Algorytmy w Inżynierii Danych - Projekt
Metody wyznaczania wartości własnych.

Jakub Korczakowski 291079, Piotr Rosa 291112
Wydział Elektryczny, Politechnika Warszawska

10 maja 2021

1 Wstęp

Tematem projektu, który wybraliśmy do zrealizowania w ramach przedmiotu Algorytmy w Inżynierii danych jest Temat 2 - Metody wyznaczania wartości własnych. Celem tego tematu jest zaimplementowanie i przeprowadzenie analizy porównawczej trzech algorytmów wyznaczania wartości i wektorów własnych macierzy. Wybrane algorytmy to: metoda potęgowa, QR oraz metoda Jacobiego. Algorytmy zostały zaimplementowane w języku Julia. Wybór padł na te algorytmy, ze względu na to, że są one stosowane do trzech różnych rodzai macierzy. Metoda potęgowa może być stosowana tylko do wyznaczania niezespolonych wartości własnych, metoda Jacobiego stosowana jest do macierzy symetrycznych, natomiast metoda QR do macierzy również niesymetrycznych.

Podczas pracy oprócz materiałów z wykładu używaliśmy dwóch źródeł: [2] oraz [1] .

Pakietem użytym przy implementacji algorytmów był pakiet *LinearAlgebra 1.5.4*, natomiast do testowania wydajności algorytmów i wizualizacji wyników użyte zostały pakiety *BenchmarkTools v0.7.0* oraz *PyPlot v2.9.0*

2 Opis algorytmów

2.1 Metoda Potęgowa

W ramach obliczania wartości własnych metodą potęgową, w każdej iteracji alokowany jest wektor własny. Z tego względu złożoność pamięciowa metody potęgowej wynosi $O(n)$. Złożoność obliczeniowa równa jest złożoności mnożenia wektora przez macierz, a więc jest to złożoność $O(n^2)$

Zbieżność metody potęgowej jest liniowa. W zależności od tego, jak bardzo największa co do modułu wartość własna λ_1 różni się od kolejnej wartości własnej λ_2, algorytm wynik może zbiegać się szybciej lub wolniej. Zależność tę można opisać za pomocą wzoru $error = (\lambda_2/\lambda_1)^2$. Jeśli największa wartość własna jest liczbą zespoloną, metoda jest rozbieżna i nie jest w stanie wyznaczyć tej wartości.

2.2 QR

W przypadku algorytmu QR do wyznaczania wartości własnych macierzy, w każdej iteracji tworzona jest pełna macierz o rozmiarach odpowiadających macierzy wejściowej. Złożoność pamięciowa tej metody to zatem $O(n^2)$. W każdej iteracji działania metoda wykonuje mnożenie macierzy o rozmiarach proporcjonalnych do macierzy wejściowej, stąd złożoność obliczeniowa wynosi $O(n^3)$.

W wyniku metody otrzymywana jest macierz górnotrójkątna, gdzie na diagonali znajdują się wartości własne macierzy. Jeśli wartości te zostaną posortowane malejąco co do modułu, to elementy pod diagonalą dążą do zera zgodnie ze wzorem $|a_{ij}^{(k)}| = O(|\lambda_i/\lambda_j|^k)$, gdzie k jest numerem iteracji, λ oznacza wartość własną oraz $i > j$

