

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:

- $\{Y, U\} \rightarrow \{Y, Y\}$,
- $\{Y, N\} \rightarrow \{U, U\}$,
- $\{N, U\} \rightarrow \{N, N\}$,

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.

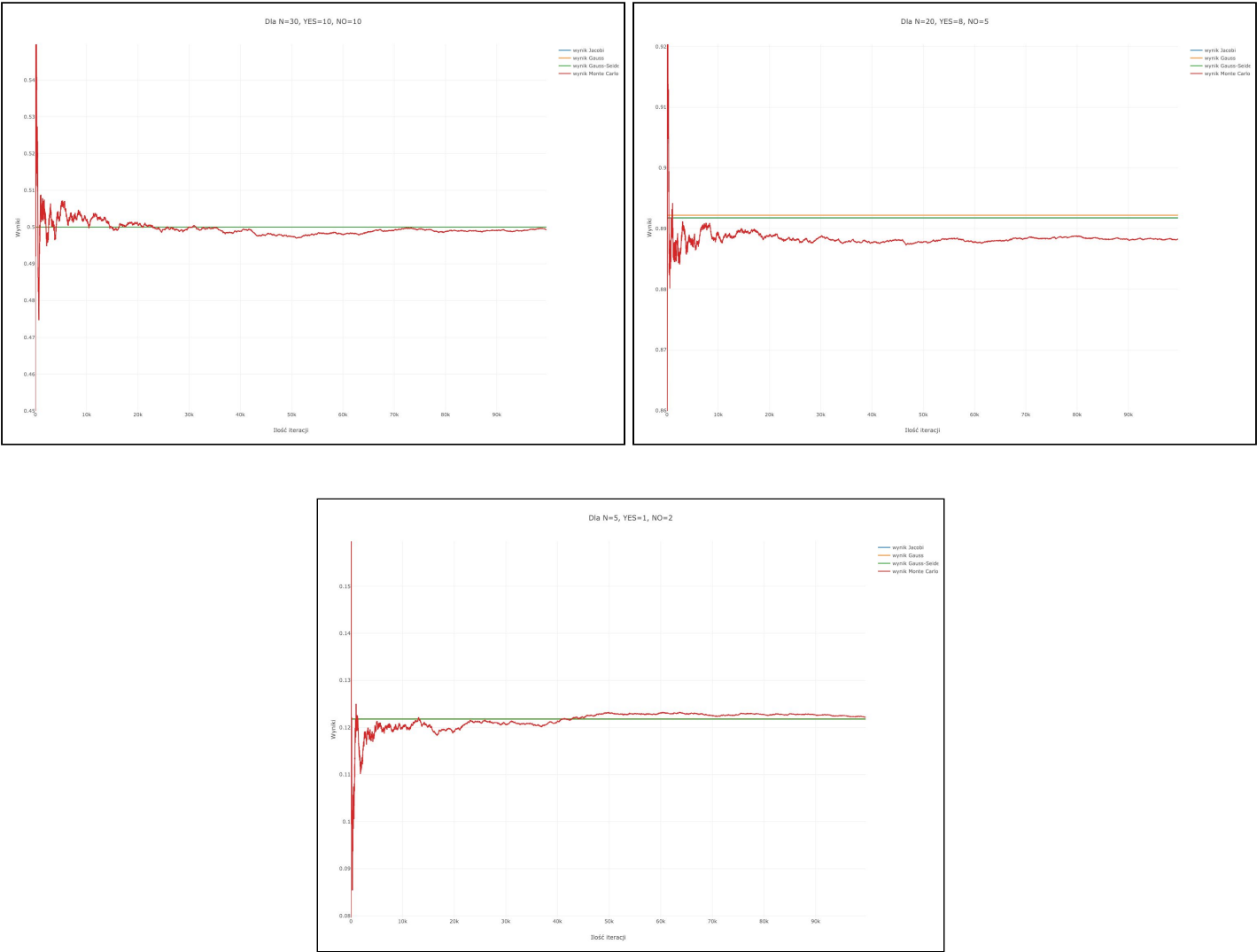
Aby zbudować układ równań należy rozważyć wszystkie możliwe przypadki głosowania przy liczbie agentów równej N . Weźmy przykład gdy $N=3$. Układ równań prezentuje się wtedy następująco ($P_{T,N}$, gdzie T - agenci na tak, N - agenci na nie). (Tabela 1.1)

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. Przeprowadzono 100 000 symulacji głosowania, które kończyło się jednym z trzech rezultatów (wszyscy na tak, wszyscy na nie, wszyscy niezdecydowani). Dzięki temu obliczamy prawdopodobieństwo zagłosowania wszystkich na tak. Wyniki symulacji Monte Carlo porównano z otrzymanymi wynikami ze wszystkich trzech metod, Gaussa z częściowym wyborem, Gaussa-Seidela i Jacobiego.

