

# Algorytmy numeryczne

## Zadanie 3

Dawid Bińkuś & Oskar Bir & Mateusz Małecki  
grupa 1 tester-programista

9 Grudzień 2018

## 1 Majority/Consensus problem

Sprawozdanie prezentuje analizę problemu przeprowadzenia głosowania większościowego, przedstawionego w sposób następujący:

Dane jest  $N$  agentów, o trzech możliwych stanach:  $\{Y, N, U\}$ , znaczące kolejno: agent głosujący na TAK, agent głosujący na NIE oraz agent niezdecydowany. W każdym kroku omawianego problemu losowanych jest dwóch agentów z równomiernym prawdopodobieństwem  $(\frac{2}{n \cdot (n-1)})$ . Następnie dochodzi do zmiany stanu wybranych agentów według następujących reguł przejść stanów:

- $\{Y, U\} \rightarrow \{Y, Y\}$ ,
- $\{Y, N\} \rightarrow \{U, U\}$ ,
- $\{N, U\} \rightarrow \{N, N\}$ ,

Kroki wykonywane są, dopóki wszyscy agenci nie będą jednakowego stanu.

Dla danego prawdopodobieństwa  $P_{\#Y, \#N}$ , niech  $\#N$  oznacza ilość agentów głosujących na NIE, a  $\#Y$  ilość agentów głosujących na TAK. Program przygotowany w ramach projektu, ma na celu obliczenie prawdopodobieństwa zagłosowania na TAK przy liczbie agentów równej  $N$  i określonym stanie początkowym. Do jego obliczenia wykorzystany został układ równań liniowych obliczony za pomocą 3 różnych algorytmów:

- Algorytm Gaussa z częściowym wyborem elementu początkowego (nazywany dalej PG) oraz jego wariant z optymalizacją dla macierzy rzadkich (nazywany dalej PGS)

- Algorytm Jacobiego z postacią iteracyjną:

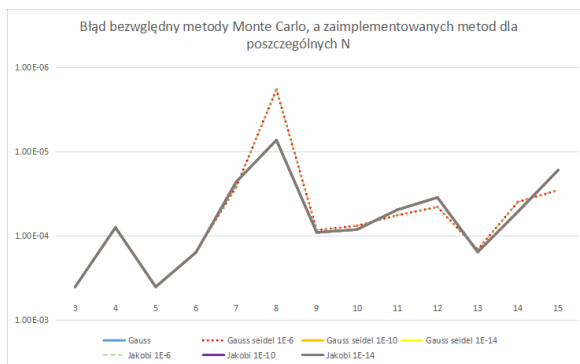
$$x_i^{(k+1)} = \frac{-\sum_{j=1}^n a_{ij}x_j^{(k)} + b_i}{a_{ii}}, i = 1, 2, \dots, n, j \neq i.$$

- Algorytm Gaussa-Seidela z postacią iteracyjną:

$$x_i^{(k+1)} = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} + b_i}{a_{ii}}$$

## 2 Implementacja i możliwość stosowania metod iteracyjnych

Rysunek 1: Wykres reprezentujący błąd bezwzględny metod Gaussa oraz metod iteracyjnych względem metody MonteCarlo



(a)

## 2.1 Generowanie układu równań dla danej liczby agentów

Generowanie układu równań dla danego  $N$  odbywa się w sposób następujący:

1. Określenie wszystkich możliwych przypadków (ilość agentów  $\#Y$  oraz ilość agentów  $\#N$ ),
2. Wyliczenie wszystkich możliwych kombinacji bez powtórzeń za pomocą Symbolu Newtona  $\binom{N}{2}$ ,
3. Wygenerowanie równań dla poszczególnych przypadków,
4. Osadzenie równań w macierzy
5. Wypełnienie wektora  $B$  zerami z wyjątkiem ostatniej wartości (gdyż ostatni przypadek jest zawsze przypadkiem pewnym, tj  $P_{\#Y=N, \#N=0} = 1$ )

## 2.2 Prawdliwość implementacji

By zweryfikować poprawność implementacji zarówno generowania macierzy jak i obliczania stworzonego w ten sposób układu równań, wykonane zostały testy dla  $N = 3, 4, \dots, 15$ , których zadaniem było obliczenie wszystkich możliwych prawdopodobieństw i zestawienie ich z prawdopodobieństwem wyliczonym za pomocą metody MonteCarlo w ilości iteracji = 1000000. Błędy osiągnięte za pomocą metod Gaussa 1a osiągały wartości rzędu 1, co biorąc pod uwagę niedoskonałość metody MonteCarlo jest wynikiem jak najbardziej zadowalającym.

