Laboratorium 1

Podstawy Metod Komputerowych w Obliczeniach Inżynierskich
Informacje organizacyjne, wprowadzenie, narzędzia i biblioteki

Marek Wodziński

AGH UST

15.10.2020

Spis Treści

Informacje ogólne

Informacje ogólne

Prowadzący laboratorium: **Marek Wodziński**
Kontakt:

- B1 p. 205, wodzinski@agh.edu.pl
- Zajęcia będą odbywać się z wykorzystaniem MS Teams.

Informacje ogólne

Prowadzący laboratorium: **Marek Wodziński**
Kontakt:

- B1 p. 205, wodzinski@agh.edu.pl
- Zajęcia będą odbywać się z wykorzystaniem MS Teams.

Sala laboratoryjna:

- Zajęcia zdalne z zaliczeniem w formie tradycyjnej.
- Sala do kolokwium zaliczeniowego zostanie podana w późniejszym terminie.

Informacje ogólne

Prowadzący laboratorium: **Marek Wodziński**
Kontakt:

- B1 p. 205, wodzinski@agh.edu.pl
- Zajęcia będą odbywać się z wykorzystaniem MS Teams.

Sala laboratoryjna:

- Zajęcia zdalne z zaliczeniem w formie tradycyjnej.
- Sala do kolokwium zaliczeniowego zostanie podana w późniejszym terminie.

Terminy:

- Obligatoryjny termin czwartkowy (19:45-22:00).
- Dodatkowe terminy konsultacyjne (jak w planie).

Zagadnienia

Zagadnienia

- **Laboratorium 1:** Zajęcia wprowadzające, przedstawienie programu laboratorium. Praktycznie wprowadzenie do problemów numerycznych. Przypomnienie podstaw biblioteki NumPy/SciPy/Matplotlib. Wprowadzenie do biblioteki PyTorch. Zadania do samodzielnego rozwiązania w ramach opanowania podstaw wyżej wymienionych bibliotek (3h).

Zagadnienia

- **Laboratorium 1:** Zajęcia wprowadzające, przedstawienie programu laboratorium. Praktycznie wprowadzenie do problemów numerycznych. Przypomnienie podstaw biblioteki NumPy/SciPy/Matplotlib. Wprowadzenie do biblioteki PyTorch. Zadania do samodzielnego rozwiązania w ramach opanowania podstaw wyżej wymienionych bibliotek (3h).
- **Laboratorium 2:** Różniczkowanie numeryczne. Sposoby obliczania gradientu. Podstawowe typu gradientu. Splot. Analiza błędów numerycznych w porównaniu do rozwiązań analitycznych. Implementacja gradientów analitycznych. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 1:** Zajęcia wprowadzające, przedstawienie programu laboratorium. Praktycznie wprowadzenie do problemów numerycznych. Przypomnienie podstaw biblioteki NumPy/SciPy/Matplotlib. Wprowadzenie do biblioteki PyTorch. Zadania do samodzielnego rozwiązania w ramach opanowania podstaw wyżej wymienionych bibliotek (3h).
- **Laboratorium 2:** Różniczkowanie numeryczne. Sposoby obliczania gradientu. Podstawowe typu gradientu. Splot. Analiza błędów numerycznych w porównaniu do rozwiązań analitycznych. Implementacja gradientów analitycznych. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 3:** Całkowanie numeryczne. Podstawowe algorytmy całkowania numerycznego. Analiza błędu całkowania. Całkowanie numeryczne, a analityczne. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 3:** Całkowanie numeryczne. Podstawowe algorytmy całkowania numerycznego. Analiza błędu całkowania. Całkowanie numeryczne, a analityczne. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 4:** Wprowadzenie do problemu interpolacji. Interpolacja liniowa/dwuliniowa. Interpolacja wyższych rzędów. Błąd interpolacji. Aproksymacja. Ekstrapolacja. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 3:** Całkowanie numeryczne. Podstawowe algorytmy całkowania numerycznego. Analiza błędu całkowania. Całkowanie numeryczne, a analityczne. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 4:** Wprowadzenie do problemu interpolacji. Interpolacja liniowa/dwuliniowa. Interpolacja wyższych rzędów. Błąd interpolacji. Aproksymacja. Ekstrapolacja. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 5:** Metody rozwiązywania układów równań. Obliczanie wartości i wektorów własnych. Rozkład według wartości osobliwych (SVD). Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 6:** Macierz Hessego (Hessian). Macierz Jacobiego (Jacobian). Wyznacznik macierzy Jacobiego. Wprowadzenie do problemu optymalizacji. Metoda najmniejszych kwadratów. Metoda Newtona. Metoda Gaussa-Newtona. Podstawowe równania różniczkowe. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 6:** Macierz Hessego (Hessian). Macierz Jacobiego (Jacobian). Wyznacznik macierzy Jacobiego. Wprowadzenie do problemu optymalizacji. Metoda najmniejszych kwadratów. Metoda Newtona. Metoda Gaussa-Newtona. Podstawowe równania różniczkowe. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 7:** Problem wartości odstających. Algorytm RANSAC. Algorytm PROSAC. Detekcja wartości odstających. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).

