

Algorytmy Numeryczne Projekt 3

Zestaw danych do zadania:

$l = 5$ (liczba ścian kostki)

$k = 8$ (liczba grzybów na planszy)

1. Stosowanie metod iteracyjnych.

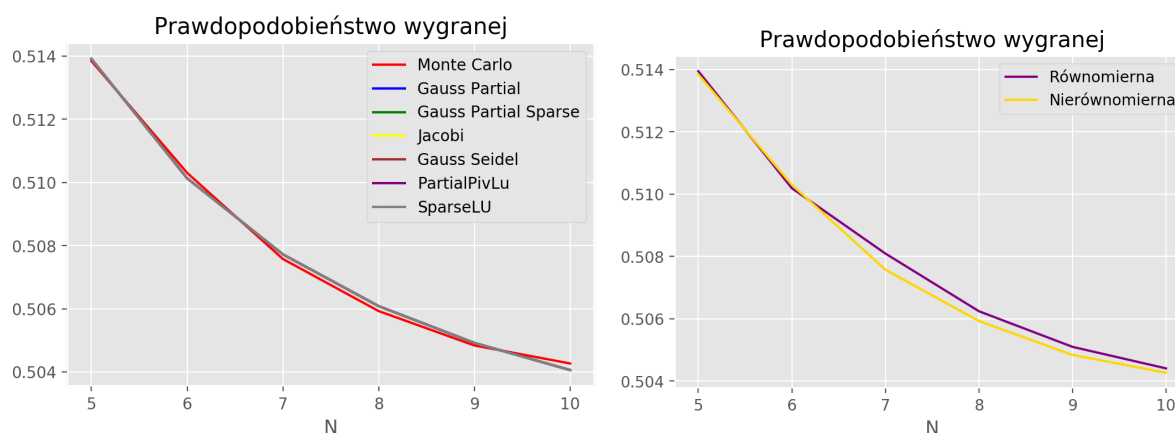
Proces iteracyjny $X^{(k+1)} = CX^{(k)} + F$ jest nieskończony. Tylko w nielicznych przypadkach może się zdarzyć, że dokładne rozwiązanie układu równań zostanie uzyskane w skończonej ilości iteracji. Warunkiem zakończenia iteracji może być wyrażenie:

$$\frac{q}{1-q} \|X^{(k)} - X^{(k-1)}\| < \varepsilon$$

Dlatego zakłada się, że $X^{(k)}$ jest tylko przybliżeniem dokładnego rozwiązania równania. Warto też wspomnieć o wrażliwości wektora rozwiązań na zaburzenia wartości macierzy głównej i wektora wyrazów wolnych układu. $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ jest wskaźnikiem uwarunkowania zadania. Dla każdej macierzy $\kappa(A) > 1$. Mała wartość $\kappa(A)$ oznacza niewielką wrażliwość rozwiązania na zaburzenia B . Z tego wynika, że przy zastosowaniu zbyt małej liczby iteracji, bądź zbyt dużej liczby ε , możemy otrzymać wyniki mocno odbiegające od dokładnego rozwiązania układu równań. Należy podawać dostatecznie mały ε i stosować wzory na normy wektorów w celu upewnienia się, że dostateczna liczba iteracji jest wystarczająca, aby otrzymać najdokładniejszy wynik.

2. Weryfikacja metodą Monte Carlo.

Bez grzybów



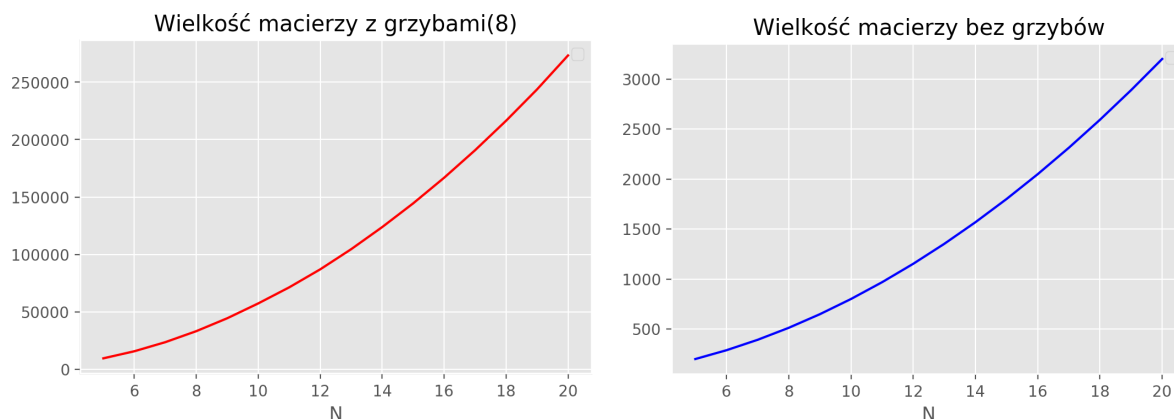
Na podstawie powyższych wykresów stwierdziliśmy, że wyniki zwrócone przez zaimplementowane funkcje pokrywają się z tymi otrzymanymi metodą Monte Carlo.

