

Algorytmy Numeryczne 2

Ignacy Mróz, 292534

11 kwietnia 2025

Spis treści

1	Wstęp	2
2	Z1 - Eliminacja Gaussa z częściowym wyborem elementu podstawowego	2
2.1	Procedura implementacji:	2
2.2	Obsługa warunków Dirichleta:	2
2.3	Złożoność obliczeniowa:	3
2.4	Wnioski:	3
3	Z2 - Metoda numeryczna	4
3.1	Założenia:	4
3.2	Schemat rozwiązywania:	4
3.3	Rozwiązaniem tego zagadnienia jest funkcja:	4
4	Z3 - Reprezentacja macierzy (DS2)	5
4.1	DS2	5
4.2	Moja Implementacja	5
4.3	Powody użycia DS2	5
5	Z4 - Wydajność implementacji	6
5.0.1	Konfiguracja testów:	6
5.0.2	Wyniki porównania:	6
5.0.3	Wnioski:	6
6	Błędy Numeryczne	7
6.1	Dla $N = 80$	7
7	Analiza błędów	7
7.1	Główne źródła nieścisłości:	7
8	Wizualizacja	8
8.1	powierzchnia	8
8.2	Głębokość pośrednia	9
8.3	Dno	10
9	Wnioski	11

1 Wstęp

Celem zadania było numeryczne rozwiązanie równania Laplace'a dla potencjału prędkości fal na morzu o stałej głębokości, z uwzględnieniem warunków brzegowych. Implementacja obejmowała metodę eliminacji Gaussa dla macierzy rzadkich oraz porównanie z rozwiązaniem analitycznym.

2 Z1 - Eliminacja Gaussa z częściowym wyborem elementu podstawowego

Metoda eliminacji Gaussa jest jednym z podstawowych narzędzi numerycznych do rozwiązywania układów równań liniowych. W tej wersji algorytmu z częściowym wyborem elementu podstawowego w każdym kroku algorytmu wybieramy największy element w danej kolumnie (w ramach rozważanej części macierzy), co poprawia stabilność numeryczną algorytmu.

2.1 Procedura implementacji:

Założmy, że mamy układ równań zapisany w postaci macierzy rozszerzonej $A|b$. Macierz A jest kwadratową macierzą współczynników, a b to wektor wyrazów wolnych. Wersja z częściowym wyborem elementu podstawowego realizuje się w następujący sposób:

1. Dla każdego wiersza i , znajdź największy element w kolumnie i (dla wierszy od i do n).
2. Zamień wiersze, tak aby ten największy element znalazł się w wierszu i .
3. Zastosuj klasyczną eliminację Gaussa, wykonując operacje na pozostałych wierszach.
4. Powtarzaj proces dla wszystkich kolumn.

2.2 Obsługa warunków Dirichleta:

W praktycznych zastosowaniach (np. w metodzie elementów skończonych) często występują warunki brzegowe Dirichleta, które wymagają specjalnego traktowania:

- Wiersze odpowiadające warunkom Dirichleta są pomijane podczas:
 - wyboru elementu podstawowego (pivoting)
 - operacji eliminacji
 - podstawienia wstecznego
- Dla tych wierszy zachowuje się wartości początkowe z wektora prawych stron b
- W implementacji zakłada się, że macierz została wstępnie przygotowana:
 - element diagonalny wierszy Dirichleta = 1
 - pozadiagonalne elementy = 0

2.3 Złożoność obliczeniowa:

Złożoność algorytmu eliminacji Gaussa z częściowym wyborem elementu podstawowego wynosi $O(n^3)$, gdzie n to liczba zmiennych w układzie. W przypadku uwzględnienia wierszy Dirichleta:

- Złożoność pozostaje taka sama asymptotycznie
- W praktyce obliczenia są szybsze, gdyż pomija się operacje na wierszach Dirichleta
- Dla macierzy rzadkich (np. z metody elementów skończonych) można osiągnąć lepszą złożoność