Zagadnienia

- **Laboratorium 6:** Macierz Hessego (Hessian). Macierz Jacobiego (Jacobian). Wyznacznik macierzy Jacobiego. Wprowadzenie do problemu optymalizacji. Metoda najmniejszych kwadratów. Metoda Newtona. Metoda Gaussa-Newtona. Podstawowe równania różniczkowe. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 7:** Problem wartości odstających. Algorytm RANSAC. Algorytm PROSAC. Detekcja wartości odstających. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 8:** Kolokwium zaliczeniowe - zajęcia stacjonarne (3h).

Zagadnienia

- **Laboratorium 6:** Macierz Hessego (Hessian). Macierz Jacobiego (Jacobian). Wyznacznik macierzy Jacobiego. Wprowadzenie do problemu optymalizacji. Metoda najmniejszych kwadratów. Metoda Newtona. Metoda Gaussa-Newtona. Podstawowe równania różniczkowe. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 7:** Problem wartości odstających. Algorytm RANSAC. Algorytm PROSAC. Detekcja wartości odstających. Przykłady praktyczne. Zadania do samodzielnego rozwiązania (3h).
- **Laboratorium 8:** Kolokwium zaliczeniowe - zajęcia stacjonarne (3h).
- **Laboratorium 9:** Dodatkowy termin zaliczeniowy (1h).

Harmonogram

- **Laboratorium 1:** 15.10.2020
- **Laboratorium 2:** 22.10.2020
- **Laboratorium 3:** 29.10.2020
- **Laboratorium 4:** 05.11.2020
- **Laboratorium 5:** 12.11.2020
- **Laboratorium 6:** 19.11.2020
- **Laboratorium 7:** 26.11.2020

Harmonogram

- **Laboratorium 1:** 15.10.2020
- **Laboratorium 2:** 22.10.2020
- **Laboratorium 3:** 29.10.2020
- **Laboratorium 4:** 05.11.2020
- **Laboratorium 5:** 12.11.2020
- **Laboratorium 6:** 19.11.2020
- **Laboratorium 7:** 26.11.2020
- **Laboratorium 8:** 17.12.2020
- **Laboratorium 9:** 14.01.2021

Harmonogram w trakcie semestru może ulec zmianie.

Zasady zaliczenia

Zasady zaliczenia

W trakcie laboratorium można zdobyć maksymalnie **100** punktów, gdzie:

1. Kolokwium zaliczeniowe: **100 pkt**

Zasady zaliczenia

W trakcie laboratorium można zdobyć maksymalnie **100** punktów, gdzie:

1. Kolokwium zaliczeniowe: **100 pkt**

Oceny wystawiane są na podstawie skali ocen AGH. Poprawa oceny z laboratorium jest możliwa jedynie w przypadku uzyskania oceny negatywnej. Studentowi przysługują dwa dodatkowe terminy zaliczenia polegające na kolokwium zaliczeniowym analogicznym do kolokwium w trakcie ostatniego laboratorium. Dopuszczalna jest maksymalnie jedna nieusprawiedliwiona nieobecność.

Cel przedmiotu

Cel przedmiotu

Cel

Celem przedmiotu jest poznanie **podstaw** metod numerycznych stosowanych w **praktyce**.

Cel przedmiotu

Cel

Celem przedmiotu jest poznanie **podstaw** metod numerycznych stosowanych w **praktyce**.

Nie będziemy omawiać aspektów teoretycznych analizy numerycznej (twierdzeń, dowodów).

Powiązanie z innymi przedmiotami

Powiązanie z innymi przedmiotami

1. Cyfrowe przetwarzanie sygnałów (IV semestr)

Powiązanie z innymi przedmiotami

1. Cyfrowe przetwarzanie sygnałów (IV semestr)
2. Wizualizacja danych medycznych (V semestr)

Powiązanie z innymi przedmiotami

1. Cyfrowe przetwarzanie sygnałów (IV semestr)
2. Wizualizacja danych medycznych (V semestr)
3. Automatyka i robotyka (V semestr)

Powiązanie z innymi przedmiotami

1. Cyfrowe przetwarzanie sygnałów (IV semestr)
2. Wizualizacja danych medycznych (V semestr)
3. Automatyka i robotyka (V semestr)
4. Podstawy przetwarzania obrazów cyfrowych (V semestr)

Powiązanie z innymi przedmiotami

1. Cyfrowe przetwarzanie sygnałów (IV semestr)
2. Wizualizacja danych medycznych (V semestr)
3. Automatyka i robotyka (V semestr)
4. Podstawy przetwarzania obrazów cyfrowych (V semestr)
5. Techniki obrazowania medycznego (VI semestr)

Co będziemy robić..