Porównanie wyników poszczególnych metod z symulacją Monte Carlo



Jak więc można zauważyć, w przypadku dla N=30 oraz YES=10 i NO=10, wraz ze wzrostem symulacji, prawdopodobieństwo otrzymane metodą Monte Carlo przybliża się do prawdopodobieństwa obliczonego za pomocą równań liniowych. W przypadku dla N=20, YES=8 i NO=5, prawdopodobieństwo otrzymane metodą Monte Carlo dużo wolniej zbliża się do prawdopodobieństwa otrzymanego z równań liniowych, natomiast dla N=5 oraz YES=1 i NO=2, prawdopodobieństwo otrzymane metodą Monte Carlo waha się , a blisko stu tysięcznej iteracji znacząco przybliża się do prawdopodobieństwa obliczonego za pomocą równań liniowych. Pozwala to stwierdzić, że metody Gaussa z częściowym wyborem, Gaussa-Seidela oraz Jacobiego zostały zaimplementowane poprawnie z wysokim prawdopodobieństwem. Łatwo zauważyć również, że wyniki uzyskane za pomocą ów trzech metod są identyczne, ponieważ na wykresie niemal na siebie zachodzą. Porównano więc błędy bezwzględne pomiędzy nimi dla różnych N i różnych danych początkowych.

			Metoda		
N	Y	N	Gauss oraz Jacobi	Gauss oraz Gauss-Seidel	Jacobi oraz Gauss-Seidel
5	1	2	0.0	0.0	0.0
20	8	5	8.881784197001252E-16	4.3405906189086796E-4	1.1102230246251565E-16

30	10	10	1.6653345369377348E-16	4.440892098500626E-16	2.7755575615628914E-16
40	12	19	7.28583859910259E-17	1.0408340855860843E-16	3.122502256758253E-17
80	32	31	3.3706427093882496E-8	3.3706426538770984E-8	5.551115123125783E-16
100	47	42	1.3967832054317952E-6	1.3967832056538398E-6	2.220446049250313E-16

Z powyższych wyników można zauważyć, że pomiędzy metodą Gaussa z częściowym wyborem, a metodami iteracyjnymi Gaussa-Seidela oraz Jacobiego, błąd bezwzględny rośnie wraz ze wzrostem N , natomiast błędy pomiędzy metodami iteracyjnymi są niemal identyczne niezależnie od wzrostu N . 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	0.17342103813911655
500	3.423836453575191E-4	3.4238364535760235E-4
1000	2.2252669346078946E-8	2.2252669484856824E-8
1500	1.44445566618856E-12	1.44445566618856E-12
2000	2.7755575615628914E-17	2.7755575615628914E-17
3500	2.7755575615628914E-17	2.7755575615628914E-17
5000	2.7755575615628914E-17	2.7755575615628914E-17

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 2000, 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 bez optymalizacji dla macierzy rzadkich wynosi 12.368s, a z optymalizacją dla macierzy rzadkich zaledwie 0.849s.

Liczba iteracji	Metoda Gaussa-Seidela	Metoda Jacobiego	Różnica w %
100	1.457s	3.571s	145.09%
500	6.301s	18.422s	192.37%
1000	12.54s	36.225s	188.88%
1500	19.525s	54.722s	180.27%
2000	24.626s	73.246s	197.43%
3500	44.085s	129.179s	193.02%
5000	63.594s	195.519s	207.45%

Zestawiając ze sobą wyniki błędów bezwzględnych oraz czasów wykonywania metod iteracyjnych można dojść do wniosku, że obie metody iteracyjne osiągają identyczne wyniki, jednak obliczenia metodą Jacobiego wykonują się niemal trzykrotnie dłużej od metody Gaussa-Seidela. Metoda Gaussa z częściowym wyborem przy liczbie agentów $N=80$ wykonuje się 12.368s bez optymalizacji dla macierzy rzadkich, z optymalizacją 0.849s, a wyniki wychodzą niemal identyczne dla liczby iteracji 2000. Przy tej liczbie iteracji obliczenia metodą Gaussa z częściowym wyborem z dodatkową optymalizacją dla macierzy rzadkich wykonują się niemal 30 krotnie szybciej względem metody Gaussa-Seidela i niemal 87 krotnie względem metody Jacobiego.

We wcześniejszych rozważaniach użyto terminu “optymalizacja dla macierzy rzadkich” w kontekście metody Gaussa z częściowym wyborem. Odnosząc się do tabeli 1.1, w prawie wszystkich wierszach występuje bardzo duża ilość zer. Można więc zoptymalizować metodę Gaussa z częściowym wyborem w taki sposób, że sprowadzając macierz do macierzy schodkowej można pominąć zerowanie elementu, jeśli jest on już zerem. Powinno to znacząco przyspieszyć czas wykonywania algorytmu co udowadnia poniższe porównanie czasów wykonywania algorytmu Gaussa z częściowym wyborem bez oraz z optymalizacją dla macierzy rzadkich dla różnych N .



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	Gausa	Jacobiego	Gausa-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	Gausa	Jacobiego	Gausa-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