

Laboratorium 2

Podstawy Metod Komputerowych w Obliczeniach Inżynierskich Różniczkowanie, Splot, Gradient

Marek Wodziński

AGH UST

21.10.2020

Spis Treści

- 1 Różniczkowanie
 - Wprowadzenie
 - Zadania
- 2 Splot
 - Wprowadzenie
 - Zadania
- 3 Gradient oraz pochodne wyższych rzędów
 - Wprowadzenie
 - Zadania

Wprowadzenie



Zadanie 1

Zadanie 1

Zaimplementuj funkcje (za pomocą pętli for, bez użycia gotowych funkcji) do obliczenia pochodnej funkcji jednej zmiennej "centralnie", "do przodu" i "do tyłu". Jakie są zalety poszczególnych rozwiązań? Jakie są wady? Kiedy warto użyć jednej wersji zamiast drugiej?

```
def our_derivative(x, step=1, mode="central "):  
    #central, forward, backward  
    pass
```

Wprowadzenie



Zadanie 2

Zadanie 2

Zaimplementuj funkcje do obliczania splotu 1-D (iteracyjnie, bez transformacji Fouriera, bez użycia gotowych funkcji). Porównaj działanie zaimplementowanej funkcji z funkcją biblioteczną (scipy) oraz czas wykonywania obliczeń. Która funkcja jest szybsza? O ile? Dlaczego? Czy wyniki są identyczne? Porównaj działanie funkcji bibliotecznej liczącą splot z definicji oraz z transformacji Fouriera (sprawdź dla różnych wielkości tablic wejściowych).

```
def our_convolution(x1, x2, mode="ad"):  
    #pad, reflect  
    pass
```

Wprowadzenie

DEMO

Zadanie 3

Zadanie 3

Zaimplementuj funkcje (za pomocą pętli for, bez użycia gotowych funkcji) do obliczenia pochodnej funkcji jednej zmiennej "centralnie", która umożliwi wybór rzędu pochodnej (1, 2, 3, 4) oraz trybu (splot lub sekwencyjnie). Porównaj wynik z wartościami analitycznymi na przykładowym wielomianie wyższego stopnia. Przeanalizuj błędy numeryczne (m.in. porównaj obliczenia za pomocą splotu z sekwencyjnym obliczaniem pierwszej pochodnej). Porównaj czas wykonywania obliczeń. Skąd się biorą różnice? Które rozwiązanie jest szybsze? Które rozwiązanie jest bardziej stabilne? Które rozwiązanie jest obciążone mniejszym błędem?

```
def our_derivative(x, step=1, order=1, mode="conv "):  
    # 1, 2, 3, 4  
    #conv / sequential  
    pass
```


Zadanie 4

Zadanie 4

Zaimplementuj funkcje (za pomocą pętli for, bez użycia gotowych funkcji) do obliczenia gradientu centralnego funkcji dowolnej liczby zmiennych (wejście w postaci tablicy N-wymiarowej). Porównaj działanie (błędy numeryczne, szybkość obliczeń) do wbudowanej funkcji gradient w bibliotece numpy.

```
def our_gradient(x, step=1):  
    pass
```

Zadanie 5

Zadanie 5

Bazując na wszystkich wcześniejszych zadaniach zaimplementuj funkcję do obliczenia gradientu funkcji dowolnej liczby zmiennych stopnia 1-4 (jako argument wejściowy), o wybranym sposobie jego obliczenia (centralny, do przodu, wstecz) z zachowaniem jak najmniejszego błędu (odpowiednie wyprowadzenia zostały przedstawione na wykładzie). Wykorzystaj splot.

```
def ultimate_gradient(x, step=1, order=1,
    mode="central"):
    # order = {1, 2, 3, 4}
    # mode = {"forward", "backward", "central"}
    pass
```