Co będziemy robić..

- Implementacja prostych algorytmów numerycznych w Pythonie korzystając z NumPy/SciPy/PyTorch

Co będziemy robić..

- Implementacja prostych algorytmów numerycznych w Pythonie korzystając z NumPy/SciPy/PyTorch
- Wykorzystanie gotowych implementacji bardziej złożonych algorytmów

Co będziemy robić..

- Implementacja prostych algorytmów numerycznych w Pythonie korzystając z NumPy/SciPy/PyTorch
- Wykorzystanie gotowych implementacji bardziej złożonych algorytmów
- Analiza zachowania poszczególnych metod w zależności od parametrów i zadanego problemu

Co będziemy robić..

- Implementacja prostych algorytmów numerycznych w Pythonie korzystając z NumPy/SciPy/PyTorch
- Wykorzystanie gotowych implementacji bardziej złożonych algorytmów
- Analiza zachowania poszczególnych metod w zależności od parametrów i zadanego problemu
- Porównanie gotowych implementacji z opracowanymi samodzielnie

Czego nie będzie - Zaawansowane algorytmy

Nie będziemy (a powinniśmy):

Czego nie będzie - Zaawansowane algorytmy

Nie będziemy (a powinniśmy):

- Implementować zaawansowanych algorytmów numerycznych - szczególnie dotyczących optymalizacji i rozwiązywania równań różniczkowych wyższych rzędów

Czego nie będzie - Zaawansowane algorytmy

Nie będziemy (a powinniśmy):

- Implementować zaawansowanych algorytmów numerycznych - szczególnie dotyczących optymalizacji i rozwiązywania równań różniczkowych wyższych rzędów
- Implementować algorytmów dziedzinowych (przetwarzanie sygnałów medycznych, obrazowanie medyczne, wizja komputerowa w medycynie, robotyka medyczna)

Czego nie będzie - Zaawansowane algorytmy

Nie będziemy (a powinniśmy):

- Implementować zaawansowanych algorytmów numerycznych - szczególnie dotyczących optymalizacji i rozwiązywania równań różniczkowych wyższych rzędów
- Implementować algorytmów dziedzinowych (przetwarzanie sygnałów medycznych, obrazowanie medyczne, wizja komputerowa w medycynie, robotyka medyczna)
- Dokonywać implementacji praktycznych (C/C++, odpowiedni dostęp do pamięci, wykorzystanie cache) ukierunkowanych na gotowe, stosowalne biblioteki

Czego nie będzie - Zaawansowane algorytmy

Nie będziemy (a powinniśmy):

- Implementować zaawansowanych algorytmów numerycznych - szczególnie dotyczących optymalizacji i rozwiązywania równań różniczkowych wyższych rzędów
- Implementować algorytmów dziedzinowych (przetwarzanie sygnałów medycznych, obrazowanie medyczne, wizja komputerowa w medycynie, robotyka medyczna)
- Dokonywać implementacji praktycznych (C/C++, odpowiedni dostęp do pamięci, wykorzystanie cache) ukierunkowanych na gotowe, stosowalne biblioteki
- Zrównoleglać obliczeń ani na CPU, ani na GPU (OpenMP/OpenCL/CUDA)

Python - Wprowadzenie

Python - Zadania do samodzielnego rozwiązania

Zadanie 1

Napisz funkcję przyjmującą dowolną liczbę argumentów, która zwróci ich sumę podniesioną do kwadratu.

```
print("Example: ")
print(func(1, 2, -4, 4))
print(func(1, 2, -4, 4, 5))
```

```
Example:
9
64
```

Python - Zadania do samodzielnego rozwiązania

Zadanie 2

Napisz generator, który będzie generować liczby całkowite w zadanym zakresie.

```
print("Example: ")
example_gen = gen(1, 5) # Start/Stop
for item in example_gen:
    print(item)
```

```
Example:
1
2
3
4
```

