Algorytmy numeryczne - Projekt 3

Mateusz Soroka 250999
Bartłomiej Skopiński 246830
Patryk Szczepański 246760
Grupa 1 - aplikacje internetowe i bazy danych

9 grudnia 2018

Opracowanie dotyczy obliczenia prawdopodobieństwa zagłosowania na **TAK** w głosowaniu większościowym (ang. majority). Agenci biorący udział w głosowaniu przyjmują trzy możliwe stany *Y* - tak, *N* - nie, *U* - niezdecydowany. W trakcie obliczeń agenci zmieniają swoje stany według następujących reguł:

- {Y, U} -> {Y, Y},
- {Y, N} -> {U, U},
- {N, U} -> {N, N},

w pozostałych przypadkach stan nie ulega zmianie.

Program został napisany w języku Java (wersja 8.0.181), testy przeprowadzono na komputerze MacBook Air wyposażonym w procesor Intel Core i5 (1,8 GHz), pamięć 8 GB 1600MHz DDR3 z wersją systemu Mojave 10.14.1. Do testów wykorzystano typ zmiennoprzecinkowy double.

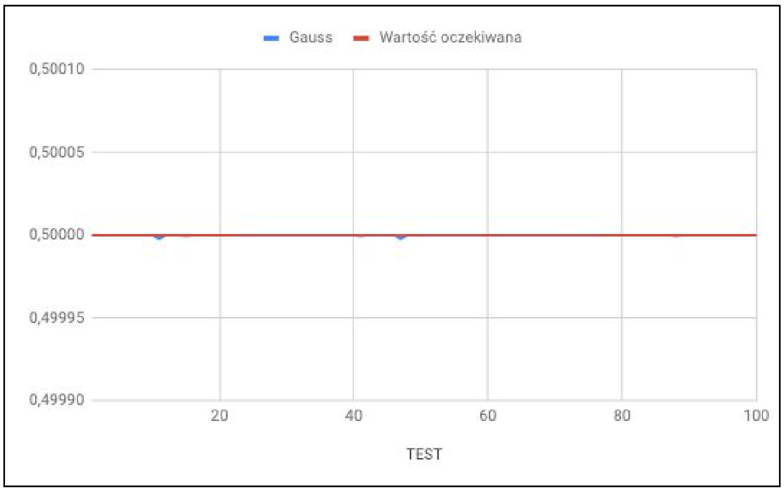
Aby zbudować układ równań należy rozważyć wszystkie możliwe przypadki głosowania przy liczbie agentów równej *N*. Weźmy przykład gdy *N=3*. Układ równań prezentuje się wtedy następująco ($P_{T,N}$, gdzie *T* - agenci na tak, *N* - agenci na nie).

Równania	**Otrzymana macierz kwadratowa *10x10***
$P_{0,0} = 0$ $P_{0,1} = 2/3\ P_{0,2} + 1/3\ P_{0,1}$ $P_{0,2} = 2/3\ P_{0,3} + 1/3\ P_{0,2}$ $P_{0,3} = 0$ $P_{1,0} = 2/3\ P_{2,0} + 1/3\ P_{1,0}$ $P_{1,1} = 1/3\ P_{2,1} + 1/3\ P_{1,2} + 1/3\ P_{0,0}$ $P_{1,2} = 2/3\ P_{0,1} + 1/3\ P_{1,2}$ $P_{2,0} = 2/3\ P_{3,0} + 1/3\ P_{2,0}$ $P_{2,1} = 2/3\ P_{1,0} + 1/3\ P_{2,1}$ $P_{3,0} = 1$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{2}{3} & 0 & 0 & \frac{2}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & -1 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{2}{3} & 0 & 0 & 0 & 0 & -\frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{2}{3} & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & 0 & -\frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