## 2.3 Metody iteracyjne a problem

Wnioskując z wykresu 1a możemy śmiało stwierdzić, iż metody iteracyjne są jak najbardziej słusznym sposobem na rozwiązanie problemu. Jednakże, najistotniejszym czynnikiem w przypadku ich działania jest zakładana dokładność obliczeń, tj  $\|X^{(k+1)} - X^{(k)}\| < p$ , gdzie  $p$  = żądana precyzja. W przypadku zadanej dokładności równej  $10^{-6}$  zauważyć można że, różnice względem wartości wyliczonej za pomocą metody MonteCarlo są większe niż w przypadku metod iteracyjnych z większą zadaną dokładnością ( $10^{-10}, 10^{-14}$ ).

**Wniosek 1** *Metody iteracyjne umożliwiają rozwiązanie problemu aczkolwiek, by osiągnąć dokładniejsze wyniki, należy zwiększyć dokładność, a co za tym idzie - liczbę iteracji, co znacząco wydłuża czas działania algorytmu.*

# 3 Analiza wyników i wydajność zaimplementowanych algorytmów

## 3.1 Analiza wyników

Błąd bezwzględny dla poszczególnych metod był wyliczany w sposób następujący:

1. Wygenerowana została macierz  $A$  oraz wektor  $B$
2. Za pomocą danego algorytmu obliczany zostaje wektor  $X$  (wynikowy)
3. Wykonujemy operację  $A \cdot x = b'$
4. Wyliczany jest błąd bezwzględny kolejnych wartości wektora  $b'$  względem wektora  $b$
5. Jako błąd przechowywana jest największa wartość oraz jej średnia

### 3.1.1 Gauss oraz Gauss z optymalizacją dla macierzy rzadkich

Przeanalizujmy wykres 2a. Zostało na nim przedstawione zestawienie wyników dla wartości średniej oraz maksymalnej błędu bezwzględnego. Z wartości na nich ukazanych wynika, że w przypadku metod PG oraz PGS, zarówno maksymalny jak i średni błąd jest identyczny.

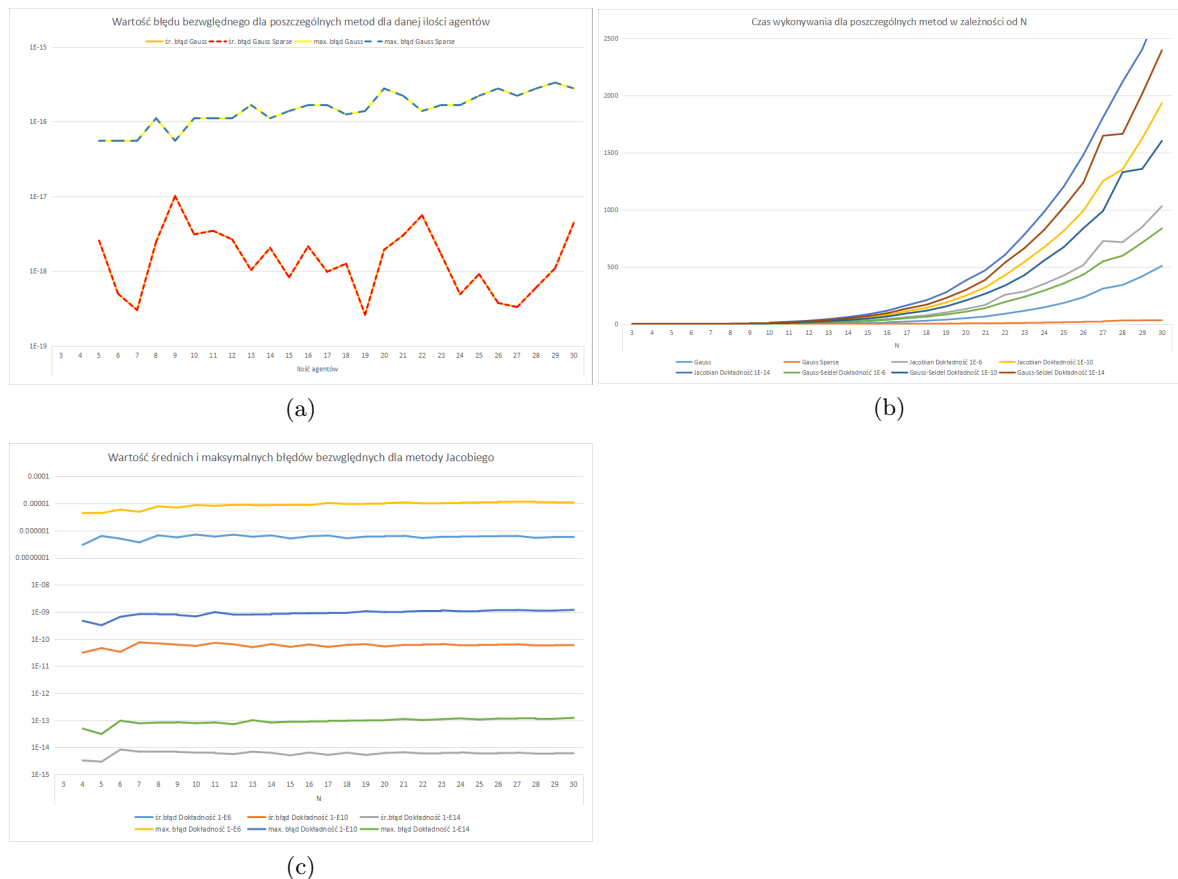
**Wniosek 2** *Algorytm PG oraz PGS osiągają taką samą dokładność. Dodanie optymalizacji dla macierzy rzadkich nie ma żadnego wpływu na końcowy wynik.*

