

Metody obliczeniowe w nauce i technice

Labolatorium 2

Interpolacja

Mateusz Kowalski

1. Wstęp

Na poprzednim laboratorium zajmowaliśmy się zagadnienie interpolacji, badaniem jej błędu oraz pomiaru czasu wykonania różnych metod.

2. Metoda Gaussa-Jordana

Zaimplementowałem metodę Gaussa-Jordana z częściowym poszukiwaniem. Poniżej znajdują się wyniki pomiaru czasu dla $N = 500, 1000, \dots, 1400$ dla 3 różnych metod. Jak możemy zauważyć na podstawie wyników poniżej, możemy śmiało stwierdzić że metody wbudowane takie jak *linalg.solve* są najszybsze.,

N	500	600	700	800	900	1000	1100	1200	1300	1400
Gauss	0.833	1.330	1.912	2.577	3.313	4.206	5.246	6.461	7.762	9.194
Regular	0.033	0.011	0.071	0.025	0.016	0.020	0.101	0.046	0.177	0.074
Least Squares	0.002	0.003	0.009	0.017	0.008	0.014	0.029	0.035	0.046	0.032

3. Faktoryzacja LU

Funkcja faktoryzacji została poprawnie zaimplementowana. Na dowód tego wielokrotne testy $\|A-LU\|$ przechodzą bez żadnych problemów. Funkcja *in-situ* (w miejscu).

4. Analiza obwodu elektrycznego - nadokreślony układ równań

Użyłem networkx oraz matplotlib jako narzędzia do wizualizacji grafów, a do rozwiązywania układów równań użyłem numpy. Zaimplementowałem wczytywanie grafu z pliku oraz wszystkie metody generacji grafów w podpunkcie d. Wykorzystałem pierwsze oraz drugie prawo Kirchhoffa do utworzenia i rozwiązywania równań. Do wyznaczenia cykli prostych użyłem wbudowanej funkcji w networkx `cycle_basis`. Kierunek przepływu prądu był ustalany poprzez założenie na podstawie 1 odkrytego cyklu prostego. Później cykle przyległe do wcześniej wspomnianego dopasowywały kierunek do wcześniej już ustalonego. Metoda potencjałów węzłowych również została zaimplementowana. (niestety z powodu braku czasu, dla grafów wygenerowanych przez metodę potencjałów węzłowych, nie działa dla nich rysowanie grafu skierowanego. Zostały przygotowane grafy w programie dla n liczby wierzchołków $n = 15, 30, \dots, 200$).