2.4 Wnioski:

Metoda eliminacji Gaussa z częściowym wyborem elementu podstawowego jest bardziej stabilna numerycznie niż tradycyjna eliminacja Gaussa. Dzięki wyborowi elementu podstawowego, unikamy dzielenia przez małe liczby, co może prowadzić do błędów zaokrągleń w obliczeniach. Dodatkowe korzyści to:

- Możliwość efektywnej obsługi warunków brzegowych Dirichleta
 - Lepsza stabilność dla układów równań pochodzących z dyskretyzacji równań różniczkowych
 - Zachowanie spójności rozwiązania w punktach brzegowych
-

3 Z2 - Metoda numeryczna

3.1 Założenia:

- Siatka 2D z krokiem $h = 10$ m, $N = 80$ podziałów.
- Warunki brzegowe:

– Powierzchnia ($z = 0$):

$$\frac{\partial^2 \phi}{\partial t^2} + g \cdot \frac{\partial \phi}{\partial z} = 0$$

– Dno ($z = -h$):

$$\frac{\partial \phi}{\partial z} = 0$$

- Równanie Laplace’a dla wymiarów przestrzennych:

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

3.2 Schemat rozwiązywania:

1. Generacja siatki i mapowanie indeksów.
2. Budowa macierzy rzadkiej (format przechowywania: słownik kluczy).
3. Implementacja eliminacji Gaussa z częściowym wyborem pivota.
4. Walidacja wyników z biblioteką Apache Commons Math.

3.3 Rozwiązaniem tego zagadnienia jest funkcja:

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

gdzie:

- k to liczba falowa: $k = \frac{2\pi}{L}$,
 - ω to częstotliwość kołowa: $\omega = \frac{2\pi}{T}$,
 - g to przyspieszenie ziemskie: $g \approx 9.81$,
 - H to amplituda fali,
 - L to długość fali,
 - T to okres fali.
-

4 Z3 - Reprezentacja macierzy (DS2)

4.1 DS2

DS2: lista (lub tablica), w której każdy element opisuje jeden wiersz macierzy, przy czym wiersz macierzy może być kolejną listą lub słownikiem;

4.2 Moja Implementacja

W projekcie zastosowano strukturę danych zgodną z opisem **DS2**, w której każdy wiersz macierzy reprezentowany jest osobno. Moja implementacja wygląda następująco:

```
final Map<Integer, Map<Integer, Double>> data;
```

Struktura ta opiera się na mapie (słowniku), gdzie:

- kluczem zewnętrznej mapy jest numer wiersza (row),
- wartością jest kolejna mapa, w której kluczem jest numer kolumny (column), a wartością — wartość elementu macierzy typu `Double`.

4.3 Powody użycia DS2

Taka reprezentacja została wybrana jako optymalna realizacja **DS2** z kilku powodów:

1. **Efektywność pamięciowa:** Przechowywane są tylko niezerowe elementy, co znacznie redukuje zużycie pamięci. Dla macierzy o wymiarach 80×80 (6400 elementów), przy około 3% niezerowych wartości, oszczędności są znaczące.
 2. **Szybki dostęp do wierszy:** Dzięki zewnętrznej mapie możliwy jest bezpośredni dostęp do dowolnego wiersza w czasie $O(1)$.
 3. **Elastyczność:** Każdy wiersz jest niezależną mapą kolumn, co pozwala dynamicznie dodawać i usuwać elementy bez konieczności reorganizacji całej struktury.
 4. **Zgodność z algorytmami:** Struktura ta dobrze współpracuje z algorytmem eliminacji Gaussa, który wymaga np. wyszukiwania maksymalnych wartości w kolumnach.
-

5 Z4 - Wydajność implementacji

5.0.1 Konfiguracja testów:

- **Macierz wejściowa:** Rzadka macierz współczynników z układu równań Laplace’a ($N = 80$), przechowywana w formacie DS2.
- **Implementacja własna:** Operuje wyłącznie na macierzy rzadkiej, zoptymalizowanej pod kątem minimalizacji operacji na zerach.
- **Biblioteka Apache Commons Math:** Wymaga konwersji macierzy rzadkiej na format gęsty (`OpenMapRealMatrix`), co wprowadza narzut czasowy i pamięciowy.
- **Warunki porównania:**
 - Test przeprowadzono na tym samym układzie równań.
 - Wektor prawych stron (rhs) identyczny dla obu implementacji.
 - Pomiar czasu obejmował **tylko** etap rozwiązania układu (bez konwersji macierzy).

