

#### LISTA 4

Do przygotowania:

1. Na czym polega prealokacja tablic?
2. Na czym polega wektoryzacja? (przeanalizuj przykłady z wykładu)

Zadania

##### Zad. 1

Wygeneruj **a)** za pomocą pętli bez prealokacji, **b)** za pomocą pętli z prealokacją, **c)** bez użycia pętli wektor 100 000-elementowy zgodny z poniższą formułą:

$$A(n) = -1 * \frac{3n^2 + 4n + 1}{5n^3 + 3n - 1}$$

Który ze sposobów jest szybszy? Pamiętaj o średnikach na końcu linii, aby nie wyświetlać każdej z 100 000 liczb.

##### Zad 2.

Wygeneruj przykładową stronę z dziennika ocen 30 osobowej klasy. Każdy z uczniów uzyskał 7 ocen. Ocena musi spełniać następujące wymagania  $x \in \{x: X \sim U(2,5) \wedge x \in N\}$ .

- a) Wylicz średnie arytmetyczne każdego z uczniów za pomocą pętli „for” oraz w sposób zwektoryzowany. Porównaj czas obliczeń w obu przypadkach.
- b) Wylicz średnie ważone: wektor wag  $w = [3 \ 3 \ 1 \ 1 \ 1 \ 0.5 \ 0.5]$ . Porównaj czasy obliczeń kodu opartego o pętlę „for”, kodu zwektoryzowanego.
- c) Sprawdź uzyskiwane czasy dla obliczeń przy 300 000 uczniów.

Pamiętaj, żeby nie powtarzać wielokrotnie tych samych obliczeń: np. sumy elementów wektora wag.

##### Zad3.

Zoptymalizuj kod. Sprawdź, jakie jest uzyskane przyspieszenie oraz który element optymalizacji daje największy zysk?

```
alpha=10:5:80; % kat w stopniach
g=10;          %Przyspieszenie
V_0=20;        % Predkosc poczatkowa
t=0:0.0001:5;  %Ustalenie czasu

for k=1:length(alpha)
    V_0y(k) = sind(alpha(k))*V_0; %skladowa y (pionowa) predkosci poczatkowej
    V_0x(k) = cosd(alpha(k)).*V_0; %skladowa x (pozioma) predkosci poczatkowej
    for w=1:length(t)
        y(w,k)=V_0y(k).*t(w)-(g*t(w).^2)/2; % ruch w kierunku osi y (wznoszenie i opadanie);
        x(w,k)=V_0x(k).*t(w); % ruch w kierunku osi x
    end
end
for k=1:length(alpha)
    plot(x(:,k),y(:,k))
    axis([0 40 0 20]);
    animacja(k) = getframe; % tworzy ramke z wykresu/rysunku
```

```
end
movie(animacja)
```

#### Zad 4.

Utwórz sygnał sinusoidalny na przedziale [0, 20] próbkowany z częstotliwością 100 Hz. Dodaj do niego szum modelowany przez zmienną losową  $S \sim N(0, 0.1)$ . Napisz instrukcje, które uśrednią sygnał oknem o szerokości 3 (wartość uśrednionego sygnału w momencie „i” to średnia z wartości z sygnału w momencie „i”, „i – 1”, „i+1”). W rozwiązaniu wykorzystaj pętle „for”, a następnie zaproponuj zwektoryzowane rozwiązanie. Porównaj czasy realizacji przy częstotliwości próbkowania  $f_1 = 100$  Hz oraz większych częstotliwościach  $f_2 = 1$  kHz,  $f_3 = 10$  kHz.

\* (zadanie dodatkowe) Zaimplementuj procedurę uśredniającą, jako funkcję. Argumentami wejściowymi tej funkcji powinny być sygnał oraz szerokość okna.

#### Zad 5\*. (zadanie dodatkowe)

Zaimplementuj algorytm klastrowania K-średnich, którego początkowy etap został przedstawiony na wykładzie. Do testów możesz wykorzystać zestaw punktów wygenerowanych poniższym skryptem:

```
N=10000
x1=randn(N,2).*repmat([0.5 1],N,1)+repmat([5 3],N,1);
x2=randn(N,2).*repmat([0.9 0.9],N,1)+repmat([3 -6],N,1);
x3=randn(N,2).*repmat([0.8 2],N,1)+1.5;
plot(x1(:,1),x1(:,2),'*r',x2(:,1),x2(:,2),'ob',x3(:,1),x3(:,2),'+g')
data=[x1;x2;x3];
```