

# Programowanie i metody numeryczne

Aleksandra Fijałkowska

22 marca 2018

## Zadanie 4. (1 p)

Znaleźć epsilon maszynowy dla zmiennych typu float i double. Wynik porównać z wartością otrzymaną ze statycznej metody epsilon w szablonie std::numeric\_limits.

## Zadanie 5. (4 p)

a). Napisz funkcję liczącą pochodną dwupunktową metodą centralną:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad (1)$$

Sprawdzić działanie programu dla  $f = \sin(x)$  dla  $x = \frac{2\pi}{3}$  oraz  $f = x^3$  dla  $x = 2, 5$ , wykorzystaj zmienne typu float.

Wskazówka: (przyda się w kolejnej części zadania)

Zaprojektuj program tak, aby każda część algorytmu znalazła się w osobnej funkcji np.:

```
float getFunValue(float x);\nfloat getFunDerivative(float x);\nfloat getFunDerivativeCentralMeth(float x, float h);\nfloat getDerivativeError(float x, float h);
```

b). Zmodyfikuj program dla funkcji  $f = x^3$  tak, aby w wygodny sposób umożliwić sprawdzenie algorytmu dla zmiennych typu float i double. Funkcje przyjmujące zmienne typu float przepisz na szablony funkcji.

Wskazówka:

Przykładowy szablon funkcji, która zwraca kwadrat przyjętego argumentu:

```
template<typename T> T square(T x)\n{\n    return x*x;\n}
```

c). Wyrysuj błąd metody w funkcji parametru  $h$  w skali log-log dla zmiennych typu float i double.

## Zadanie 6. (2 p)

Napisz program sumujący liczby od 1 do 10000 zmiennymi int, float i double. Sprawdzić wynik sumując od przodu, od tyłu i metodą Kahana [https://en.wikipedia.org/wiki/Kahan\\_summation\\_algorithm3](https://en.wikipedia.org/wiki/Kahan_summation_algorithm3).

**Zadanie 7. (2 p)**

Napisz program alokujący dynamicznie tablice liczb całkowitych o wczytanym rozmiarze. Następnie program powinien wyzerować podaną liczbę elementów z początku tablicy lub nie, w zależności od wyboru użytkownika. Po tym użytkownik zdecyduje, czy zwolnić zaalokowaną pamięć i następnie kiedy zakończyć program.

**Zadanie 8. (3 p)**

Napisz procedurę wczytującą liczby typu float z wejścia standardowego. Po wpisaniu pustej linii procedura powinna wyświetlić wczytane liczby na ekran w odwrotnej kolejności.

Wskazówka:

Najłatwiejszą metodą rozwiązania tego zadania jest skorzystania z wektora (za 2 punkty). Rozmiar wektora może być łatwo powiększany metodą `push_back`. Zachęcam do zaimplementowania trudniejszego algorytmu, bazującego na tablicy (za 3 punkty). Po przekroczeniu początkowo zadanego rozmiaru tablicy powinna ona dwukrotnie zwiększać swój rozmiar.

**Zadanie 9. (1 p)**

Zapisz procedurę z zadania 2 przy pomocy szablonu tak, by mogła przyjmować tablicę dowolnego typu.