

# Sprawozdanie - zadanie 3

Krzysztof Kulewski, 238149, grupa 2, 20.12.2017

## Opis badania

Badanie polegało na obliczeniu prawdopodobieństwa wygranej w grze losowej, w której bierze udział dwóch graczy -  $G_1$  i  $G_2$ , rozgrywanej na planszy o wielkości  $2N + 1$  pól.

Pola ponumerowane są od  $-N$  do  $N$  i stanowią odpowiednio pozycje startowe graczy  $G_2$  i  $G_1$ .

Celem gry jest dotarcie do pola o numerze 0, a grę rozpoczyna gracz  $G_1$ . Kolejne ruchy wykonywane są na przemian.

Symulacja przebiegu gry oraz operacje na macierzach zostały zaimplementowane w języku C#, używając typu *double*. Wyniki zestawiono z efektem działania metod *partialPivLu* (macierze ogólne) oraz *SparseLU* (macierze rzadkie) biblioteki *Eigen3* w C++.

Testy przeprowadzono na komputerze z procesorem Intel i7-6700K z 16 GB RAM.

## Implementacja

Celem obliczenia prawdopodobieństwa wygranej, program najpierw generuje listę wszystkich możliwych *stanów* gry, na podstawie danych wejściowych.

*Stan* to struktura przechowująca informację o pozycjach obu graczy, oraz wartość logiczną, określającą czyja jest *tura*, tj. kto rzuca kostką.

Następnie, każdemu ze *stanów* przyporządkowana jest lista możliwych *przejęć*, wraz z ich prawdopodobieństwami.

Na podstawie uzyskanej listy stanów, tworzony jest układ równań, zapisany w formie macierzy kwadratowej i wektora wyrazów wolnych. Każdy wiersz macierzy opisuje określony stan.

Po rozwiązaniu powyższego układu równań, uzyskujemy wektor z prawdopodobieństwem wygranej w każdym z możliwych stanów.

Zapamiętując indeks stanu początkowego, jesteśmy w stanie odnaleźć odpowiadające mu prawdopodobieństwo w wektorze.

Program rozwiązuje powyższy układ za pomocą metod:

- Gaussa z częściowym wyborem - standardowa i zoptymalizowana dla macierzy rzadkich,
- iteracyjnych - Jacobiego i Gaussa-Seidela,
- z biblioteki *Eigen3* - *PartialPivLU* oraz *SparseLU*.

Odniesieniem podczas liczenia normy wynikowego wektora był wynik metody *SparseLU*.

Wyniki zweryfikowano również metodą Monte-Carlo, symulując przebieg gry 10 000 000 razy.

## Zastosowanie metod iteracyjnych

Macierze uzyskane w zadaniu są przekątniowo dominujące, dzięki czemu do ich obliczenia, możemy stosować metody iteracyjne: Jacobiego oraz Gaussa-Seidela.

