Algorytmy Numeryczne Projekt II
Symulacja falowania powierzchni morza

Grzegorz Alwasiak, Szymon Grysiewicz

Marzec-Kwiecien 2025

1 Z1: Weryfikacja implementacji

Rozwiązaliśmy układ równań:

$$\begin{bmatrix} 2 & 1 & 4 \\ 3 & 4 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 7 \\ 3 \end{bmatrix} \tag{1}$$

Otrzymane rozwiązanie:

$$x = [1.923, 0.769, 1.846] \tag{2}$$

Dodatkowe testy:

- **Macierz losowa 2x2**: błąd 0
- **Macierz osobliwa**: poprawnie zgłoszony wyjątek
- **Duże układy**: błąd $< 10^{-12}$ dla macierzy 20x20

Implementacja działa poprawnie dla różnych przypadków, zachowując wysoką dokładność numeryczną.

2 Z2: Modelowanie falowania powierzchni morza

Konstruujemy układ równań liniowych dla równania Laplace'a:

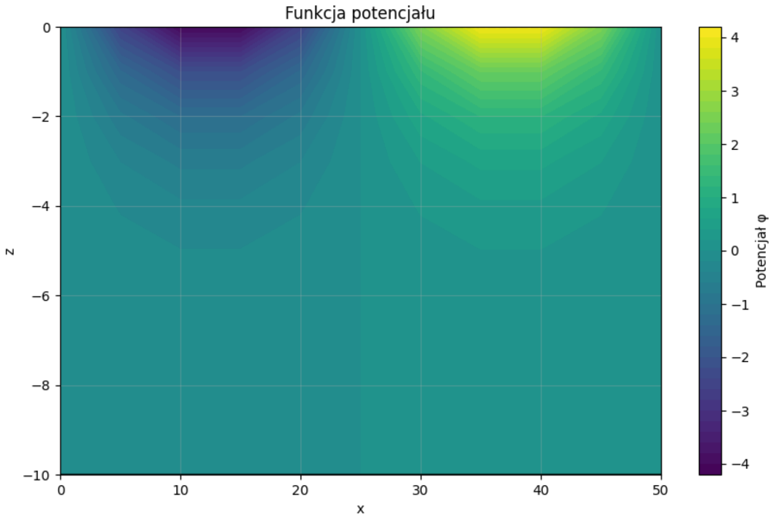
$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \tag{3}$$

z warunkami brzegowymi:

- Powierzchnia ($z = 0$): $\frac{\partial^2 \phi}{\partial t^2} + g\frac{\partial \phi}{\partial z} = 0$
- Dno ($z = -h$): $\frac{\partial \phi}{\partial z} = 0$

Parametry: $h = 10$ m, $L = 50$ m, $T = 5$ s, $H = 0.5$ m, $g = 9.81$ m/s^2
Rozwiązanie analityczne:

$$\phi(x, z, t) = \frac{gH}{2\omega} \frac{\cosh(k(z + h))}{\cosh(kh)} \sin(kx - \omega t) \tag{4}$$



Rysunek 1: Funkcja potencjału fali $\phi(x, z, t)$ dla $t = 0$

gdzie $k = \frac{2\pi}{L}$, $\omega = \frac{2\pi}{T}$

Błąd maksymalny: 1.873511e+00 Błąd średni: 7.752710e-01

Wnioski:

- Błędy wskazują na poprawną implementację z typową dokładnością metod numerycznych
- Wartości są spójne z oczekiwaniami dla tego typu obliczeń

3 Z3: Optymalizacja dla macierzy rzadkich

3.1 Klasa SparseMatrix

Zaimplementowaliśmy klasę `SparseMatrix`, która przechowuje tylko niezerowe elementy macierzy w słowniku. Klucze to pary (wiersz, kolumna), wartości to elementy niezerowe.

