

# Implementacja population protocols oraz testy wydajnościowe

Grams, Stanisław  
Jezierski, Maciej  
Korczakowski, Juliusz  
MFI UG  
Algorytmy Numeryczne

16 stycznia 2019

## 1 Operacje na macierzach

### 1.1 O implementacji

Program „*protocols*” został napisany w języku C++, a wyniki działania programu zapisywane są do poszczególnych plików \*.csv.

### 1.2 Zaimplementowane algorytmy

- Algorytm Gaussa z częściowym wyborem elementu (PG - Partial Gauss)
- Algorytm Gaussa z częściowym wyborem elementu i optymalizacją dla macierzy rzadkich (PGS - Partial Gauss for Sparse Matrices)
- Algorytm Jacobiego
- Algorytm Gaussa-Seidela
- Metoda Monte Carlo

## 2 Implementacja i jej poprawność

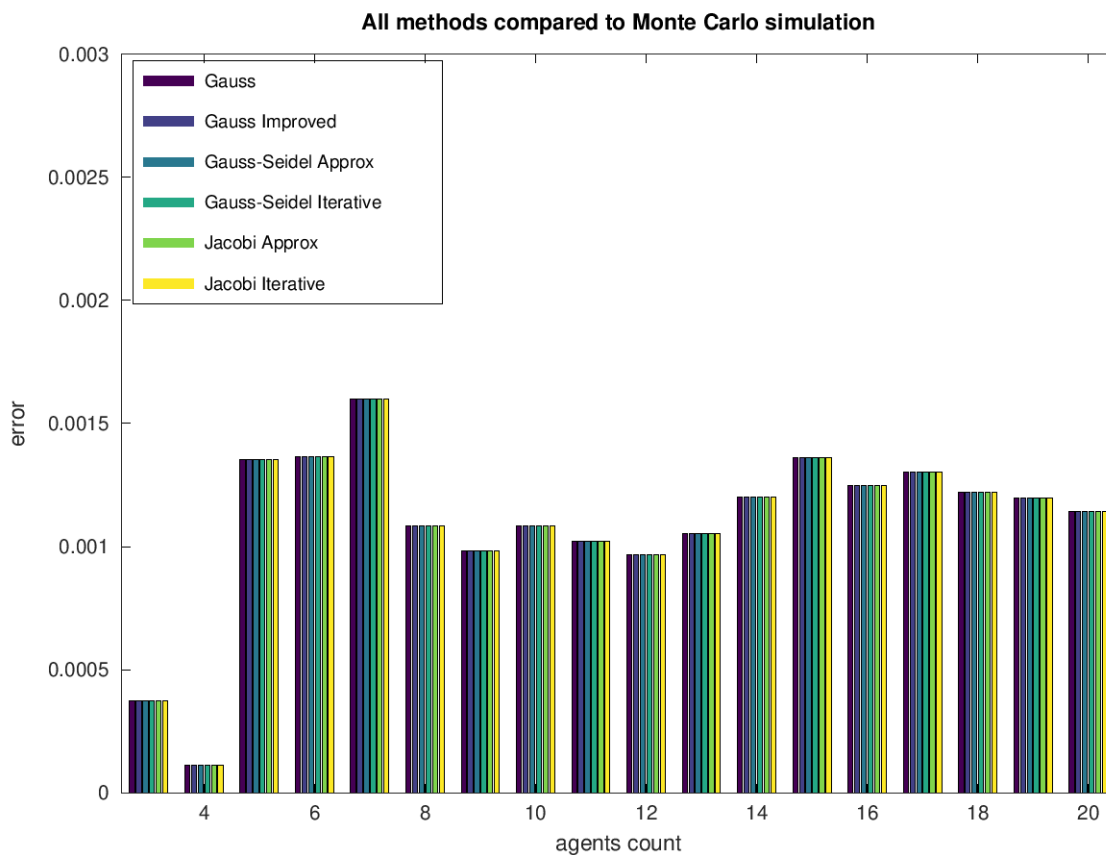
### 2.1 Generowanie układu równań dla danej liczby agentów

Generowanie układu równań dla danej liczby agentów  $N$  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.

