

Sprawozdanie - Laboratorium nr 5

Metoda potęgowa z ortogonalizacją Gramma-Schmidta

Hubert Wdowiak, 02.04.2020

1. Wstęp Teoretyczny

Podczas trwania zajęć analizowaliśmy zagadnienie metody potęgowej. Jest to rozwiązanie iteracyjne, pozwalające na znalezienie wartości własnej o największym module, oraz odpowiadający jej wektor własny. Znaleziona wartość jest nazywana - dominującą.

Metoda Potęgowa, jest metodą iteracyjną. Oznacza to, że przy każdym kolejnym przejściu pętli, otrzymywany wynik zyskuje na dokładności. Możliwe jest także znalezienie pozostałych wartości i wektorów własnych, jednak każda kolejna para jest obliczana z coraz to mniejszą precyzją.

1.1 Opis Metody

Przyjmując, że n wektorów własnych macierzy $(\vec{x}_1, \dots, \vec{x}_n)$ \mathbf{A} jest niezależnych liniowo i tworzy bazę przestrzeni liniowej. Dla każdego wektora \vec{v}_0 zachodzi równość:

$$\vec{v}_0 = \sum_{i=1}^n a_i \vec{x}_i, \quad (1)$$

Poprzez λ_i nazywać będziemy i -tą wartość własną macierzy \mathbf{A} .

$$\mathbf{A}\vec{v}_0 = \sum_{i=1}^n a_i \lambda_i \vec{x}_i, \quad (2)$$

$$\vec{v} = \mathbf{A}^m \vec{v}_0 = \sum_{i=1}^n a_i \lambda_i^m \vec{x}_i. \quad (3)$$

W sytuacji, gdy $|\lambda_i|$ ma największą wartość, ze wszystkich modułów wartości własnych macierzy \mathbf{A} , oznacza to, że jest to wartość dominująca. Dla każdego wektora y nieortogonalnego do wektora \vec{x}_1 , prawdziwe są wtedy równania:

$$\lim_{m \rightarrow \infty} (\mathbf{A}^m \vec{v}_0) / \lambda_1^m = \alpha_1 \vec{x}_1, \quad (4)$$

$$\lambda_1 = \lim_{m \rightarrow \infty} (\vec{y}^T \vec{v}_{m+1}) / (\vec{y}^T \vec{v}_m) \quad (5)$$

W ten sposób, za pomocą metody potęgowej, tak jak przepowiedzieliśmy wcześniej, jesteśmy w stanie otrzymać wartość dominującą. Resztę wartości, wraz z odpowiadającymi im wektorami możemy uzyskać stosując metodę Gramma-Schmidta, zwaną też ortogonalizacją Gramma-Schmidta.

Rozwiązuje ona problem spowodowany, przez błędy numeryczne zaokrągleń wektorów rozpatrywanych w problemie. Podstawą konieczną do poprawności rozwiązania jest ich ortogonalność, która z każdym kolejnym wymiarem podprzestrzeni, jest coraz cięższa do spełnienia.

W celu zapewnienia tego warunku, wspomniana powyżej metoda przekształca rozpatrywane wektory w układ wektorów ortogonalnych, w taki sposób by układ sprzed przekształcenia, był tożsamy, z tym po wykonaniu operacji.

2. Zadanie do wykonania

2.1. Opis problemu

W trakcie laboratorium naszym zadaniem było oszacowanie wartości własnych macierzy A , o wymiarze $n = 7$, której elementy określał wzór:

$$A_{ij} = 1 / (\sqrt{2 + |i - j|}). \quad (6)$$

Dokonać to mieliśmy, stosując metodę potęgową, oraz ortogonalizację Gramma-Schmidta.

Kolejnym krokiem było wyznaczenie postaci macierzy D zdefiniowanej jako iloczyn:

$$D = X^T A X, \quad (7)$$

i jej zapis do pliku.

Aby wykonać powyższe polecenia, posłużyliśmy się biblioteką *Numerical Recipes* i jej funkcjami: *matrix* i *vector*, które umożliwiły nam odpowiednie przechowanie danych numerycznych. Poza tymi, nie potrzebowaliśmy, żadnych zewnętrznych funkcji. Samemu zapisaliśmy kilka podprogramów wykonujących podstawowe operacje na macierzach i wektorach, takie jak: mnożenie dwóch macierzy, mnożenie dwóch wektorów, mnożenie macierzy przez wektor.