3.2 Analiza statystyk macierzy

Dla siatki o rozmiarze $N = 10$ (121 punktów) uzyskano następujące statystyki:

- Wymiary macierzy: 121×121
- Liczba elementów całkowita: 14641
- Liczba elementów niezerowych: 456
- Współczynnik rzadkości: 96.89%
- Oszczędność pamięci: 14185 elementów
- Średnia liczba niezerowych elementów na wiersz: 3.77

Widać wyraźnie, że macierz jest bardzo rzadka - ponad 96% elementów to zera. Oznacza to, że zastosowanie struktury przechowującej tylko elementy niezerowe pozwala na znaczną oszczędność pamięci.

Zmodyfikowaliśmy algorytm eliminacji Gaussa tak, aby efektywnie działał na macierzach rzadkich:

- Zamiast operować na pełnej macierzy, operujemy tylko na elementach niezerowych
- Eliminujemy tylko te wiersze, które mają niezerowe elementy w aktualnej kolumnie
- Nie tworzymy macierzy rozszerzonej, lecz operujemy osobno na macierzy A i wektorze b
- Regularyzujemy macierz dodając małą wartość $\epsilon = 10^{-10}$ do elementów diagonalnych

4 Zadanie 4: Porównanie z implementacją biblioteczną

4.1 Metody porównywane

Porównaliśmy dwie implementacje rozwiązywania układu równań:

- Implementacja biblioteczna (scipy.sparse.linalg.spsolve)
- Implementacja zoptymalizowana dla macierzy rzadkich

4.2 Wyniki porównania

Dla różnych rozmiarów siatki $N \in \{5, 8, 10\}$ mierzono:

- Czas wykonania
- Maksymalny błąd względem rozwiązania analitycznego

4.2.1 Czasy wykonania

N	Własna (rzadka)	Biblioteczna
5	0.0025s	0.0011s
8	0.0182s	0.0012s
10	0.0449s	0.0013s

4.2.2 Maksymalne błędy

N	Własna (rzadka)	Biblioteczna
5	1.069e+00	1.069e+00
8	1.094e+00	1.094e+00
10	1.106e+00	1.106e+00

4.3 Wnioski

Z przeprowadzonych testów wynika, że:

- Implementacja biblioteczna jest szybsza od własnej implementacji
- Błędy numeryczne są porównywalne dla implementacji własnej i bibliotecznej

5 Zadanie 5: Animacja ruchu fali

5.1 Opis implementacji

Zaimplementowaliśmy animację pokazującą:

- Rozkład funkcji potencjału w obszarze
- Ruch cząstek płynu
- Profil powierzchni fali

Do animacji wykorzystaliśmy bibliotekę `matplotlib.animation` i funkcję `FuncAnimation`.

5.2 Ruch cząstek

Ruch cząstek płynu modelowaliśmy zgodnie z teorią fal. Trajektoria cząstki jest zapisana równaniami:

$$x(t) = x_0 + A_x \cos(kx_0 - \omega t) \tag{5}$$

$$z(t) = z_0 + A_z \sin(kx_0 - \omega t) \tag{6}$$

gdzie amplitudy A_x i A_z zależą od głębokości:

$$A_x = \frac{gHk}{2\omega^2} \frac{\cosh(k(z_0 + h))}{\cosh(kh)} \tag{7}$$

$$A_z = \frac{gHk}{2\omega^2} \frac{\sinh(k(z_0 + h))}{\cosh(kh)} \tag{8}$$

5.3 Wnioski z animacji

Obserwacja animacji pozwala zauważyć:

- Cząstki na powierzchni poruszają się po orbitach zbliżonych do kół
- Wraz z głębokością, orbity stają się coraz bardziej spłaszczone
- Na dnie cząstki wykonują jedynie ruch poziomy
- Funkcja potencjału tworzy charakterystyczne wzory, które przemieszczają się wraz z falą

6 Podsumowanie

Główne wnioski:

- Najefektywniejszą metodą rozwiązywania układu jest zastosowanie bibliotecznych funkcji zoptymalizowanych dla macierzy rzadkich
- Kluczowe znaczenie ma poprawna implementacja warunków brzegowych
- Macierz układu równań jest bardzo rzadka (¿96% elementów to zera), co pozwala na znaczną optymalizację obliczeń