### 2.2 Prawidłowość implementacji

By zweryfikować poprawność implementacji, zarówno generowania macierzy jak i obliczania stworzonego w ten sposób układu równań, wszelkie obliczenia porównywane były z wynikami wyliczonymi metodą Monte Carlo. Poniższy wykres obrazuje dokładność wszystkich zaimplementowanych algorytmów względem metody Monte Carlo, na podstawie którego można wnioskować o poprawności zaimplementowanych metod.

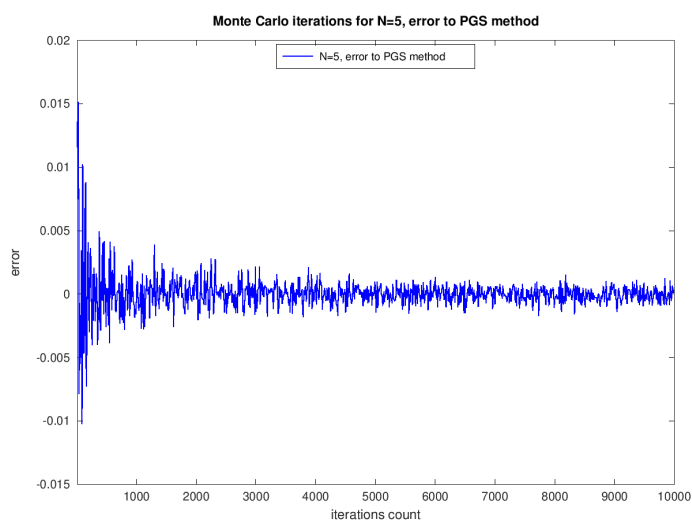
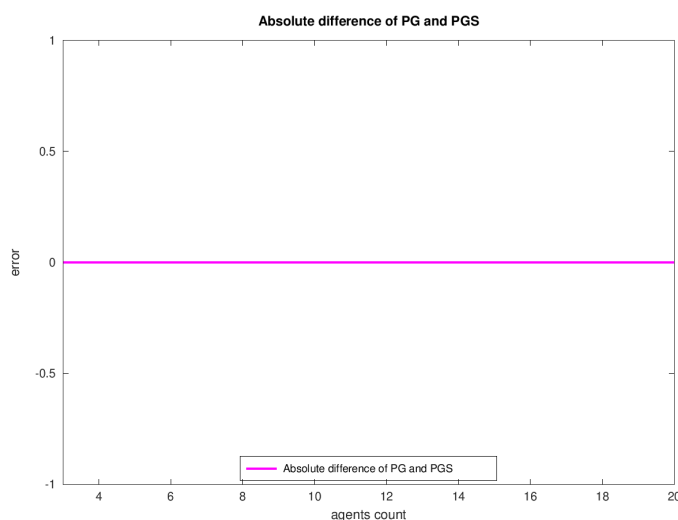


### 3 Analiza wyników i wydajność zaimplementowanych algorytmów

#### 3.1 Analiza wyników

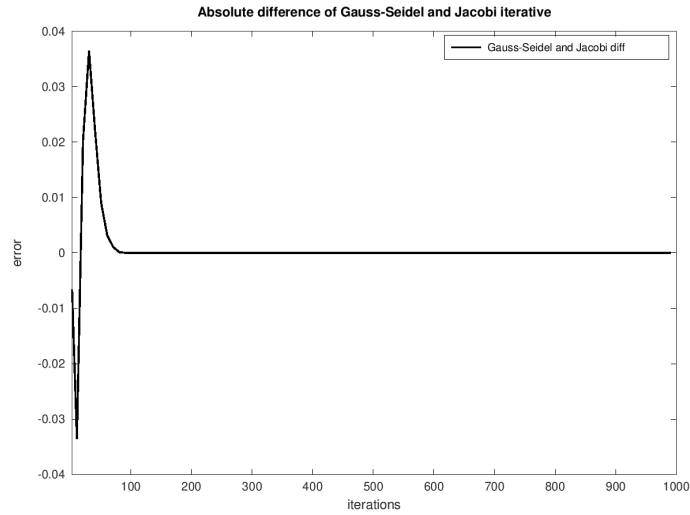
##### 3.1.1 Gauss oraz Gauss z optymalizacją dla macierzy rzadkich

Przeanalizujmy poniższy wykres. Wynika z niego jednoznacznie, że optymalizacja nie wpływa na dokładność. Wyraźnie widać, że na całej długości wykresu błąd wynosi 0.