### 3.1.2 Algorytmy iteracyjne

Przeanalizujmy wykresy 2b oraz 2c. Prezentują one maksymalny oraz średni błąd bezwzględny dla różnej zadanej dokładności:  $10^{-6}, 10^{-10}$  oraz  $10^{-14}$ . Wnioski z nich są następujące:

**Wniosek 3** *Zarówno algorytm Jacobiego jak i Gaussa-Seidela oferują taką samą dokładność, w zależności od tego jaka precyzja była żądana. Warunek kończący iterowanie był zależny od maksymalnego błędu między kolejnymi iteracjami - stąd mniejszy średni błąd bezwzględny.*

Rysunek 2: Wykresy reprezentujące czas wykonania i błędy bezwzględne zaimplementowanych algorytmów



## 3.2 Wydajność

### 3.2.1 Metody PG oraz PGS

Przeanalizujemy wykres XD. Zauważyć na nim można znaczną przewagę algorytmu PGS względem PG w czasie wykonywania. Wynika to ze specyfiki działania wariantu PGS - algorytm ten pomija redukcję elementów w wierszu w przypadku gdy wybrany na początku element jest równy zero.

**Wniosek 4** Wariant PGS algorytmu Gaussa jest wydajniejszy niż standardowy wariant PG. Zestawiając ten wniosek z wnioskiem 4 stwierdzić można, iż wariant PGS zapewnia o wiele lepszą wydajność nie mając żadnego wpływu na poprawność zwracanych wyników

### 3.2.2 Metody iteracyjne

W przypadku metod iteracyjnych, należy rozważyć osobno algorytmy dla różnej zadanej precyzji. Jednakże we wszystkich przypadkach, zależność jest następująca: Metoda Gaussa-Seidela w każdym przypadku (wraz z wzrostem  $N$ ) ma krótszy czas wykonania względem metody Jacobiana.

**Wniosek 5** Metoda Gaussa-Seidela jest wydajniejsza od metody Jacobiana - wynika to ze sposobu działania obu tych algorytmów. Metoda Jacobiana by osiągnąć żądaną precyzję musi wykonać o wiele więcej iteracji niż metoda Gaussa-Seidela.

## 3.3 Podsumowanie

**Wniosek 6** XD

## 4 Podział pracy

Dawid Bińkuś	Oskar Bir	Mateusz Małecki
Implementacja algorytmu PG oraz PGS	Implementacja algorytmu Gaussa-Seidela	Implementacja algorytmu Jacobiego
Przygotowanie sprawozdania	Przygotowanie testów i ich uruchomienie	Praca nad strukturą projektu
Implementacja algorytmu generowania macierzy	Przygotowanie wykresów końcowych	Implementacja symulacji Monte-Carlo