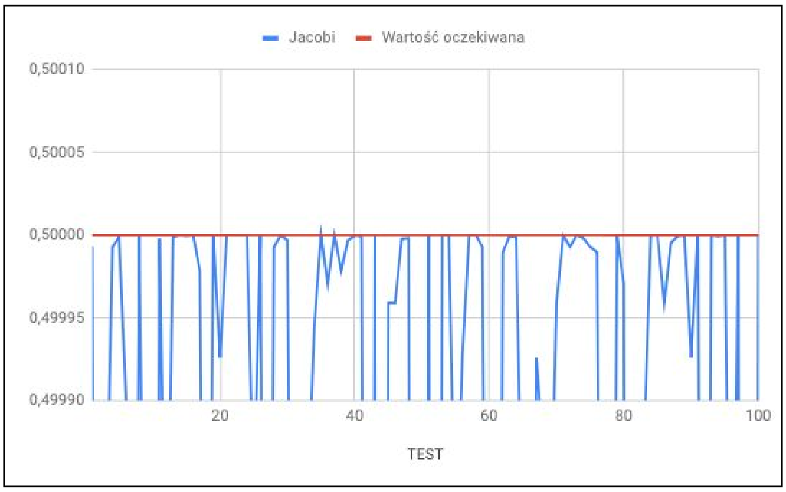
Do obliczenia prawdopodobieństwa wykorzystano układ równań liniowych oraz metody Gaussa (z częściowym wyborem elementu głównego) oraz iteracyjne: Jacobiego oraz Gaussa-Seidela.

Prawidłowość otrzymanych wyników zweryfikowano za pomocą metody Monte Carlo, losując liczbę parzystą agentów *N* z przedziału *[20, 120]* oraz biorąc stan początkowy *YES=NO=N/2*. Przewidywaną wartością prawdopodobieństwa otrzymania wyniku głosowania na tak, w sytuacji gdy połowa agentów jest na tak, a połowa na nie, jest *0.5*. Liczba iteracji w metodach iteracyjnych Jacobiego oraz Gaussa-Seidela została ustalona na *1000*.

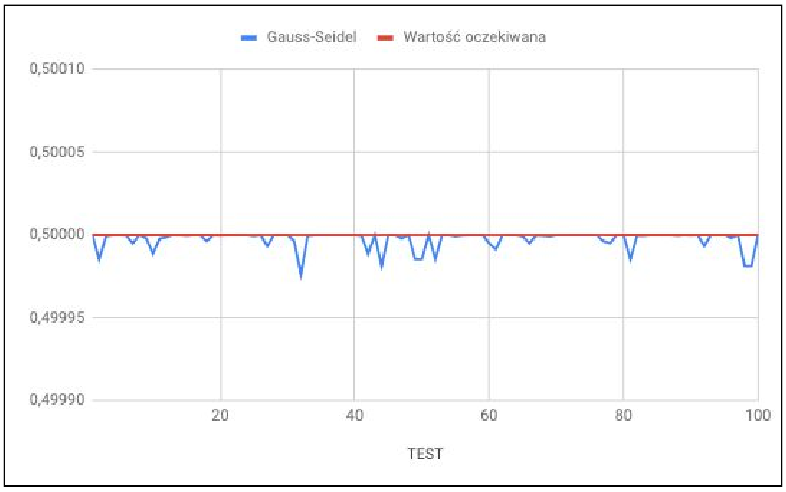
Metoda Gaussa z częściowym wyborem



Metoda Jacobiego



Metoda Gaussa-Seidela



Poniżej tabela z różnicami pomiędzy największym otrzymanym odchyłem, a wartością oczekiwaną we wszystkich trzech metodach.

Metoda Gaussa z częściowym wyborem	0.0000021800061
Metoda Jacobiego	0.001747781301818
Metoda Gaussa-Seidela	0.000023694374368

Powyższe wyniki pozwalają z wysokim prawdopodobieństwem stwierdzić, że wszystkie metody zostały zaimplementowane poprawnie. Łatwo zauważyć również, że w metodach iteracyjnych Jacobiego oraz Gaussa-Seidela odchyły od oczekiwanego wyniku są dużo większe niż w metodzie Gaussa z częściowym wyborem elementu podstawowego. Porównano zatem błędy bezwzględne pomiędzy metodą Gaussa, a metodami iteracyjnymi w zależności od liczby iteracji dla liczby agentów *N=80* i stanie początkowym *Y=34, N=39*.

Liczba iteracji	Błąd Gauss - Gauss-Seidel	Błąd Gauss - Jacobii
100	0.17342103813911655	0.17342772922658295
500	3.4238364535765786E-4	0.00797670523302449
1000	2.2252669595879127E-8	1.3421696458543186E-5
2000	1.6653345369377348E-16	3.415354110636315E-11
3500	1.6653345369377348E-16	1.6653345369377348E-16
5000	1.6653345369377348E-16	1.6653345369377348E-16

Jak więc widać, liczba iteracji ma znaczący wpływ na dokładność obliczeń. Używanie metod iteracyjnych przy tego typu obliczeniach ma sens tylko i wyłącznie wtedy, kiedy liczba iteracji ustalona jest na odpowiednio wysoką. Dla przeprowadzanego eksperymentu nie ma sensu ustawiać liczby iteracji na większą niż 3500, ponieważ obie metody osiągają wtedy maksymalny najmniejszy błąd bezwzględny. Poniżej tabela z porównaniem czasów wykonania algorytmu w metodach iteracyjnych w zależności od liczby iteracji dla liczby agentów *N=80*. Dla porównania czas wykonania algorytmu metodą Gaussa z częściowym wyborem wynosi 12.368s.