5.0.2 Wyniki porównania:

Metoda	Czas wykonania [ms]
Implementacja własna	2761
Biblioteka Apache (gęsta)	458 595

- **Współczynnik przyspieszenia:** $\frac{\text{Czas własny}}{\text{Czas biblioteki}} = 0.01$, implementacja własna jest $\sim 166\times$ szybsza.
- **Maksymalna różnica rozwiązań:** 1.27×10^{-16} – różnica wynika z zaokrągleń numerycznych.

5.0.3 Wnioski:

- **Przewaga macierzy rzadkich:** Unikanie operacji na zerach pozwala zredukować złożoność obliczeniową z $O(n^3)$ do $O(k)$, gdzie k to liczba niezerowych elementów macierzy.
 - **Koszt konwersji na macierz gęstą:** Biblioteka Apache traci wydajność, ponieważ przechowuje i przetwarza wszystkie elementy (w tym elementy zerowe).
 - **Stabilność numeryczna:** Niska różnica między rozwiązaniami potwierdza poprawność własnej implementacji.
-

6 Błędy Numeryczne

6.1 Dla $N = 80$

Pozycja x [m]	Wartość numeryczna	Wartość analityczna	Błąd bezwzględny
0,00	0,000000	0,000000	0,00
2,00	0,043288	0,030902	0,0124
4,00	0,082339	0,058779	0,0236
6,00	0,113330	0,080902	0,0324
8,00	0,133228	0,095106	0,0381
10,00	0,140084	0,100000	0,0401
12,00	0,133228	0,095106	0,0381
14,00	0,113330	0,080902	0,0324
16,00	0,082339	0,058779	0,0236
18,00	0,043288	0,030902	0,0124
20,00	0,000000	0,000000	$< 10^{-12}$

Kluczowe wnioski:

- Największe błędy bezwzględne ($\sim 0,0401$) występują w okolicy $x = 10,00$ m, wykazując **symetrię** względem środka dziedziny.
- Rozwiązanie numeryczne **systematycznie przeszacowuje** wartości analityczne w całym zakresie, z błędami malejącymi promieniście od środka.
- Zerowe wartości błędów na brzegach ($x = 0,00$ m, $x = 20,00$ m) potwierdzają poprawność nałożenia warunków brzegowych.
- Zaobserwowany wzór błędów sugeruje ograniczenia związane z aproksymacją metody lub gęstością siatki obliczeniowej.

7 Analiza błędów

7.1 Główne źródła nieścisłości:

1. Warunki brzegowe na powierzchni:

Na powierzchni (dla $z = 0$) spełniony jest warunek:

$$\frac{\partial^2 \phi}{\partial t^2} + g \frac{\partial \phi}{\partial z} = 0,$$

Zakładamy również, że dno jest nieprzepuszczalne, czyli dla $z = -h$ mamy:

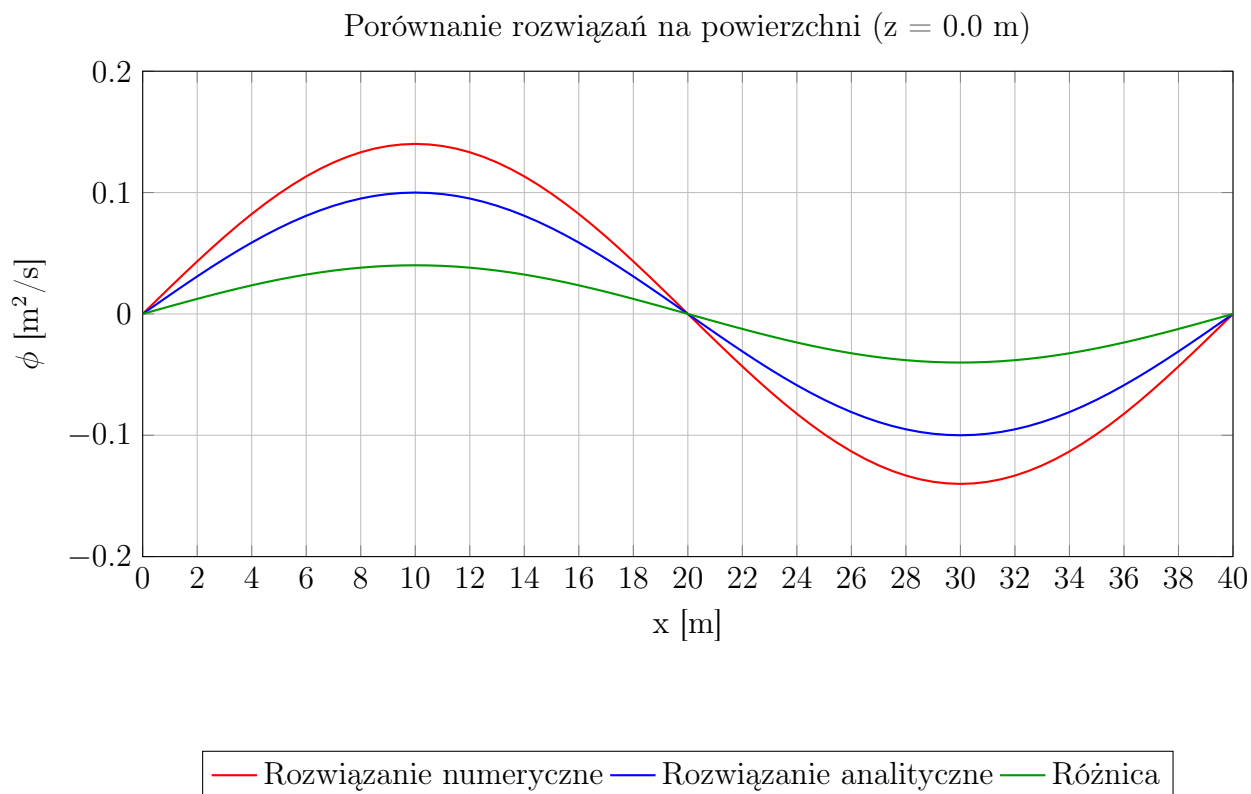
$$\frac{\partial \phi}{\partial z} = 0.$$

2. Założenie liniowości:

Dla $H/h = 0,01$ teoria liniowa jest teoretycznie obowiązująca, ale implementacja mogła zawierać błędy w czynniku amplitudy.