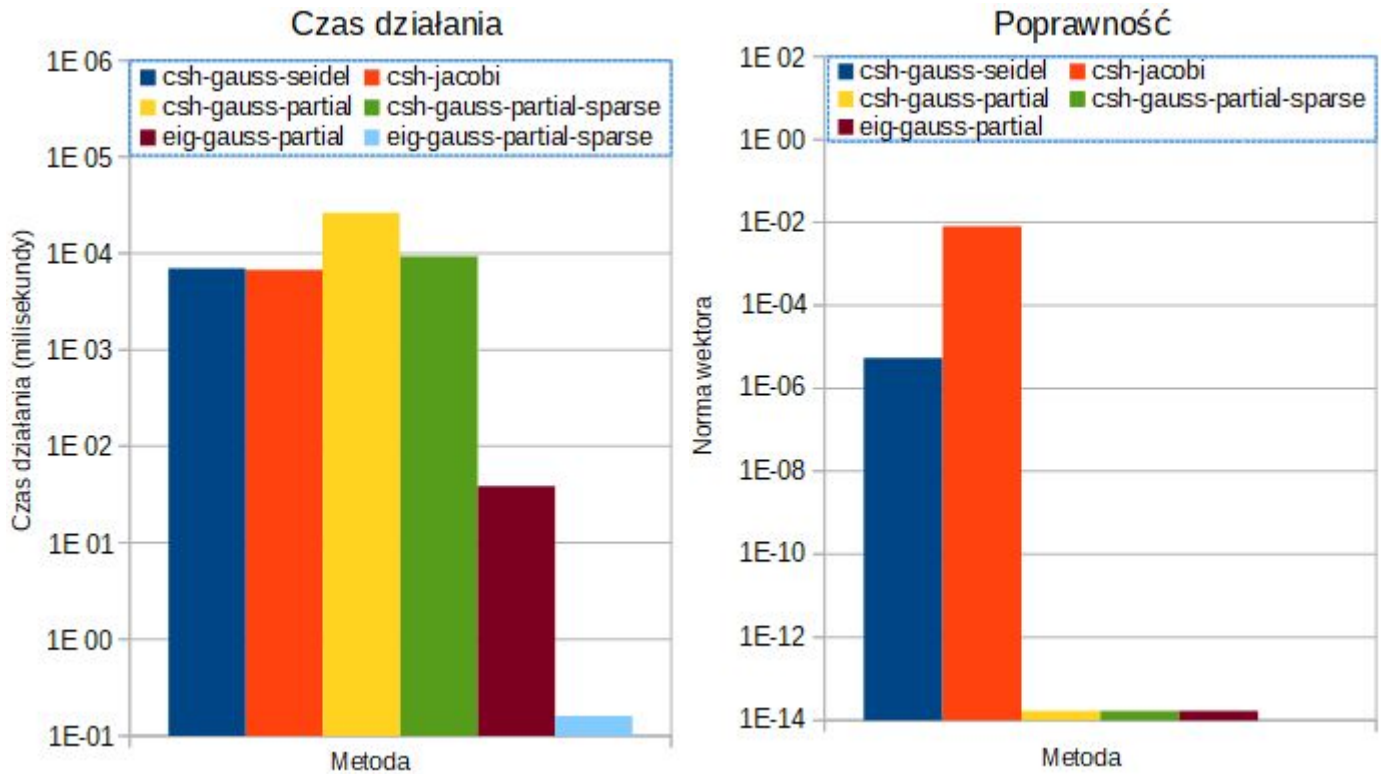
Przydatność tych metod rośnie wraz ze wzrostem macierzy. Charakteryzują się one krótszym czasem działania, ale również mniejszą dokładnością niż metoda Gaussa. Dokładność tę możemy poprawić poprzez zwiększenie liczby iteracji.

Zgodnie z przewidywaniami, metoda Gaussa-Seidela, która wykorzystuje obliczone w aktualnym kroku przybliżenia składowych, daje lepsze przybliżenie, niż metoda Jacobiego dla określonej liczby iteracji.

