

Rozdział 2

Wstępne zapoznanie się z octavem

W przypadku przeznaczenia dwóch laboratoriów scenariusz każdego zajęcia polega na rozwiązaniu możliwie dużej ilości kolejnych zadań z tego rozdziału. W przypadku jednych zajęć należy dokonać wyboru rozwiązując tylko kilkanaście najważniejszych pozycji z listy zadań.

Zadania w tym rozdziale ilustrują podstawowe operacje, struktury i własności octave'a. Przedstawimy w zadaniach octave jako kalkulator naukowy. Omówimy też operator dwukropek służący np. tworzeniu indeksów. Przetestujemy, jak tworzyć macierze, wektory; jak zapisywać zmienne do plików (czytać z plików) w formatach: tekstowym i binarnym. Sprawdzimy tworzenie macierzy z podmacierzy, wycinanie podmacierzy i inne podstawowe operacje na macierzach - mnożenie, dodawanie, transponowanie, funkcje matematyczne od macierzy, normy wektorów/macierzy.

Zadania obejmują również tworzenie wykresów funkcji matematycznych przy pomocy funkcji octave'a `plot()`. Sprawdzimy też tworzenie i używanie skryptów i funkcji (m-pliki) w octave, oraz podstawowe instrukcje warunkowe i pętle:

- `if else endif;`
- `switch case endswitch`
- `while() do endwhile;`
- `do .. until();`
- `for .. endfor.`

Zadania obejmą też wskaźniki do funkcji (function handle) i operator `@` - zwracający wskaźnik do funkcji.

Zadanie 1 **Octave jako kalkulator.** Otwórz sesję octave'a. Zapoznaj się z pomocą do funkcji `sqrt()` oraz `sin()`. Policz w octave'ie, ile wynosi pierwiastek z 5 oraz policz wartość funkcji sin na tym pierwiastku.

Zadanie 2 **Operacje macierzowe. Operator dwukropek.**

- Utwórz wektor z liczbami od 1 do 20 oraz wektor ze wszystkimi liczbami parzystymi od -6 do 4.
- Utwórz dowolne macierze 3×4 A i 3×5 B, a następnie macierz 3×8 C, której pierwsze 3 kolumny to A, a kolejne to B.
- Z macierzy C 'wytnij' podmacierz D składającą się z 1 głównego minora tzn. 3×3 od C(1,1) do C(3,3).
- Zamień kolejność kolumn D.
- Zamień kolejność wierszy D.
- Wytnij dolnotrójkątną, a potem górnortrójkątną część macierzy D
- Wstaw D z powrotem do C jako główny minor.
- Policz $\sin(D) = (\sin(D_{ij}))$ od D.
- Zapisz D do pliku (binarnego i ASCII) - zamień element D(1,1) na -100 i wczytaj nową macierz do octave'a.

Zadanie 3 **Funkcje matematyczne od macierzy. Normy macierzy i wektorów**

Policz dyskretną normę maksimum od $(\sin(x))^2$ na $[0, 1]$ (wektorowo-czyli bez użycia pętli).

Zadanie 4 **Wykresy funkcji matematycznych.**

- Narysuj wykres funkcji $\sin(x)$ na odcinku $[0, 4]$.
- Narysuj wykres funkcji $\sin(x)$ na odcinku $[0, 4]$ - wykres powinien być podpisany, narysowany za pomocą gwiazdek w kolorze czerwonym.
- Narysuj w jednym oknie podpisane wykresy funkcji $\sin(x)$ i $\log(x)$ na odcinku $[1, 6]$ podpisane odpowiednio.

Zadanie 5 Znajdź przybliżone maksimum i minimum funkcji

$$f(x) = x * (3 + 2 * \cos(x))$$

na odcinku $[-1, 5]$ bez użycia pętli, oraz przybliżenia punktów ekstremalnych.

Powtórz to zadanie dla jakiegoś wielomianu stopnia dwa i trzy np. $x^3 + x^2 - x - 4$.

Zadanie 6 Funkcje w octave'ie, m-pliki, czyli tzw. pliki funkcyjne.

Utwórz funkcję w m-pliku obliczającą dla zadanego x wartość funkcji

$$f(x) = (\sin(x))^2.$$

Zadanie 7 Zmienne globalne

Utwórz m-plik z funkcją octave'a obliczającą wartość funkcji matematycznej z parametrem: $\sin(a * x)$ - parametr przekaz jako zmienną globalną.

Zadanie 8 Implementacja wektorowa funkcji w octave'ie.

Utwórz m-plik z funkcją obliczającą wartość funkcji $f(x) = 1 + (\cos(x))^2$ dla argumentu będącego macierzą, tzn. jeśli $X = (x_{i,j})_{i,j}$ macierz wymiaru $M \times N$, to funkcja powinna zwrócić Y macierz wymiaru $M \times N$ taką, że

$$Y = (y_{i,j})_{i,j}, \quad y_{i,j} = f(x_{i,j}).$$

Narysuj wykres f na odcinku $[-1, 5]$ z wykorzystaniem tylko jednego wywołania tak zaimplementowanej funkcji octave'a.

Zadanie 9 Funkcje anonimowe.

Utwórz funkcję uchwyt do funkcji anonimowej, która dla danego argumentu x zwraca wartość równą $(\sin(x))^2$.

Zadanie 10 Pętle, instrukcje warunkowe, instrukcja printf().

- Zapoznaj się z pomocą octave'a do pętli **while**, pętli **for**, instrukcji warunkowej **if** oraz funkcji drukującej napisy na ekranie **printf()**.

- Przy pomocy pętli

for (k = ...)

endfor

while (warunek stopu)

endwhile

oblicz sumę

$$S_N = 1 + \dots + N$$

dla $N = 100$.

- Użyj instrukcji warunkowej

if ()

endif

by sprawdzić, czy otrzymane S_N zostało obliczone poprawnie, tzn. czy otrzymaliśmy $0.5 * N * (N + 1)$

- wyprowadź na ekran komunikat używając

printf ()