Zasadę działania programu można opisać następująco:

Dla każdej wartości własnej, na początku inicjalizowaliśmy wektor startowy, złożony z samych jedynek i *IT_MAX* razy (*IT_MAX* = 12, liczba podana w treści zadania) wykonywaliśmy poniższe operacje:

Do “aktualnego przybliżenia wektora k ” - x_k^{i+1} , zapisywaliśmy wartość iloczynu macierzy A oraz “poprzedniego przybliżenia wektora k ” - x_k^i .

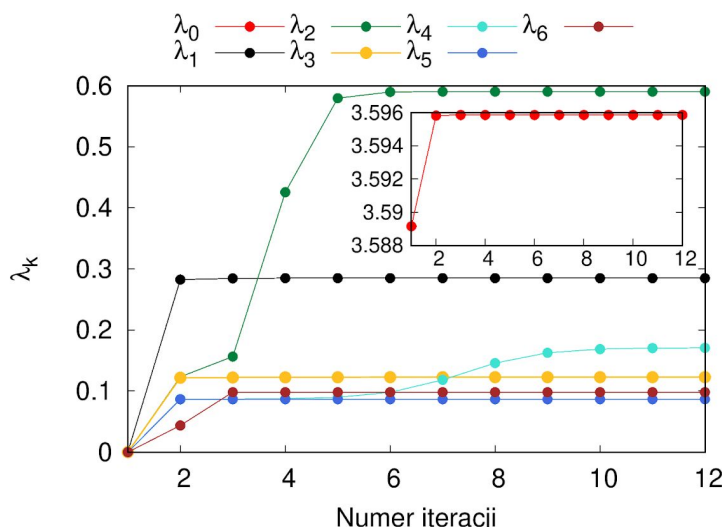
k (numer wyznaczanej wartości własnej) razy, odejmowaliśmy od “obecnego przybliżenia wektora k ” wartość równą iloczynowi transponowanego “obecnego wektora”, oraz j -tego wiersza macierzy X , przy czym j jest równe temu, ile razy powtórzyliśmy tą operację do tej pory.

Obliczaliśmy przybliżenie i -tej wartości własnej , przy czym i jest równe temu, ile razy powtórzyliśmy tą operację do tej pory. “Obecna k -ta wartość własna” jest równa wspomnianemu już “obecnemu przybliżeniu k -tego wektora własnego” pomnożonego przez “poprzednie przybliżenie k -tego wektora”, a następnie podzieleniu otrzymanego wyniku przez iloczyn “poprzedniego przybliżenia k -tego wektora ” z samym sobą.

Nadpisywaliśmy “poprzednie przybliżenie k -tego wektora własnego”. Nową wartość otrzymaliśmy poprzez podzielenie “obecnego przybliżenia k -tego wektora”, przez jego normę, równą pierwiastkowi z jego iloczynu, przez siebie samego.

2.2. Wyniki wartości własnych

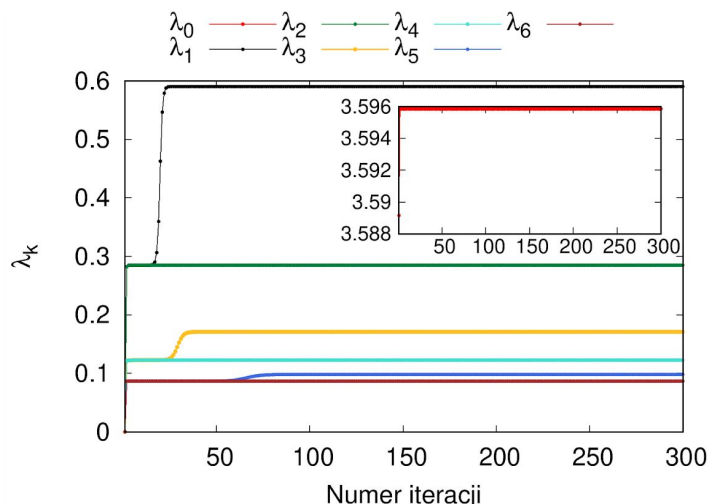
Przy użyciu programu *gnuplot* zobrazowaliśmy wyniki zapisane w pliku *lambda.dat*. Otrzymaliśmy wykres kolejnych przybliżeń wartości własnych w zależności który raz wykonywana jest iteracja.