Liczba iteracji	Metoda Gaussa-Seidela	Metoda Jacobiego
100	2.137s	1.843s
500	10.188s	9.448s
1000	20.31s	19.213
2000	41.267s	35.662s
3500	71.442s	63.347s
5000	103.736s	90.784s

Zestawiając ze sobą wyniki błędów bezwzględnych oraz czasów wykonywania metod iteracyjnych można dojść do wniosku, że metoda Gaussa-Seidela najlepsze wyniki osiąga dla liczby iteracji 2000, a metoda Jacobiego dla 3500, jednak czas wykonywania algorytmu dla liczby iteracji 3500 jest niemal połowę większy niż dla 2000. Metoda Gaussa z częściowym wyborem przy liczbie agentów *N=80* wykonuje się 12.368s, a nadal daje najdokładniejsze wyniki.

Porównano także wyniki wszystkich trzech metod w zależności od *N* i trzech dokładności, odpowiednio 10^{-6}, 10^{-10} oraz 10^{-14}.

Tabela przedstawiająca wyniki z dokładnością do 10^{-6}

			Metoda		
N	Y	N	Gaussa	Jacobiego	Gaussa-Seidela
120	57	54	0,676638	0,674869	0,676612
110	54	55	0,436662	0,435772	0,436654
100	40	37	0,695771	0,695514	0,695769
90	30	30	0,500000	0,499920	0,500000
80	30	25	0,835390	0,835379	0,835390
60	20	20	0,500000	0,500000	0,500000
30	10	10	0,500000	0,500000	0,500000

Tabela przedstawiająca wyniki z dokładnością do 10^{-10}

			Metoda		
N	YES	NO	Gaussa	Jacobiego	Gaussa-Seidela
120	57	54	0,6766383944	0,6748687398	0,6766119494
110	54	55	0,4366622936	0,4357717544	0,4366537595
100	40	37	0,6957711329	0,6955144655	0,6957694048
90	30	30	0,5000000000	0,4999198993	0,4999996733
80	30	25	0,8353903922	0,8353790591	0,8353903711
60	20	20	0,5000000000	0,4999998195	0,5000000000
30	10	10	0,4999999959	0,4999999959	0,4999999959

Tabela przedstawiająca wyniki z dokładnością do 10^{-14}

			Metoda		
N	Y	N	Gaussa	Jacobiego	Gaussa-Seidela
120	57	54	0,67663839440719	0,67486873980742	0,67661194937996
110	54	55	0,43666229364408	0,43577175437628	0,43665375950730
100	40	37	0,69577113289336	0,69551446554075	0,69576940479395
90	30	30	0,49999999999999	0,49991989929493	0,49999967327504
80	30	25	0,83539039222515	0,83539039222515	0,83539037113147
60	20	20	0,50000000000000	0,49999981951701	0,49999999996189
30	10	10	0,49999999588421	0,49999999588420	0,49999999588420

Jak można zauważyć na powyższych tabelach, wraz ze zmniejszeniem dokładności, wyniki otrzymywane różnymi metodami są bardziej zbliżone, a przy małych rozmiarach planszy oraz małych dokładnościach są wręcz takie same. Wniosek z tego płynący jest taki, że im większa dokładność tym lepsze wyniki otrzymujemy.

Zakres prac członków zespołu

Mateusz Soroka	**Bartłomiej Skopiński**	**Patryk Szczepański**
Przeniesienie metody Gaussa z częściowym wyborem z poprzedniego projektu	Matematyczne opracowanie wariantów algorytmu	Przeprowadzanie testów
Refaktoryzacja kodu	Opracowanie metody Jacobiego	Opracowanie zapisu danych z testów do CSV
Implementacja Monte Carlo	Opracowanie metody Gaussa-Seidela	Sporządzenie wykresów
Zredagowanie sprawozdania	Implementacja wszystkich wariantów algorytmów	Poprawki w sprawozdaniu