



Własności algorytmów numerycznych Sylabus zajęć

Informacje podstawowe

Kierunek studiów Technologie informatyczne	Cykl dydaktyczny 2024/25	
Specjalność -	Kod zajęć 17TINS.310N.07270.24	
Jednostka organizacyjna Nadnotecki Instytut UAM w Pile	Języki wykładowe Polski	
Poziom studiów studia inżynierskie pierwszego stopnia	Obligatoryjność Obowiązkowy/fakultatywny	
Forma studiów studia stacjonarne	Blok zajęciowy Przedmioty nieprzypisane	
Profil studiów profil praktyczny		
Koordinator zajęć	Andrzej Maćkiewicz	
Prowadzący zajęcia	Andrzej Maćkiewicz	
Okres Semestr 5	Forma zajęć / liczba godzin / forma zaliczenia • Laboratorium: 30, Zaliczenie z oceną	Liczba punktów ECTS 2

Cele kształcenia dla zajęć

Kod	Cel
C1	Przedmiot opisuje analizę numeryczną jako dyscyplinę służącą do konstrukcji i analizy efektywnych algorytmów dyskretnych do rozwiązywania ciągłych problemów analizy matematycznej przy użyciu dużej liczby danych. Efektywność oznacza tu oszczędne wykorzystanie „zasobów” takich jak czas obliczeń i wykorzystanie pamięci maszyny. Szczególną uwagę przywiązuje się zatem do takich pojęć jak: złożoność obliczeniowa, uwarunkowanie problemu, wpływ błędów zaokrągleń, stabilność i numeryczna poprawność algorytmów. Uzasadniony teoretycznie materiał jest ilustrowany przykładami i problemami przeznaczonymi do samodzielnego rozpatrzenia.

Wymagania wstępne

Analiza matematyczna I.
Algorytmy algebry liniowej.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	Zna zagadnienia matematyczne konieczne do zrozumienia podstawowych pojęć i zjawisk niezbędnych w pracy informatyka obejmujące m.in. podstawy analizy matematycznej, przybliżone metody opisu zjawisk ciągłych, metody numeryczne, podstawy algebry i algebry liniowej, podstawy logiki i matematyki dyskretnej.	TIN_K3_W01, TIN_K3_W03_inz, TIN_K3_W04_inz, TIN_K3_W06_inz, TIN_K3_W14_inz	Test, Projekt, Praca pisemna
W2	Zna i rozumie podstawowe metody projektowania, analizowania i programowania algorytmów (projektowanie strukturalne, rekurencja, metoda dziel i zwyciężaj, poprawność, metoda niezmienników, złożoność obliczeniowa)	TIN_K3_W01, TIN_K3_W03_inz, TIN_K3_W04_inz, TIN_K3_W05_inz, TIN_K3_W08_inz, TIN_K3_W09_inz	Test, Projekt, Praca pisemna
Umiejętności - Student/ka:			
U1	Potrafi projektować, analizować pod kątem poprawności i złożoności obliczeniowej oraz programować algorytmy; wykorzystywać podstawowe techniki algorytmiczne i struktury danych	TIN_K3_U01, TIN_K3_U02, TIN_K3_U05_inz	Test, Projekt, Praca pisemna
U2	Potrafi stosować techniki prowadzące do otrzymania oprogramowania wysokiej jakości .	TIN_K3_U01, TIN_K3_U02, TIN_K3_U03, TIN_K3_U04, TIN_K3_U34, TIN_K3_U35	Test, Projekt, Praca pisemna
U3	Potrafi opracować, przeanalizować i zaimplementować wybrane metody numeryczne z wykorzystaniem pakietów i bibliotek numerycznych	TIN_K3_U01, TIN_K3_U02, TIN_K3_U05_inz, TIN_K3_U34, TIN_K3_U35	Test, Projekt, Praca pisemna
U4	Potrafi przygotowywać dokumentację, opracowania i raporty w języku polskim i języku angielskim, z wykorzystaniem podstawowych ujęć teoretycznych, korzystając z różnych źródeł .	TIN_K3_U01, TIN_K3_U02, TIN_K3_U05_inz, TIN_K3_U35	Test, Projekt, Praca pisemna

