Algorytmy numeryczne

Zadanie 3
Dawid Bińkuś & Oskar Bir & Mateusz Małecki
grupa 1 tester-programista

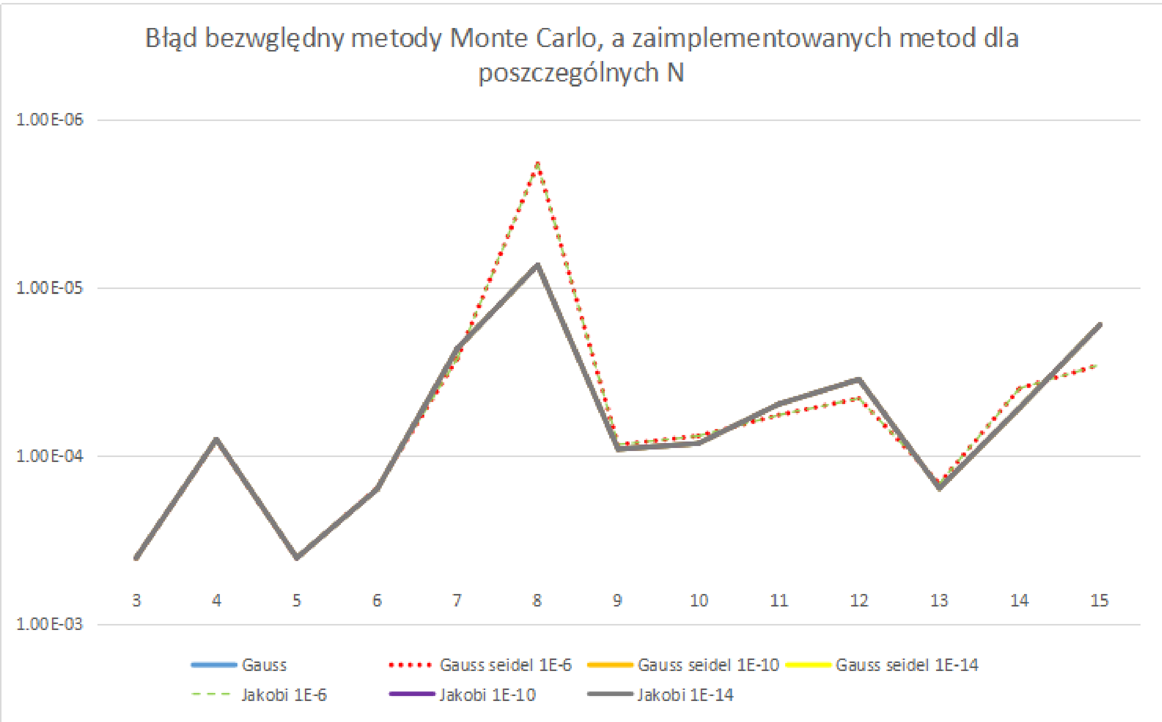
9 Grudzień 2018

1 Majority/Consensus problem

Sprawozdanie prezentuje analizę problemu przeprowadzenia głosowania większościowego. W tym celu, stworzony został program, który generuje układ równań dla danego ustawienia oraz oblicza prawdopodobieństwo zagłosowania na TAK.

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 Prawidłowość implementacji

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

2.3 Metody iteracyjne a problem

W celu zastosowania metod iteracyjnych, wybrane zostały dwie z nich:

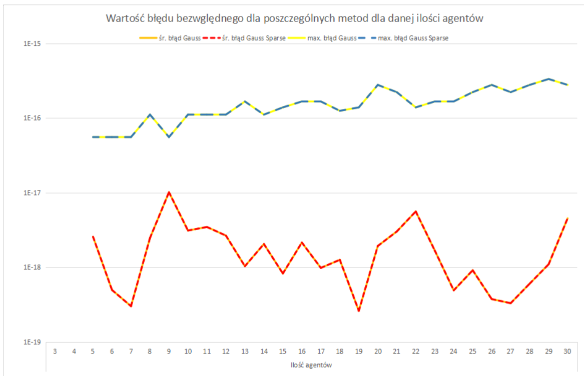
- Algorytm Jacobiego z postacią iteracyjną:
  $x_i{}^{(k+1)} = \frac{-\sum_{j=1}^{n} a_{ij} x_j{}^{k} + b_i}{a_{ii}}, i = 1, 2, ..., n, j \neq i.$
- Algorytm Gaussa-Seidela z postacią iteracyjną:
  $x_i{}^{(k+1)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j{}^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j{}^{(k)} + b_i}{a_{ii}}$

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 $X^{(k+1)} - X^{(k)}$. W przypadku zadanej dokładności równej $1 - E6$ zauważyć można że, różnice względem wartości wyliczonej za pomocą metody MonteCarlo są większe niż w przypadku metod iteracyjnych z większą zadaną dokładnością $(1 - E10, 1 - E14)$.

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

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



(a)

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 Gaussa oraz Gaussa z optymalizacją dla macierzy rzadkich, 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, ponieważ optymalizacja ta, sprawdza przed wyzerowaniem elementu, czy nie jest on już zerem.*

4 Podział pracy

Dawid Bińkuś	Oskar Bir	Mateusz Małecki