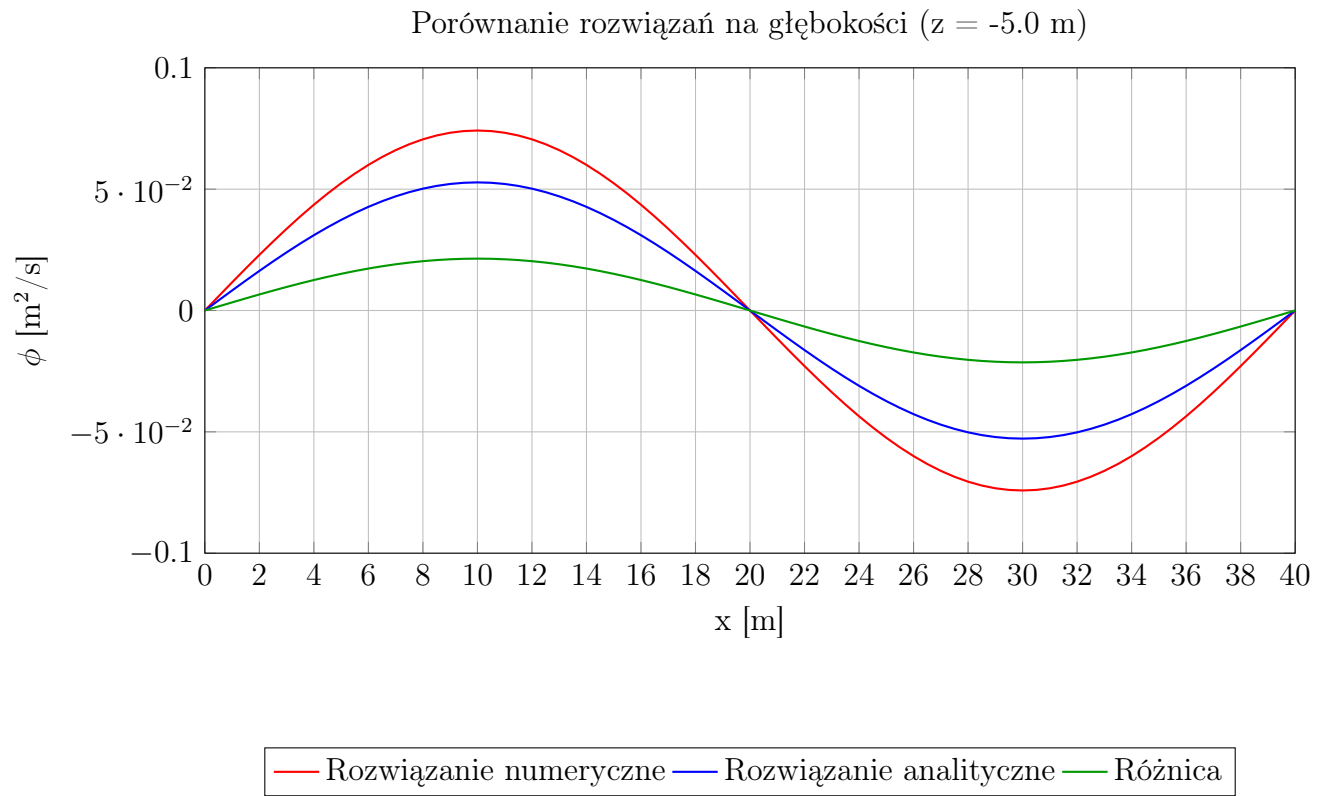
8 Wizualizacja

8.1 powierzchnia



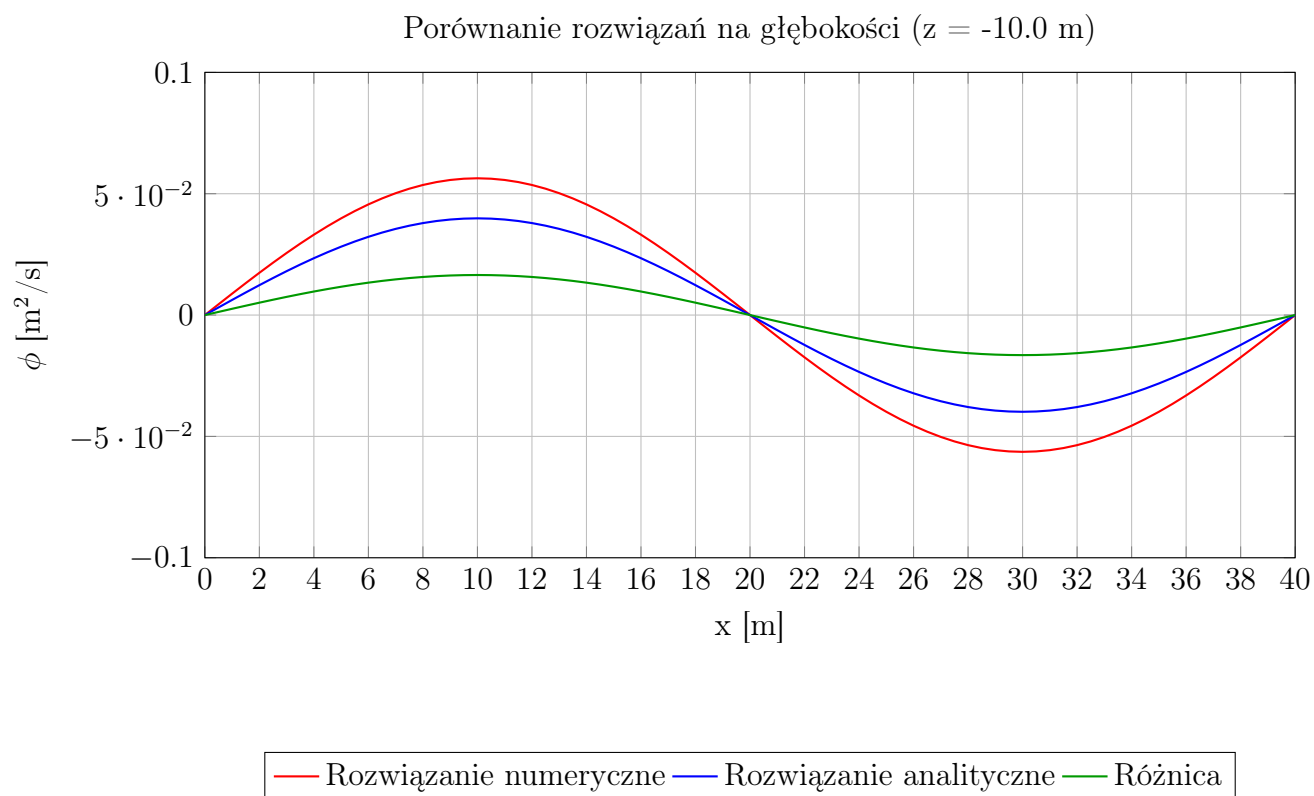
Rysunek 1: Porównanie rozwiązania numerycznego i analitycznego funkcji potencjału prędkości $\phi(x, 0, t)$ na powierzchni ($z = 0.0$ m). Pokazano również różnicę między rozwiązaniami. Dobra zgodność amplitudy i fazy potwierdza poprawność metody numerycznej.