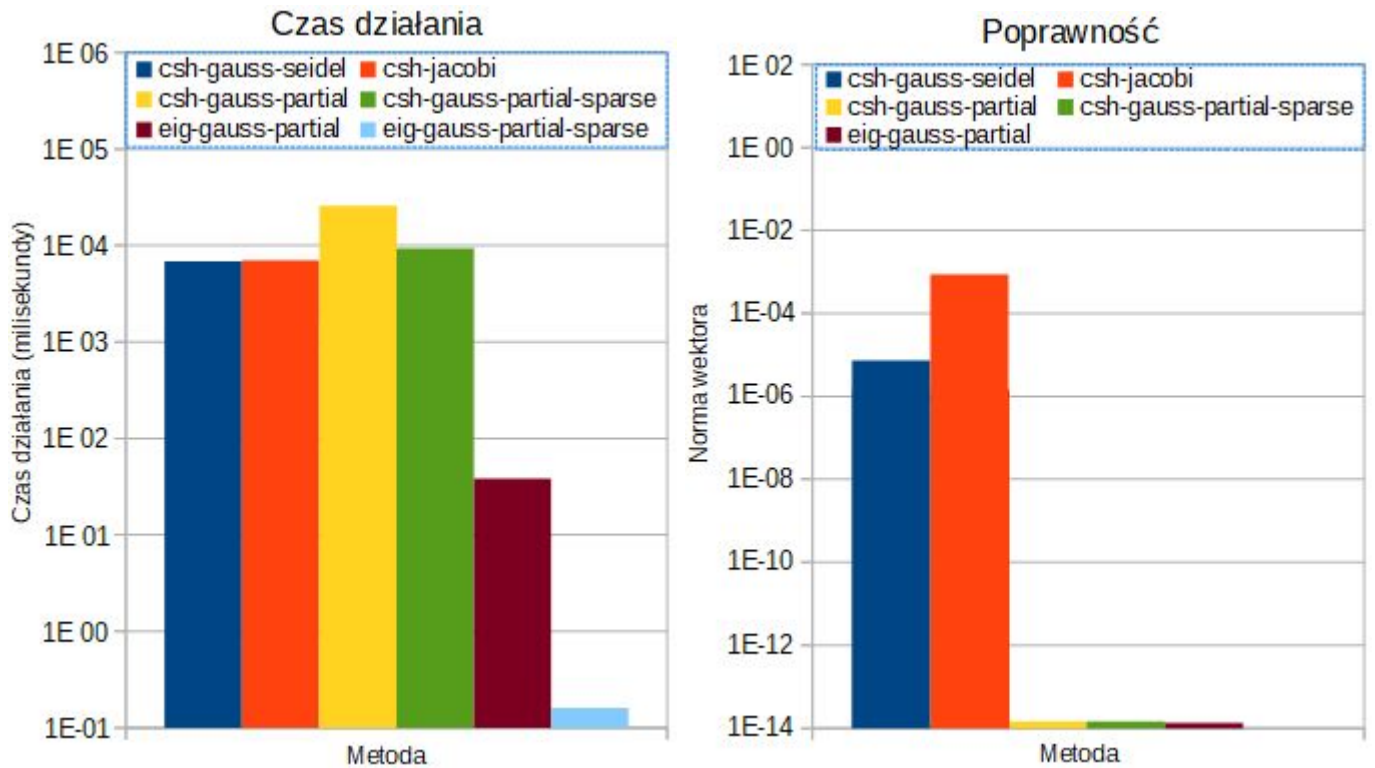
## Wyniki

- rozmiar planszy (N) = 10, pozycja  $G_1 = 10$ , pozycja  $G_2 = -10$
- 250 iteracji dla metody Jacobiego i Gaussa-Seidela