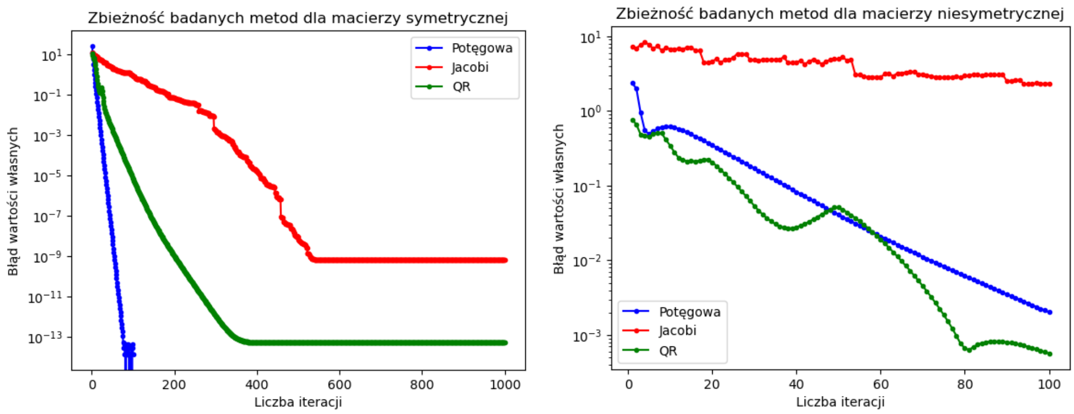
2.3 Metoda Jacobiego

Metoda Jacobiego, podobnie jak metoda QR, każdej iteracji tworzy macierz o takich samych wymiarach jak macierz wejściowa, a więc posiada złożoność pamięciową rzędu $O(n^2)$. Również jak w przypadku metody QR, w każdej iteracji wykonywane jest mnożenie macierzy, co oznacza, że metoda ta również ma złożoność obliczeniową równą $O(n^3)$.

Metoda Jacobiego jest zbieżna kwadratowo.

3 Implementacja i testy

Testy przeprowadzane były na maszynie o procesorze Intel(R) Xeon(R) CPU E31270 @ 3.40GHz, posiadającym 8 rdzeni oraz 16 wątków. Maszyna posiada 16G pamięci RAM. Dysk twardy używany podczas eksperymentów to WD7502AAEX-00Y9A0 750GB. Używany był system operacyjny Linux Ubuntu 20.04 focal. Algorytmy zostały przetestowane w programie Jupyter Notebook w przeglądarce Google Chrome. Notatnik zawierający kod oraz dodatkowe wykresy załączyliśmy wraz ze sprawozdaniem. Macierze testowe przygotowywane były jako macierze losowe. Błąd mierzony był jako wartość bezwzględna różnicy wektora wartości własnych oraz zakładanych wartości własnych (wartości własnej w przypadku metody potęgowej).



Rysunek 1: Porównanie zbieżności metod dla macierzy symetrycznej i niesymetrycznej z elementami rzeczywistymi. Od lewej: macierz symetryczna i macierz niesymetryczna. Wektor testowy inicjalizowany był jako wektor jedynek.

Metoda	Mediana czasu działania	Estymowana użytej pamięci	Liczba alokacji
Potęgowa	57.408 μs	47.47 KiB	204
Jacobi	638.932 μs	1.27 MiB	404
QR	5.170 ms	2.56 MiB	808

Tablica 1: Porównanie wyników testów wykonanych za pomocą metody @benchmark.

4 Wnioski

W tym raporcie opisaliśmy postęp prac nad zadanym projektem. Zaimplementowaliśmy wstępne wersje trzech algorytmów do wyznaczania wartości własnych, odpowiednio: metody potęgowej, algorytmu Jacobiego oraz algorytmu QR. Przeprowadziliśmy testy zbieżności oraz złożoności obliczeniowej. Wyniki, które uzyskaliśmy potwierdziły założenia teoretyczne dotyczące tych algorytmów. Zbadaliśmy również wydajność za pomocą narzędzi dostępnych w języku Julia.

W dalszej części prac nad projektem chcielibyśmy skupić się na optymalizacji trzech wspomnianych metod. Chcielibyśmy przetestować możliwości wykorzystania programowania równoległego w badanych algorytmach.

Literatura

[1] Steven Johnson. 18.335J Introduction to Numerical Methods. [online], Spring 2019. [dostęp: 2021-05-9]. 1

[2] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997. 1