

Implementacja population protocols oraz testy wydajnościowe

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

10 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

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

2 Implementacja i jej poprawność

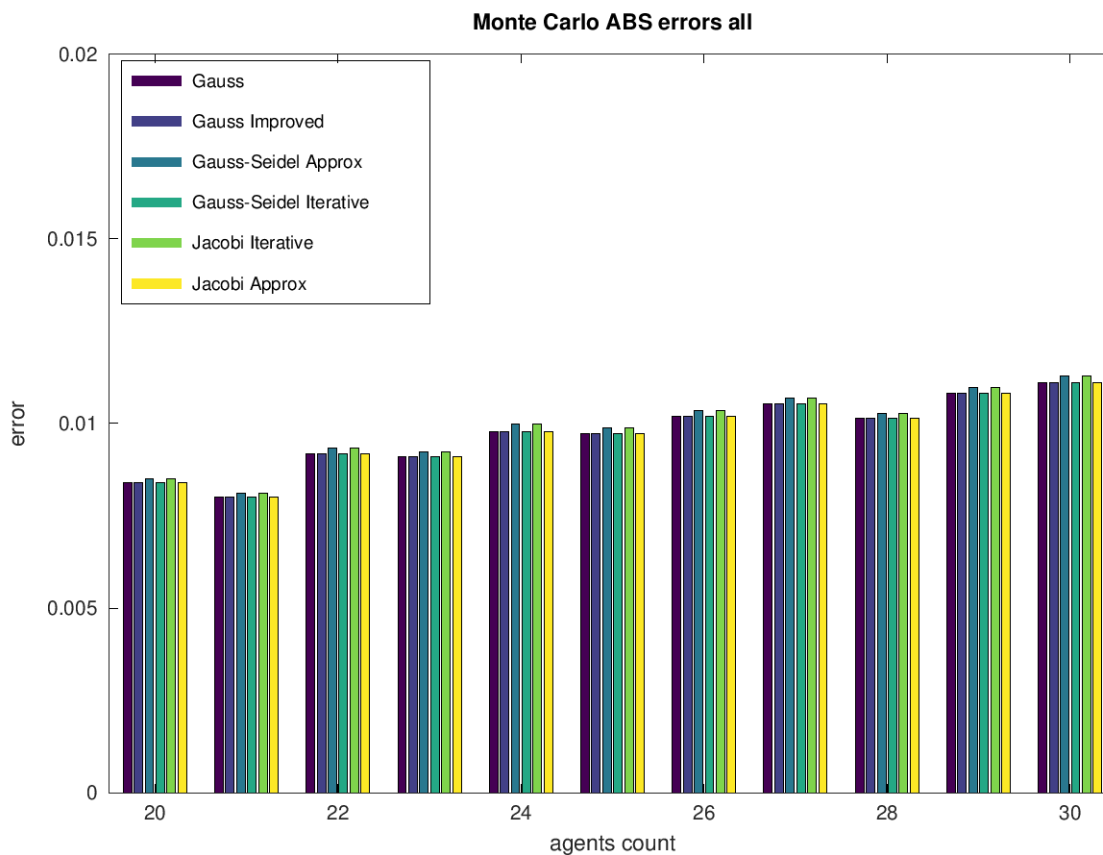
2.1 Generowanie układu równań dla danej liczby agentów

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.