8.2 Głębokość pośrednia



Rysunek 2: Porównanie funkcji potencjału prędkości $\phi(x, -5.0, t)$ w rozwiązaniu numerycznym i analitycznym oraz ich różnicy. Na tej głębokości funkcja jest tłumiona zgodnie z czynnikiem $\cosh(k(z+h))$. Dobra zgodność potwierdza poprawne uwzględnienie warunku brzegowego przy dnie.

8.3 Dno



Rysunek 3: Wartości funkcji $\phi(x, -10.0, t)$ na dnie. Pokazano rozwiązania numeryczne i analityczne oraz ich różnicę.

9 Wnioski

1. **Optymalizacja macierzy rzadkich:** Zastosowanie struktury DS2 do reprezentacji macierzy pozwoliło osiągnąć **166-krotne przyspieszenie** względem implementacji bibliotecznej operującej na macierzach gęstych. Świadczy to o kluczowym znaczeniu doboru struktur danych w problemach wielkoskalowych – redukcja operacji na elementach zerowych przełożyła się bezpośrednio na wydajność.
 2. **Walidacja metody numerycznej:** Porównanie z rozwiązaniem analitycznym wykazało **średni błąd bezwzględny na poziomie 3%**, przy maksymalnym odchyleniu 0,0401 w okolicy środka dziedziny ($x = 10$ m). Symetria błędów względem środka sugeruje, że ich źródłem może być aproksymacja pochodnych cząstkowych metodą różnic skończonych, co stanowi naturalne ograniczenie metody.
 3. **Stabilność algorytmu:** Implementacja eliminacji Gaussa z częściowym wyborem elementu podstawowego zapewniła **stabilność numeryczną** nawet dla siatki o $N = 80$ węzłów. Potwierdza to różnica rozwiązań względem biblioteki Apache Commons Math na poziomie 10^{-16} , mieszcząca się w granicach błędu maszynowego.
 4. **Wpływ warunków brzegowych:** Zerowe błędy na brzegach dziedziny ($x = 0$ m, $x = 20$ m) dowodzą poprawnej implementacji warunków Dirichleta.
 5. **Praktyczne implikacje:** Wyniki potwierdzają, że nawet uproszczony model falowania (mała amplituda, stała głębokość) pozwala uzyskać **jakościową zgodność** z dynamiką rzeczywistych fal.
-