

## Grzybobranie

### Algorytmy 3

### Zadanie

Napisany został program, który oblicza prawdopodobieństwo wygrania gry przez gracza G1, mając dane wejściowe podane w kolejnych wierszach: rozmiar planszy, liczę grzybów oraz ich rozmieszczenie, pola startowe obu graczy, liczbę ścian kostki, wartości na ścianach kostki, rozkład kostki.

### Implementacja

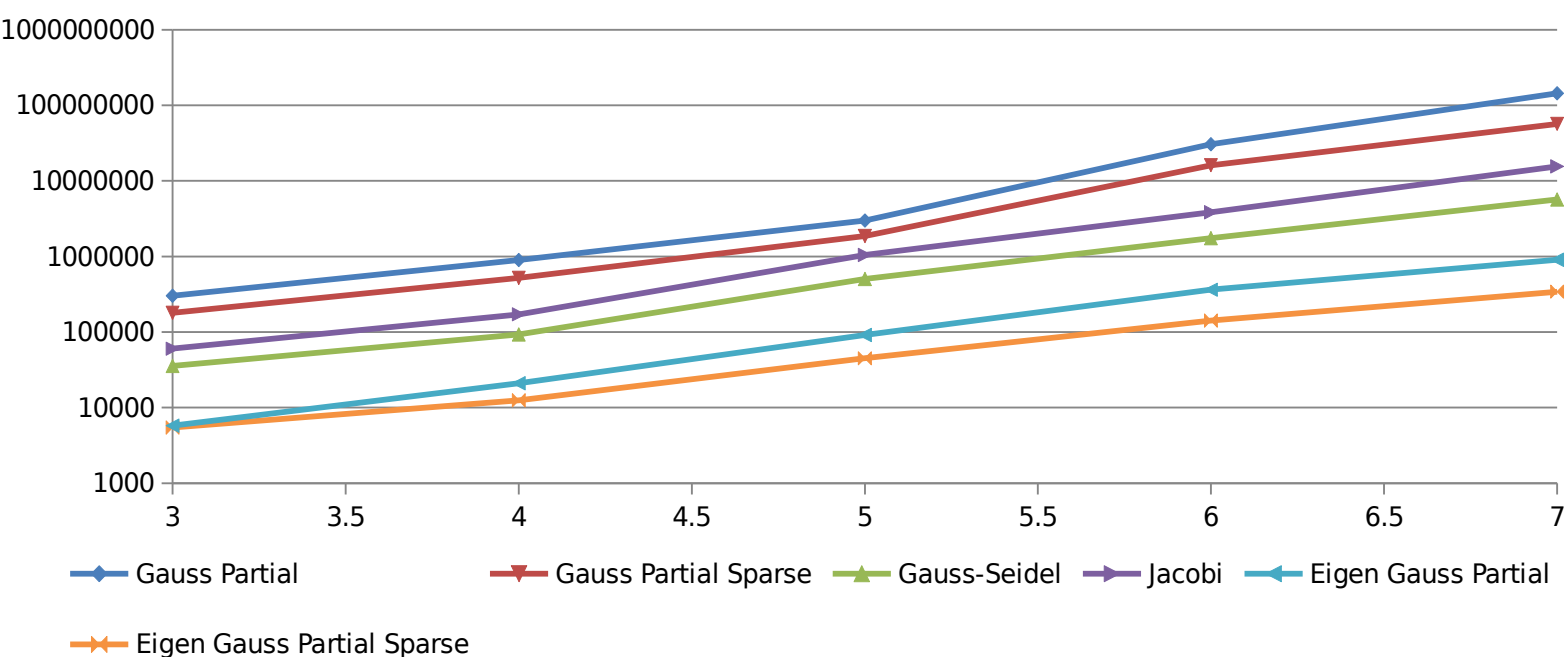
Implementacja opiera się o klasę MyMatrix z zadania drugiego, do której stworzone zostały implementacje metody Gaussa dla macierzy rzadkich, metody iteracyjne Gaussa-Seidela i Jacobiego.