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć

1.	<p>Przyczyny powstawania błędów w obliczeniach zmiennoprzecinkowych</p> <p>Reprezentacja 64-bitowych liczb zmiennopozycyjnych w standardzie IEEE.</p> <p>Podstawowe funkcje Matlaba i Pythona dotyczące arytmetyki zmiennopozycyjnej .</p> <p>Różne rodzaje błędów w obliczeniach numerycznych.</p> <p>Katastroficzna utrata cyfr znaczących.</p> <p>Przybliżanie wartości pochodnych funkcji w punkcie za pomocą ilorazów różnicowych</p> <p>Inne przykłady w których występuje utrata cyfr znaczących.</p>	W1, W2, U1, U2, U3, U4	Laboratorium
2.	<p>. Elementy teorii interpolacji wielomianowej</p> <p>O wyznaczniku macierzy Vandermonde'a.</p> <p>Istnienie i jednoznaczność wielomianu interpolacyjnego Lagrange'a.</p> <p>Wielomiany bazowe Lagrange'a.</p> <p>Błąd metody interpolacji wielomianowej.</p> <p>Postać barycentryczna wielomianu Legendre'a.</p> <p>Postać Newtona wielomianu interpolacyjnego.</p> <p>Algorytmy specjalne rozwiązywania układów równan liniowych o macierzy Vandermonde'a.</p> <p>Dyskretna transformacja Fouriera (DFT).</p>	W1, W2, U1, U2, U3, U4	Laboratorium
3.	<p>. Błąd obcięcia (metody)</p> <p>Błąd bezwzględny i względny.</p> <p>Notacja "dużego O" i "małego o" .</p> <p>Podstawowe wzory różnicowe.</p> <p>O Twierdzeniu Taylora .</p> <p>Błąd obcięcia dla ilorazów różnicowych .</p> <p>Wyprowadzanie wzorów różnicowych różnych rzędów dokładności.</p> <p>Iteracyjne rozwiązywanie skalarnych równań nieliniowych- metoda bisekcji, metoda Newtona, metoda Brenta. Wnioski z obliczeń numerycznych.</p> <p>Rząd zbieżności metod iteracyjnych.</p>	W1, W2, U1, U2, U3, U4	Laboratorium

4.	<p>. O metodzie ekstrapolacji Richardsona</p> <p>Geneza metod ekstrapolacyjnych.</p> <p>Metoda Richardsona.</p> <p>Zastosowanie metody Richardsona do różniczkowania numerycznego.</p>	W1, W2, U1, U2, U3, U4	Laboratorium
5.	<p>O splocie liniowym dwu wektorów .</p> <p>O splocie liniowym dwu wektorów.</p> <p>Szeregi czasowe.</p> <p>Mnożenie długich liczb całkowitych.</p> <p>Splot cykliczny a splot liniowy.</p> <p>Twierdzenie o splocie. Informacja o możliwości wykorzystania FFT do szybkiego wyliczenia splotu</p>	W1, W2, U1, U2, U3, U4	Laboratorium
6.	<p>. Wybrane metody całkowania numerycznego</p> <p>O metodzie trapezów.</p> <p>Złożony wzór trapezów.</p> <p>Rekurencyjna metoda trapezów .</p> <p>Złożona metoda trapezów a dyskretne szeregi Fouriera otrzymywane metodą FFT.</p> <p>Stabilność złożonych metod całkowania numerycznego .</p> <p>Metody adaptacyjne całkowania numerycznego.</p> <p>Metoda ekstrapolacyjna Romberga .</p>	W1, W2, U1, U2, U3, U4	Laboratorium
7.	<p>Metody jednokrokowe rozwiązywania zadań początkowych dla równań różniczkowych zwyczajnych (ODE).</p> <p>Ekstrapolacyjny algorytm Bulirsha-Stoera dla ODE.</p> <p>Metoda punktu środkowego .</p> <p>Algorytm Bulirsha-Stoera. Przykłady rozwiązań układów sztywnych.</p>	W1, W2, U1, U2, U3, U4	Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Laboratorium	Wykład konwersatoryjny, Metoda analizy przypadków, Metoda laboratoryjna, Pokaz i obserwacja, Rozwiązywanie zadań obliczeniowych, Rozwiązywanie zadań praktycznych

Forma zajęć	Warunki zaliczenia zajęć
Laboratorium	Projekty 30% Własna biblioteka programów przygotowana podczas zajęć 30 % Sprawdzian zaliczeniowy 40% Warunkiem zaliczenia jest uzyskanie min 60% punktów i obecność na co najmniej 60% zajęć.