Wykres 1. Zmiany przybliżeń wartości własnych w zależności od kolejnych iteracji dla $IT_MAX = 12$

Wykres 1 został wykonany dla $IT_MAX = 12$, zatem pętla przybliżająca nas do optymalnych wartości została wykonana stosunkowo niewiele razy. Mimo to każda z przedstawionych wartości staje się niemal równoległa do osi ox , maksymalnie około 6 iteracji. Większość z nich robi to nawet przy drugim - trzecim przejściu pętli.

Jednym z pytań postawionych na laboratorium brzmiało: “Czy 12 to wystarczająca liczba iteracji?”. Aby to sprawdzić ustawiliśmy wartość parametru IT_MAX i wykonaliśmy wykres otrzymanych wyników.



Drugi wykres przyniósł nam rezultaty których można by się było nie spodziewać, patrząc na pierwszy otrzymany wykres. W końcu tam, każda z wartości własnych wydawała się osiągnąć stabilizację, a mimo wszystko po zwiększeniu liczby iteracji okazało się, że niemal każda z wartości osiąga tę stabilizację dużo później. λ_1 , λ_3 oraz λ_5 wyraźnie zwiększają się około iteracji odpowiednio dwudziestej, trzydziestej i siedemdziesiątej.

2.3. Wyniki macierzy D

Wartości otrzymanej macierzy **D**, prezentuje poniższa tabela:

3.59586	-1.49012e-07	1.56928e-07	1.04308e-07	-3.62284e-07	2.98023e-08	-2.95113e-08
-1.63913e-07	0.285003	-0.0021979	-1.8822e-06	1.76951e-08	-4.65661e-09	-8.06904e-09
1.09372e-07	-0.0021979	0.590374	-2.46451e-07	-2.16069e-08	4.62751e-08	-2.08108e-08
1.63913e-07	-1.87568e-06	-2.79455e-07	0.122787	-0.000222209	-0.000131182	-3.60524e-08
-3.57337e-07	-8.76025e-09	-1.81481e-08	-0.000222211	0.170933	-0.00183325	-3.07573e-07
1.11759e-08	-4.37722e-08	3.38187e-08	-0.000131179	-0.00183327	0.0866351	-1.39911e-05
-2.50948e-08	9.27503e-09	-3.9623e-08	-4.96366e-08	-3.12382e-07	-1.40093e-05	0.0981544

Jak widać wartości leżące na głównej przekątnej przypominają wartości własne przedstawione na wykresach 1 i 2, jednak ich kolejność jest inna. Przykładowo: λ_1 na wykresie osiągnęła wartość około 0.59. W powyższej tabeli podobna wartość znajduje się w trzeciej kolumnie od lewej strony, co z pewnością nie odpowiada numerowi '1'. Jest to wynik niestabilności wykonywanej przez nas metody, która to niestabilność wynika z błędów operacji zmiennoprzecinkowych.

3. Wnioski

Poznana przez nas metoda potęgowa połączona z ortogonalizacją Gramma-Schmidta świetnie sprawdza się do szybkiego wyznaczenia wartości własnych macierzy. Jej zaletą jest łatwość zapisu, a także efektywność. Metoda uzyskuje stosunkowo dobre wyniki już przy niewielkiej liczbie iteracji, gdzie w naszym rozpatrywanym przypadku, ponad połowa wartości własnych osiągnęła stabilizację przed szóstym przejściem pętli. Mimo to w obserwowanym przez nas przykładzie, natrafiliśmy na wartość własną, która swoje optimum osiągnęło dopiero około 70 powtórzeń.

Minusem poznanej metody jest fakt, iż przy każdym wykonaniu ortogonalizacji dochodziło do zaburzenia wynikającego z pozostałości poprzednich wartości własnych. Wielokrotne wykonywanie operacji bazujących na wynikach z początkowo niewielkim błędem, skutkuje zaburzeniami kolejnych wyników, podczas dalszego działania programu.