Do tworzenia macierzy prawdopodobieństw stworzone zostały klasa Generator, która za argument przyjmuje nazwę pliku wejściowego oraz klasa State implementująca stan.

Do symulacji gry ( Metoda Monte-Carlo ) stworzone zostały klasy MonteCarlo, Dice, Field oraz Player.

### Czasy wykonania

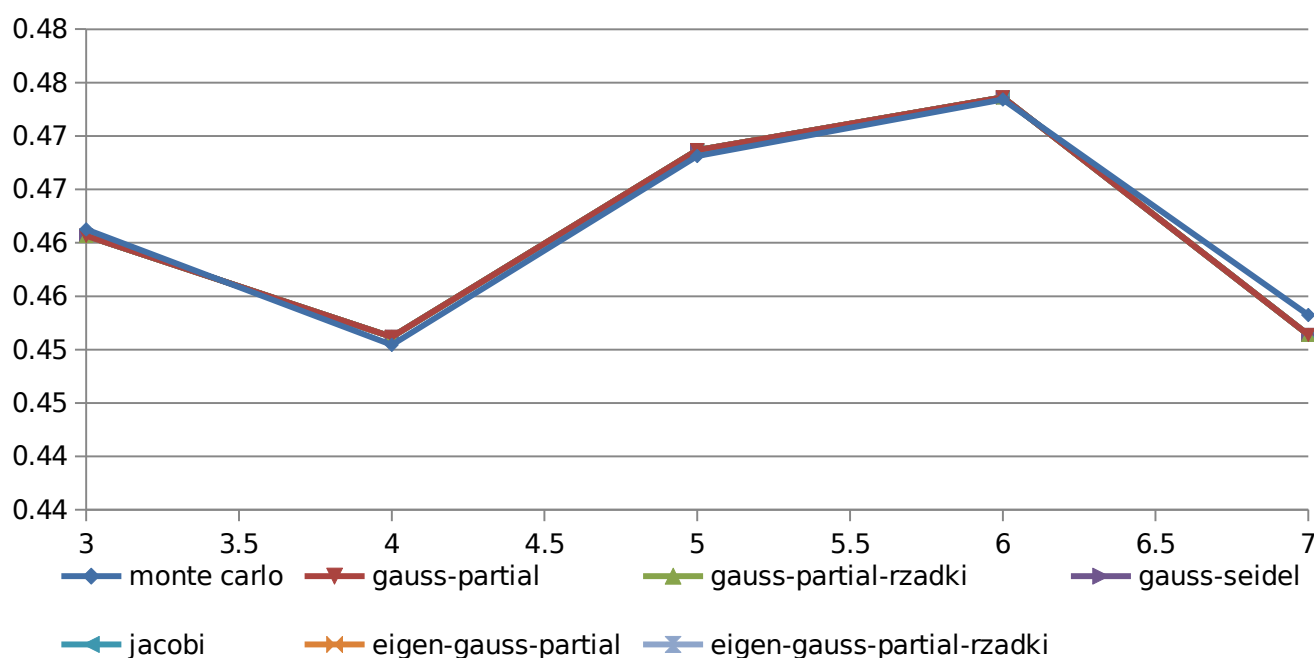
Oś y – cykle procesora, oś x – rozmiar planszy.



## Obliczenia

Stworzyłem możliwość eksport macierzy prawdopodobieństw do tabeli HTML (załączona), która to dla małych obliczeń (takich, których macierz da się zbudować ręcznie na kartce), pozwalała w początkowym stadium kontrolować poprawność generowania.

Następnie, przeprowadzone zostały obliczenia i symulacje dla różnych rozmiarów plansz, a wyniki zostały porównane.



Jak widać, wyniki obliczeń (metod) pokrywają się ze sobą, natomiast wyniki obliczeń i symulacją są bardzo zbliżone, co sugeruje poprawną implementację metod.

## Wnioski

Zgodnie z oczekiwaniami, najszybszą metodą obliczeniową jest zastosowanie `Eigen::SparseMatrix`, która jest implementacją macierzy rzadkich.