



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Algorytmy obliczeniowe [S1EiT1>AOB]

Przedmiot

Kierunek studiów

Elektronika i telekomunikacja

Rok/Semestr

1/1

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

6,00

Koordynatorzy

dr hab. inż. Rafał Krenz

rafal.krenz@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawowe wiadomości z matematyki i fizyki oraz programowania z zakresu szkoły średniej.

Cel przedmiotu

Celem przedmiotu jest nauczanie studenta rozwiązywania podstawowych inżynierskich problemów obliczeniowych występujących w elektronice i telekomunikacji stosując podejście algorytmiczne.

Przedmiotowe efekty uczenia się

Wiedza:

1. ma podstawową wiedzę dotyczącą algorytmizacji problemów obliczeniowych oraz obliczeń numerycznych
2. zna składnię języka Python oraz charakterystyczne zastosowania tego środowiska obliczeniowego
3. ma wiedzę nt. podstawowych algorytmów obliczeniowych (rozwiązywanie układów równań liniowych, całkowanie, różniczkowanie, znajdowanie miejsc zerowych funkcji)

Umiejętności:

1. potrafi dokonać algorytmizacji problemów obliczeniowych
2. potrafi wykorzystać w praktyce język Python oraz korzystać z jego środowiska obliczeniowego
3. ma umiejętność doboru właściwego dla danego problemu algorytmu obliczeniowego

Kompetencje społeczne:

1. rozumie, że wraz z rozwojem systemów komputerowych algorytmy obliczeniowe stają się coraz doskonalsze, ale równocześnie ma świadomość ich ograniczonego zastosowania

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w ramach wykładu jest weryfikowana na podstawie pisemnego egzaminu. Składa się on z 5-7 zadań otwartych, różnie punktowanych. Próg zaliczeniowy wynosi 50% punktów. Prace omawiane są indywidualnie ze studentami.

Umiejętności nabyte w ramach zajęć laboratoryjnych weryfikowane są na podstawie realizowanych na bieżąco zadań podawanych przez prowadzącego i wykonywanych przy użyciu komputera. Wymagane jest poprawne wykonanie co najmniej 50% realizowanych zadań.

Treści programowe

1. Wprowadzenie:

Pojęcie algorytmu, opis tekstowy algorytmu, graficzny opis algorytmu za pomocą schematów blokowych), proste przykłady algorytmów z różnych dziedzin życia i techniki - ilustracja instrukcji podstawienia, pętli i testowania spełnienia warunku, przykłady (algorytm Euklidesa). Wykorzystywane narzędzia programistyczne (Anaconda, PyCharm).

2. Elementy języka Python jako narzędzia obliczeń inżynierskich:

Elementy języka Python niezbędne do realizacji obliczeń inżynierskich i podstaw obliczeń numerycznych: podstawowe typy danych stosowane w Pythonie, operatory matematyczne i sekwencja ich wykonywania, instrukcje wprowadzania danych wejściowych, instrukcje do przedstawiania wyników obliczeń i komunikatów w formie tekstowej i graficznej, realizacja pętli, funkcje logiczne i ich testowanie, realizacja instrukcji warunkowych, obliczenia wykonywane na macierzach i wektorach, funkcje i ich zasady wykorzystania

3. Przykłady algorytmicznego podejścia do obliczeń:

3.1. Przedstawienie funkcji ciągłych w postaci ciągu próbek (warunki prawidłowego doboru odstępu próbek) - tablicowanie funkcji, badanie punktów nieciągłości, ilustracja graficzna tablicowanych funkcji

3.2. Obliczenia numeryczne pochodnej funkcji, podstawowe algorytmy obliczania całek oznaczonych (metody: prostokątów, trapezów, Simpsona), ocena błędów całkowania numerycznego na przykładach konkretnych funkcji.

3.3. Zasada obliczeń iteracyjnych, kryterium zatrzymania obliczeń, wyznaczanie numeryczne pierwiastków równań, poszukiwanie maksimum lub minimum funkcji metodą iteracyjną (minimum lokalne, minimum globalne - zastosowanie gradientu funkcji)

3.4. Numeryczne podstawy rozwiązywania układów równań liniowych, metoda Gaussa, metody iteracyjne

3.5. Interpretacja danych eksperymentalnych: dopasowanie krzywych do danych pomiarowych metodą sumy najmniejszych kwadratów, rozkłady prawdopodobieństwa danych pomiarowych (estymacja histogramu), obliczanie parametrów statystycznych danych pomiarowych (średnia, wariancja, odchylenie standardowe)

Tematyka zajęć

brak

Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami realizowanymi w środowisku Python.

Ćwiczenia laboratoryjne: Wykonanie zadań podanych przez prowadzącego (opracowanie algorytmu rozwiązującego dany problem oraz jego implementacja i uruchomienie w języku Python).

Literatura

Podstawowa:

Qingkai Kong, Timmy Siau, Alexandre Bayen, Python Programming and Numerical Methods - A Guide for Engineers and Scientists (online)
Robert Johansson, Matematyczny Python. Obliczenia naukowe i analiza danych z użyciem NumPy, SciPy i Matplotlib, Helion 2021
Mark Lutz, Python. Wprowadzenie. Wydanie V, Helion 2020

Uzupełniająca:

Piotr Wróblewski, Algorytmy w Pythonie. Techniki programowania dla praktyków, Helion 2022

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	150	6,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	75	3,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	75	3,00