##### 3.1.2 Algorytmy iteracyjne

Obie zaimplementowane przez nasz zespół metody wykazują się jednakową dokładnością.



## 3.2 Wydajność

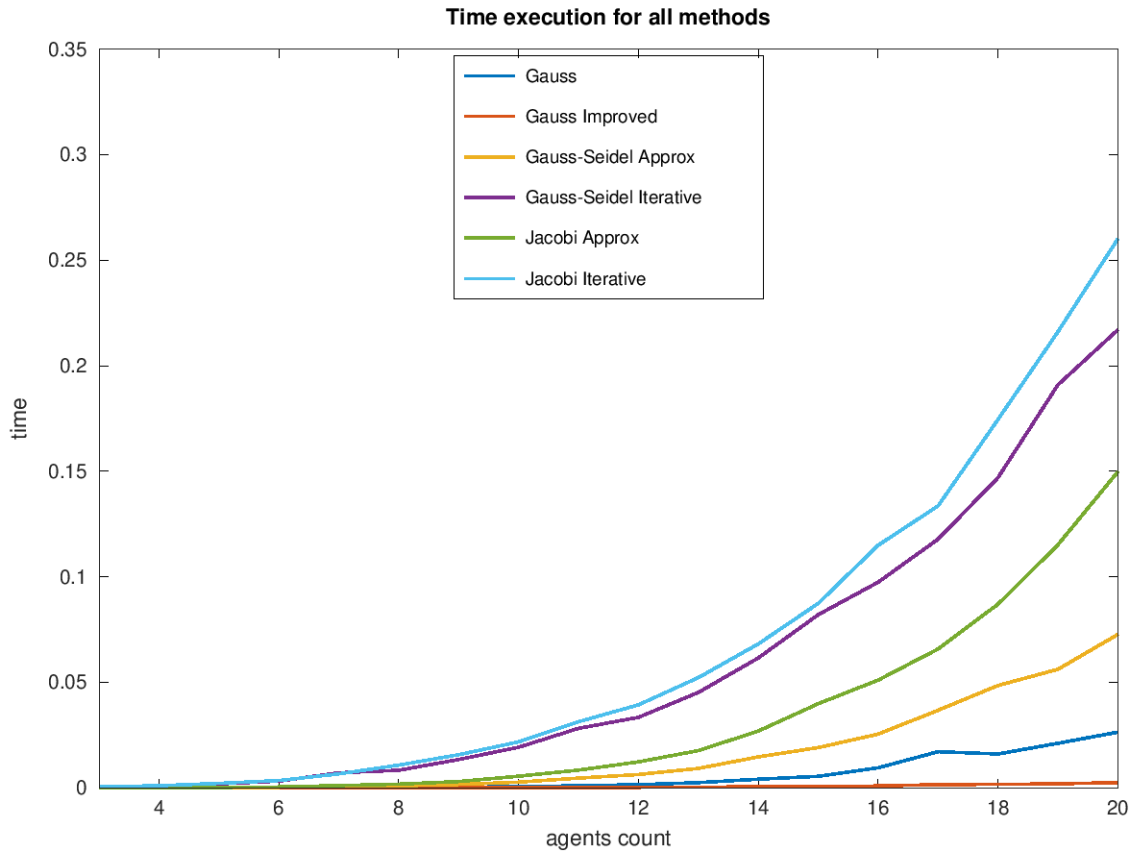
### 3.2.1 Wydajność względem wielkości planszy

Analizując pierwszy wykres w tym tekście oraz poniższy można wyciągnąć następujące wnioski:

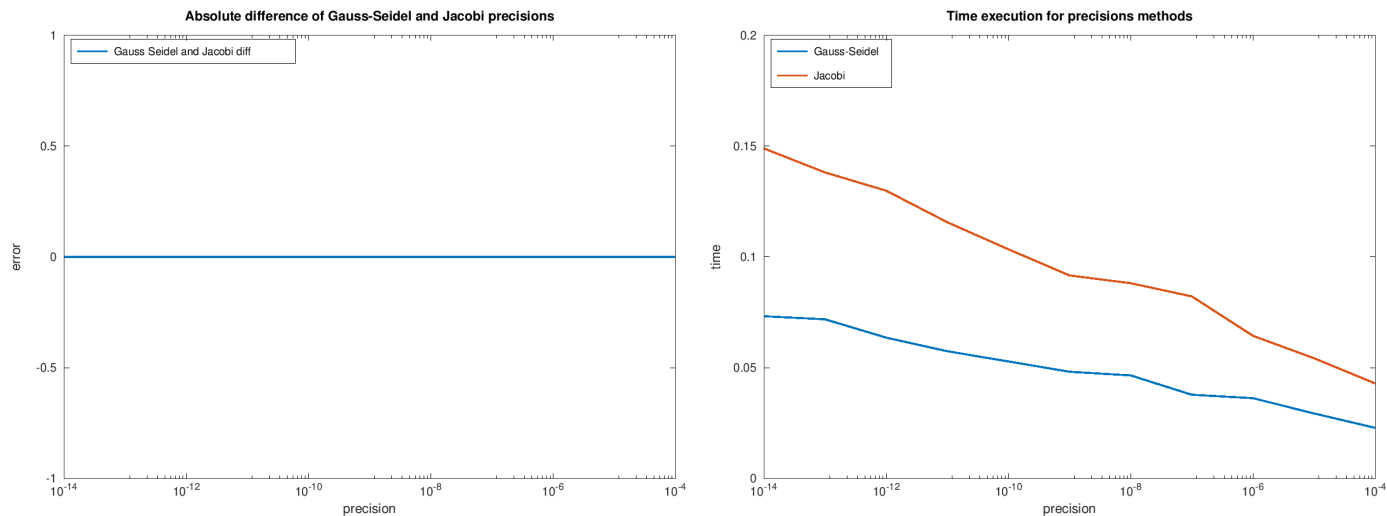
1. Pod względem błędów obliczeń wszystkie metody są bardzo podobne, przy małej wymaganej dokładności wręcz identyczne.

2. Pod względem czasu wykonania wyraźnie wygrywają obie metody na podstawie metody Gaussa, jednak wersja z usprawnieniem z racji swojej specyfikacji jest zdecydowanie najszybsza.

Powyższe wnioski wyraźnie wskazują, że w klasie wydajności względem wielkości planszy, jako optymalny wybór należy wskazać metodę Gaussa z ulepszeniem dla macierzy rzadkich.



### 3.2.2 Wydajność względem zadanej dokładności



W przypadku tego kryterium porównujemy tylko metody Jacobiego oraz Gaussa-Seidela. Na wykresie *All methods compared to Monte Carlo simulation*, jak i *Absolute difference of Gauss-Seidel and Jacobi precisions* widzimy, że metody te dają wręcz identyczną dokładność. W kwestii czasu wykonania oba algorytmy również plasują się bardzo podobnie, jednak istnieje widoczna różnica na korzyść metody Gaussa-Seidela.

Podsumowując, metody Jacobiego oraz Gaussa-Seidela są bardzo do siebie zbliżone, jednak ze względu na nieznacznie lepszy czas wykonania lepszą okazuje się być metoda Gaussa-Seidela.

## 4 Podział pracy

Stanisław Grams	Juliusz Korczakowski	Maciej Jezierski
Implementacja algorytmu Gaussa-Seidela	Implementacja algorytmu Jacobiego	Implementacja algorytmu PG oraz PGS
Implementacja symulacji Monte Carlo	Przygotowanie testów i ich uruchomienie	Analiza wykresów oraz przygotowanie sprawozdania
Implementacja algorytmu generowania macierzy	Przygotowanie wykresów końcowych	Praca nad strukturą projektu