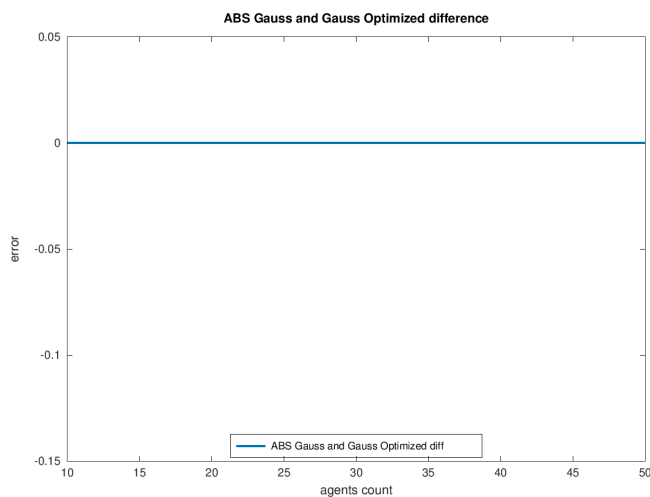


3 Analiza wyników i wydajność zaimplementowanych algorytmów

3.1 Analiza wyników

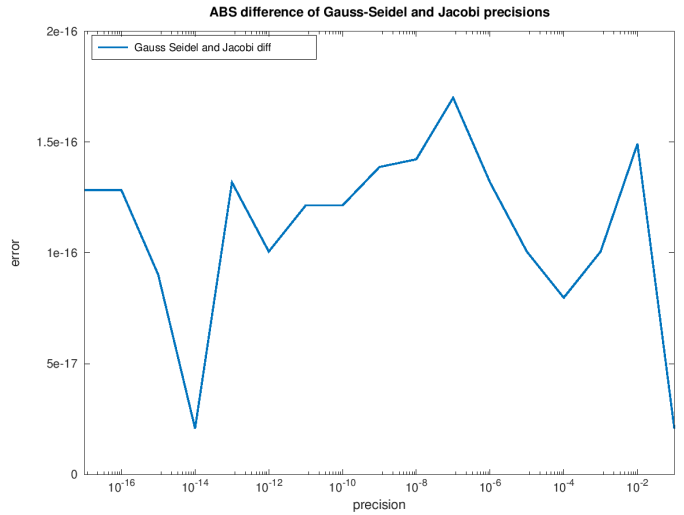
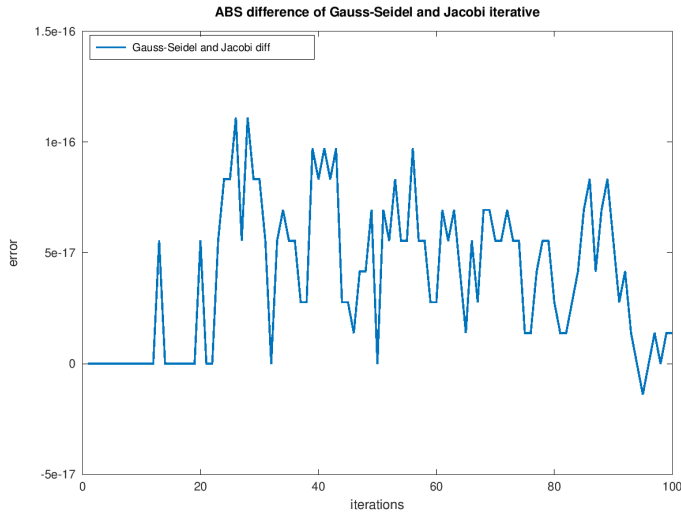
3.1.1 Gauss oraz Gauss z optymalizacją dla macierzy rzadkich

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



3.1.2 Algorytmy iteracyjne

Obie zaimplementowane przez nasz zespół metody oferują przyzwoitą dokładność jednak poniższe wykresy pozwalają wyciągnąć wniosek mówiący, że metoda Jacobi jest dokładniejsza.



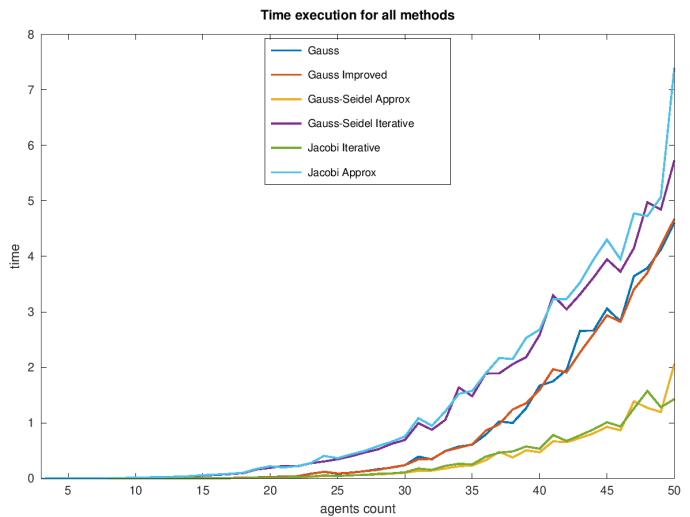
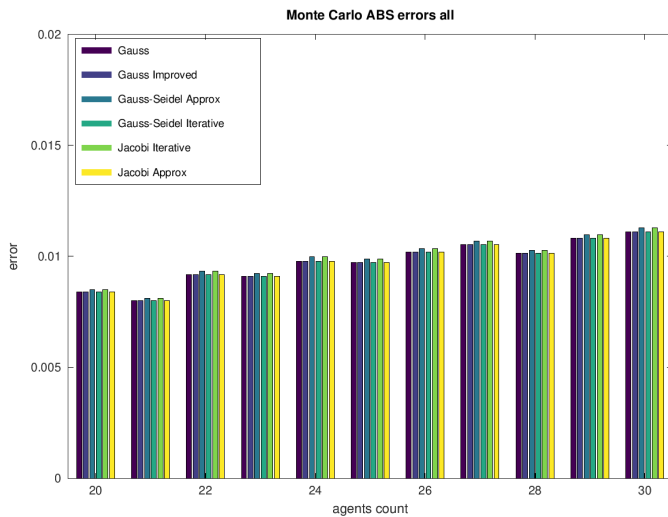
3.2 Wydajność