NumPy - Wprowadzenie

NumPy - Zadania do samodzielnego rozwiązania

Zadanie 3

Zaproponuj metodę umożliwiającą zbliżone wyznaczenie wartości ϵ, tj. najmniejszej wartości, która po dodaniu do jedności nie będzie skutkować zwiększeniem wartości. Nie wykorzystuj funkcji wbudowanych.

```
print("Epsilon for float16: ", epsilon_f16())
print("Epsilon for float32: ", epsilon_f32())
print("Epsilon for float64: ", epsilon_f64())
```

```
Epsilon for float16:  0.0004883
Epsilon for float32:  5.9604645e-08
Epsilon for float64:  1.1102230246251565e-16
```

SciPy - Wprowadzenie

Matplotlib - Wprowadzenie

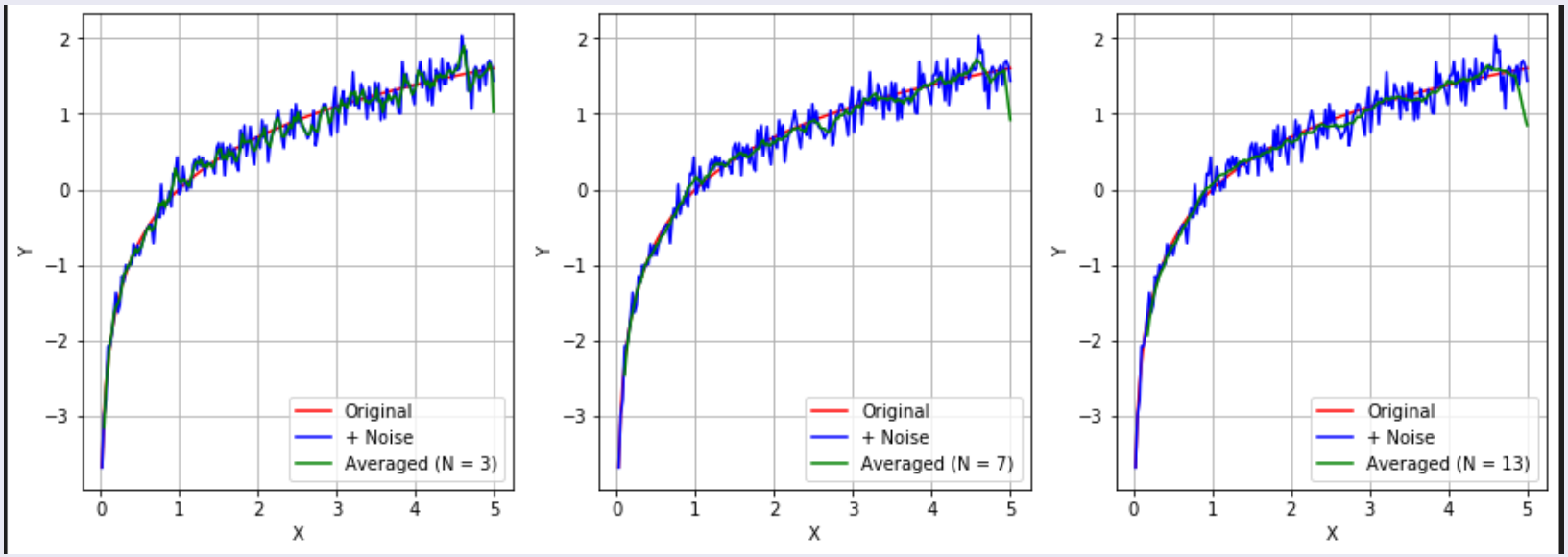
Matplotlib - Zadania do samodzielnego rozwiązania

Zadanie 4

Wygeneruj wartości funkcji logarytmu naturalnego w zakresie zmiennej od 0 do 5 dla 200 równomiernie, monotoniczne rozłożonych wartości. Dodaj do niej szum normalny o wartości średniej równej 0 i odchyleniu standardowym równym 0.2. Narysuj wykres wygenerowanych wartości. Następnie spróbuj usunąć szum poprzez uśrednienie wartości sąsiadujących stosując różny rozmiar okna (3, 7, 13). Zignoruj wartości brzegowe. Narysuj wynik. Do wykresu dodaj siatkę, podpisy osi, legendę. Jakie są wady takiego podejścia do usuwania szumu?

Matplotlib - Zadania do samodzielnego rozwiązania

Zadanie 4



PyTorch - Wprowadzenie

PyTorch - Zadania do samodzielnego rozwiązania

Zadanie 5

Powtórz Zadanie 3 wykorzystując bibliotekę PyTorch.

Zadanie 6

Wygeneruj dwa wektory 10 000 000 losowych liczb zarówno w bibliotece NumPy jak i PyTorch. Dodaj je do siebie i zmierz czas dodawania. Która biblioteka jest szybsza? Sprawdź czas dla różnych typów danych (int/float16/float32/float64). Jeżeli masz dostęp do GPU, powtórz operację po przesłaniu wektorów do pamięci karty.