Kostka nierównomierna miała następujący rozkład prawdopodobieństw:

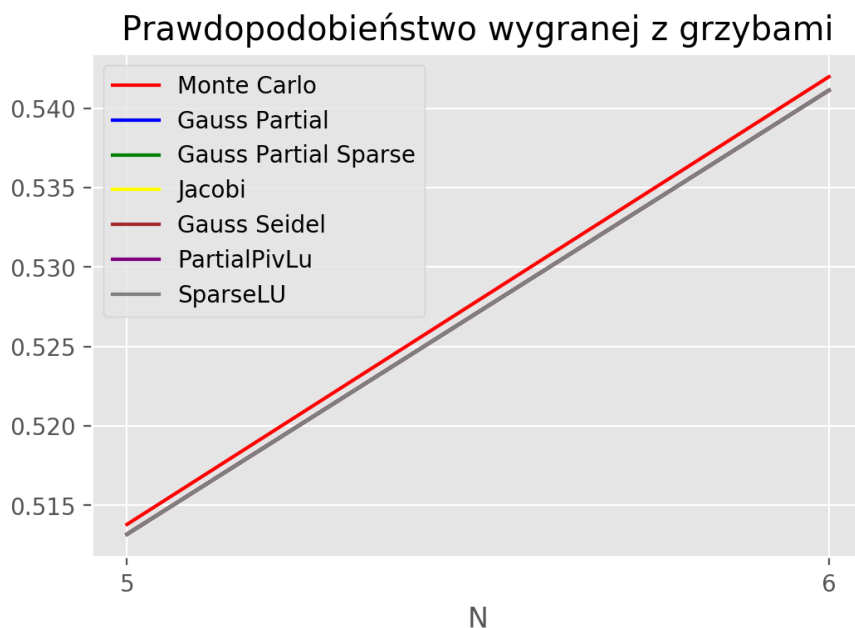
$\{-2: \frac{15}{100}; -1: \frac{30}{100}; 0: \frac{10}{100}; 1: \frac{30}{100}; 2: \frac{15}{100}\}.$

Dla tego rozkładu prawdopodobieństwo wygranej jest mniejsze.

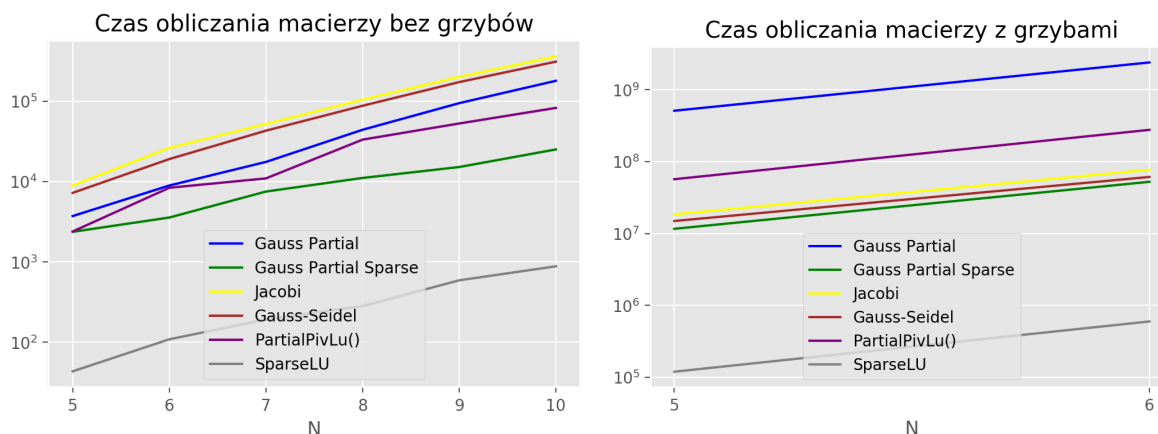
Z grzybami



Z naszym parametrem liczby grzybów na planszy $k = 8$, macierze przybierają rozmiary, które z trudnością mieszczą się w pamięci operacyjnej. Załóżmy, że chcemy obliczyć macierz dla $N = 10$, przyjmijmy że rozmiar macierzy wynosi wtedy 50 000, w takim razie potrzebowalibyśmy $50\,000 \cdot 50\,000 \cdot 8B = 20GB$ pamięci operacyjnej. Z tego względu nasze komputery pozwoliły tylko na obliczenie prawdopodobieństw dla $N = 5$ i $N = 6$.



3. Porównanie otrzymanych wyników.



- **Metody Gaussa : z optymalizacją i bez**

Algorytm Gaussa z optymalizacją okazał się mniej wydajny tylko od metody SparseLU z biblioteki Eigen3 zarówno dla wersji z grzybami, jak i bez grzybów. Dla wielkich macierzy (przy wersji z grzybami) algorytm bez optymalizacji jest najmniej wydajny z powyższego zestawu metod.

- **Metody iteracyjne: Jacobiego i Gaussa-Seidela**

Metoda Gaussa-Seidela jest szybsza od metody Jacobiego, gdyż opiera się na obserwacji, że w metodzie Jacobiego podczas obliczania $x_i^{(k+1)}$, wartości $x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ są już znane. Najprawdopodobniej są one lepszym oszacowaniem rozwiązania niż $x_1^{(k)}, \dots, x_{i-1}^{(k)}$. Dzięki temu algorytm Gaussa-Seidela potrzebuje do obliczenia wyniku mniej czasu, niż algorytm Jacobiego. Algorytmy iteracyjne dla mniejszych macierzy (wersja bez grzybów) okazały się najwolniejsze z całego zestawu.

- **Metody z biblioteki Eigen3 : partialPivLu i SparseLU**

Metody SparseLU operująca na macierzach rzadkich, okazała się najszybsza z badanego zestawu. Metoda ta jest dużo bardziej wydajna od metody partialPivLu.