3.2.1 Wydajność względem wielkości planszy

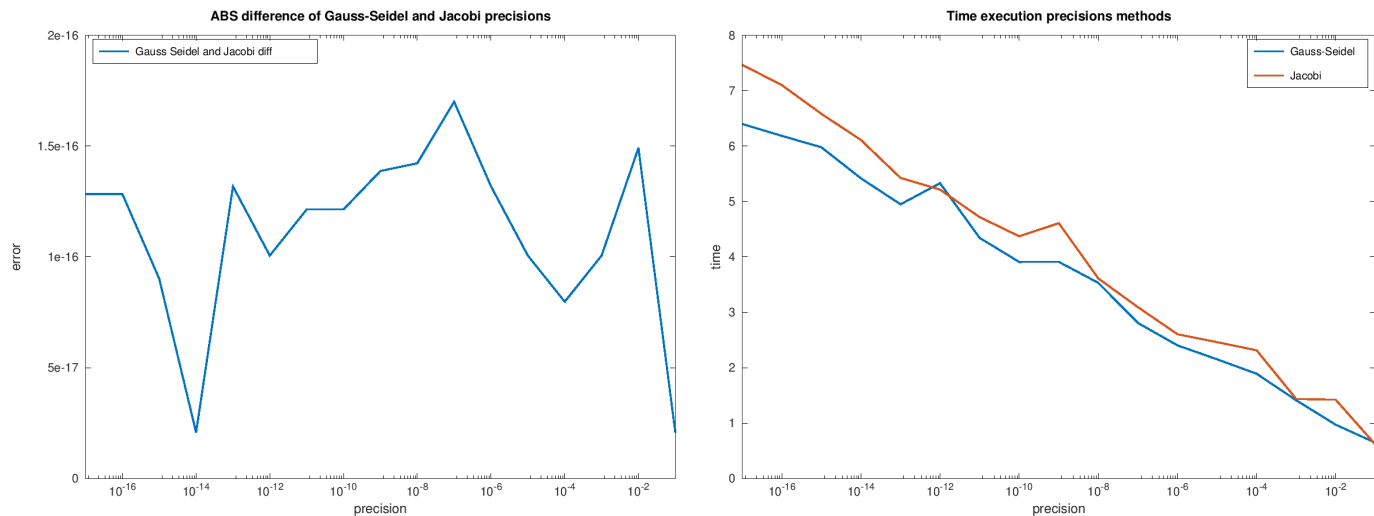
Analizując poniższe wykresy można wyciągnąć następujące wnioski:

1. Pod względem błędów obliczeń metody Gaussa oraz Gaussa z ulepszeniem dla macierzy rzadkich zdecydowanie wygrywa z innymi algorytmami oferując idealnie taką samą, wysoką dokładność.
2. Ze względu na swoją specyfikację metoda Gaussa z ulepszeniem dla macierzy rzadkich wygrywa z klasyczną wersją tej metody pod względem czasu wykonania.

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 pierwszym wykresie widzimy różnicę błędów między powyższymi metodami, w tej kategorii wygrywa metoda Jacobi. W kwestii czasu wykonania oba algorytmy plasują się bardzo podobnie, jednak tak jak w poprzednim przypadku w okolicach $\epsilon=10^{-12}$ następuje lekkie załamanie metody Gaussa-Seidela, tym razem na korzyść algorytmu Jacobi.

Podsumowując, metoda Jacobi jest dokładniejsza od Gaussa-Seidela, natomiast metoda Gaussa-Seidela jest szybsza od metody Jacobi dla tej samej ilości iteracji.

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