Literatura

Obowiązkowa

1. – R.L. Burden, J.D. Faires “Numerical Analysis 9ed.”, Cengage Learning 2012.
2. – M. L. Overton, “Numerical Computing with IEEE Floating Point Arithmetic”, SIAM 2001.

Dodatkowa

1. – J. Stoer i R. Bulirsch, „Wstęp do analizy numerycznej”, PWN, 1982.

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Laboratorium	30
Przygotowanie projektu	30
Łączny nakład pracy studenta	Liczba godzin 60
Liczba punktów ECTS	ECTS 2

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
TIN_K3_U01	Absolwent/ka potrafi zastosować wiedzę matematyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z informatyką oraz do rozwiązywania problemów praktycznych
TIN_K3_U02	Absolwent/ka potrafi pozyskiwać informacje z literatury, baz wiedzy, Internetu oraz innych wiarygodnych źródeł, integrować je, dokonywać ich interpretacji oraz wyciągać wnioski i formułować opinie
TIN_K3_U03	Absolwent/ka potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów
TIN_K3_U04	Absolwent/ka potrafi w sposób przystępny przedstawić podstawowe fakty z zakresu informatyki, porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach, w tym w języku angielskim oraz z wykorzystaniem narzędzi informatycznych
TIN_K3_U05_inz	Absolwent/ka potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym
TIN_K3_U34	Absolwent/ka potrafi utworzyć opracowanie przedstawiające określony problem z zakresu informatyki, w tym z wykorzystaniem współczesnych metod prezentacyjnych
TIN_K3_U35	Absolwent/ka potrafi przygotowywać dokumentację, opracowania i raporty w języku polskim i języku obcym, w tym z wykorzystaniem podstawowych ujęć teoretycznych, a także różnych źródeł
TIN_K3_W01	Absolwent/ka zna i rozumie zagadnienia matematyczne konieczne do zrozumienia podstawowych pojęć i zjawisk niezbędnych w pracy informatyka obejmujące m.in. podstawy analizy matematycznej, przybliżone metody opisu zjawisk ciągłych, metody numeryczne, podstawy algebry i algebry liniowej, podstawy logiki i matematyki dyskretnej, metody probabilistyczne oraz statystykę
TIN_K3_W03_inz	Absolwent/ka zna i rozumie narzędzia, technologie i urządzenia informatyczne właściwe dla wybranych obszarów zastosowań oraz rozumie podstawy ich działania
TIN_K3_W04_inz	Absolwent/ka zna i rozumie podstawowe konstrukcje programistyczne (przypisanie, instrukcje sterujące, wywoływanie podprogramów i przekazywanie parametrów) oraz pojęcia składni i semantyki języków programowania
TIN_K3_W05_inz	Absolwent/ka zna i rozumie podstawowe metody projektowania, analizowania i programowania algorytmów (projektowanie strukturalne, rekurencja, metoda dziel i zwyciężaj, programowanie z nawrotami, poprawność, metoda niezmienników, złożoność obliczeniowa)
TIN_K3_W06_inz	Absolwent/ka zna i rozumie podstawowe struktury danych i wykonywane na nich operacje (reprezentacja danych liczbowych, arytmetyka i błędy zaokrągleń, tablice, napisy, zbiory, rekordy, pliki, wskaźniki i referencje, struktury wskaźnikowe, listy, stosy, kolejki, drzewa i grafy)
TIN_K3_W08_inz	Absolwent/ka zna i rozumie problemy inżynierii oprogramowania, w tym projektowania (wzorce projektowe, architektura oprogramowania, analiza i projektowanie obiektowe), wykorzystania API, narzędzi i środowisk wytwarzania oprogramowania (narzędzia do analizy wymagań i modelowania)
TIN_K3_W09_inz	Absolwent/ka zna i rozumie architekturę współczesnych systemów (logika układów cyfrowych i reprezentacja danych, architektura procesora, wejście-wyjście, pamięć, architektury wieloprocessorowe)
TIN_K3_W14_inz	Absolwent/ka zna i rozumie podstawowe metody sztucznej inteligencji