wektorami kolumnowymi zawierającymi planowaną wielkość produkcji oraz ceny składników stopu.

Sprawdź działanie funkcji i wyznacz ilość składników potrzebnych do wyprodukowania 1000 ton aluminium 6061 (*Cu* – 0%, *Mg* – 1%, *Mn* – 0%, *Si* – 0,6% oraz *Zn* – 0%) oraz 2000 ton aluminium 2024 (*Cu* – 4,4%, *Mg* – 1,5%, *Mn* – 0,6%, *Si* – 0% oraz *Zn* – 0%).

Uwaga! Kolejne obliczenia należy przeprowadzić pojedynczymi operacjami: macierz opisująca skład powinna zostać uzupełniona o dodatkową kolumnę z zawartością procentową aluminium, w obliczeniach *ilości* należy wykorzystać mnożenie tablicowe⁷ a przy obliczeniu *kosztów* mnożenie macierzowe.

Lp.	Składnia funkcji	Przykład
1	<code>RADIANY = deg2rad(STOPNIE)</code> przelicza wielkość kąta podanego w stopniach (parametr <code>STOPNIE</code>) na radiany (parametr <code>RADIANY</code>)	<pre>>> deg2rad(180) ans = 3.1416</pre>
2	<code>[wy1, ~, wy3] = funkcja(we1, ...)</code> operator tylda pozwala na pominięcie wybranego parametru wyjściowego funkcji	<pre>>> X = [3 2 1 5 4]; >> [~, p] = max(X) p = 4</pre>
3	<code>WYNIK = linspace(X1,X2,N)</code> generuje wektor <code>WYNIK</code> o <code>N</code> równo rozłożonych elementach z przedziału <code>[X1, X2]</code>	<pre>>> X = linspace(0,10,4) X = 0 3.3333 6.6667 10.0000</pre>
4	<code>legend(etykieta1, ..., etykietaN)</code> tworzy legendę opisując kolejne elementy wykresu etykietami: <code>etykieta1, ..., etykietaN</code>	<pre>>> produkt = [1,2,3]; >> koszt = [2,1,4]; >> czas = [3,2,3]; >> plot(produkt, koszt, "r*", ... produkt, czas, "bo") >> legend("koszt", "czas")</pre>
5	<code>WYNIK = string(WARTOSC)</code> konwertuje liczbę (parametr <code>WARTOSC</code>) na łańcuch znakowy (typ <code>string</code>) i zapisuje go w <code>WYNIKU</code>	<pre>>> s = string(2.5) s = "2.5"</pre>
6	<code>WYNIK = TEKST1 + TEKST2 + ...</code> zapisuje w <code>WYNIKU</code> tekst będący połączeniem tekstów <code>TEKST1, TEKST2, ...</code>	<pre>>> s = "a " + "b " + "c" s = "a b c"</pre>
7	<code>WYNIK = WEKTOR1 .* WEKTOR1</code> <code>WYNIK = MACIERZ1 .* MACIERZ2</code> <code>WYNIK = WEKTOR .* MACIERZ</code> <code>WYNIK</code> zawiera iloczyny odpowiadających sobie elementów wektorów lub macierzy (takie same rozmiary), lub iloczyny elementów wektora przez kolumny lub wiersze macierzy (patrz przykład obok)	<pre>>> A = [1 2 3]; B = [1;2]; >> C = [1 1 1; 2 2 2]; >> D = A .* C, E = B .* C D = 1 2 3 2 4 6 E = 1 1 1 4 4 4</pre>