### Kostka symetryczna $V(-3, -2, -1, 0, 1, 2, 3)$ , $P(1, 1, 1, 1, 1, 1, 1)$

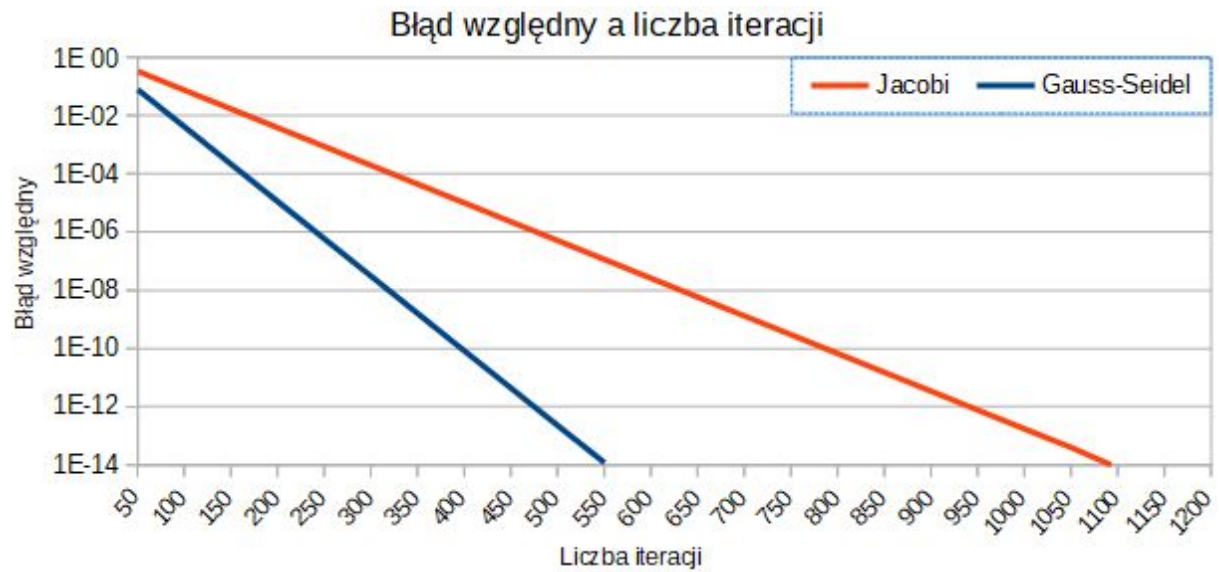


### Kostka niesymetryczna $V(-5, -2, 0, 1, 2)$ , $P(1, 3, 1, 4, 6)$



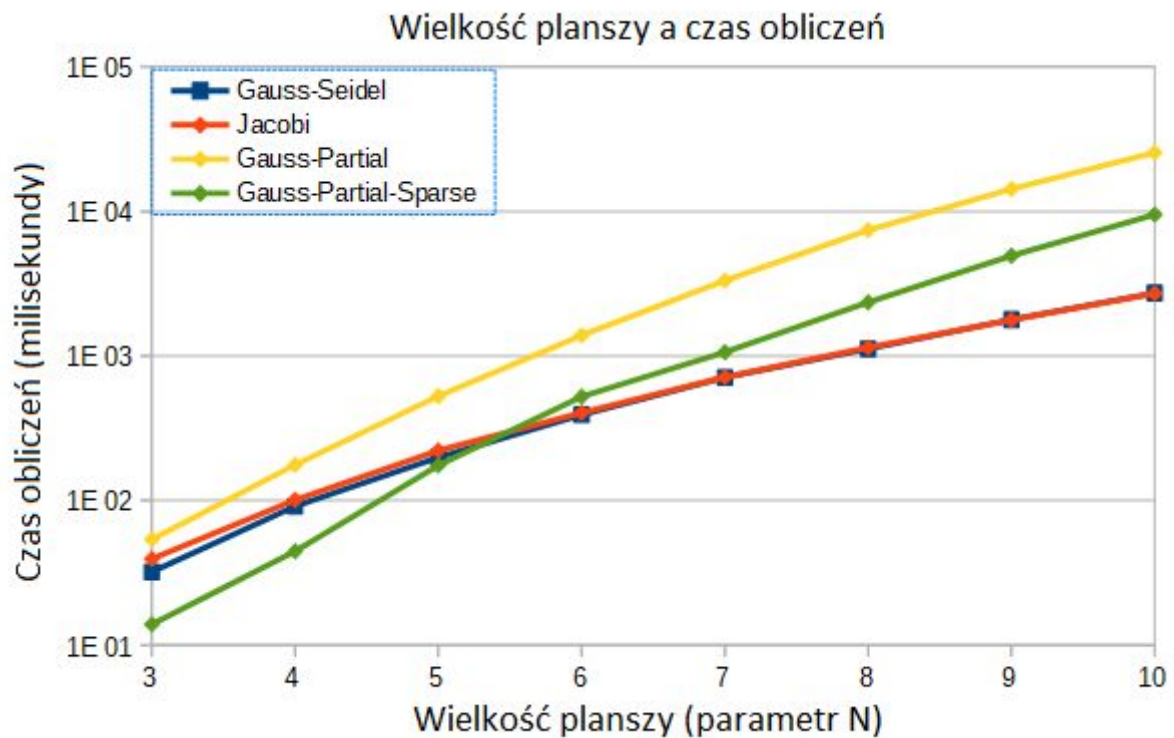
## Wyniki

- rozmiar planszy (N) = 10, pozycja  $G_1 = 10$ , pozycja  $G_2 = -10$
- kostka symetryczna  $V(-3, -2, -1, 0, 1, 2, 3)$ ,  $P(1, 1, 1, 1, 1, 1, 1)$



## Wyniki

- 100 iteracji dla metody Jacobiego i Gaussa-Seidela
- kostka symetryczna  $V(-2, -1, 0, 1, 2)$ ,  $P(1, 1, 1, 1, 1)$



## Wnioski

- Wyniki uzyskane za pomocą metod PartialPivLU, SparseLU i własnej implementacji metody Gaussa z częściowym wyborem są bardzo dokładne. Błąd oscyluje na granicy dokładności typu *double* w okolicach 0.
- Czas działania jest najkrótszy w przypadku metody SparseLU z biblioteki Eigen3, która używa dedykowanego typu do przechowywania macierzy rzadkich.
- Optymalizacja metody Gaussa dla macierzy rzadkich znacznie skraca czas obliczeń.
- Czas działania metod iteracyjnych jest zależny od ilości iteracji. W przypadku 250 iteracji i planszy o  $N = 10$ , metody te są szybsze niż metoda Gaussa z częściowym wyborem - zarówno w wersji z optymalizacją dla macierzy rzadkich, jak i bez.
- Zwiększenie ilości iteracji dla metod Jacobiego i Gaussa-Seidela znacząco zwiększa poprawność wyniku.  
Przy odpowiednio 1100 dla Jacobiego i 550 dla Gaussa-Seidela, uzyskany błąd względny oscyluje na granicy dokładności typu *double* w okolicach 0.
- Korzystanie ze składowych wyliczonych w każdym kroku w metodzie Gaussa-Seidela znacząco zmniejsza ilość iteracji potrzebnych do uzyskania dokładnego wyniku.
- Poprawność uzyskanych wyników została zweryfikowana metodą Monte-Carlo, czyli poprzez wielokrotne symulowanie przebiegu gry.