

UNIwersytet Jagielloński

Wydział Matematyki i Informatyki
Kierunek Matematyka w ramach Międzywydziałowych Studiów
Matematyczno – Przyrodniczych

Maciej Żurawski

Odwrotność Drazina

Praca Proseminaryjna

Kraków 2022

Spis treści

1	Wstęp	1
2	Motywacja	2
3	Odwrotność Drazina	6
4	Aspekty numeryczne	9
5	Zastosowania	11
5.1	Efektywna rezystancja	11
5.2	Predykcja połączeń	14
6	Implementacja	16
	Literatura	23

Rozdział 1

Wstęp

Rozszerzony kurs Algebry Liniowej obejmuje zagadnienie pseudoodwrotności danej macierzy $A \in \mathbb{C}_{m,n}$ czyli takiej macierzy $A^+ \in \mathbb{C}_{n,m}$, która spełnia własność:

$$AA^+ = A^+A.$$

Standardowo prezentowane są dwie pseudoodwrotności:

- $X^+ = (A^T A)^{-1} A^T$, przy założeniu linowej niezależności kolumn macierzy A ,
- Pseudoodwrotność Moora-Penrose’a, przy opuszczeniu powyższego założenia.

W pracy zaprezentowana została nowa pseudoodwrotność, nazywana odwrotnością Drazina. W Rozdziale 2. przedstawiona jest motywacja wprowadzenia tego obiektu ukazana w języku algebraicznych równań różniczkowych (DEA). Jednakże sama definicja jest niezależna od tego działu matematyki, dlatego w Rozdziale 3. zdefiniowana została ona w języku algebry.

Korzystanie z tej pseudoodwrotności wiąże się z problemami numerycznymi, których zagadnienie zostało przedstawione w Rozdziale 4. Autor zakłada znajomość czytelnika z rozkładem Schura, który odgrywa kluczową rolę przy implementacji algorytmu obliczania pseudoodwrotności.

Zastosowania odwrotności Drazina rozciągają się poza dziedzinę algebraicznych równań różniczkowych. W Rozdziale 4. zaprezentowane zostało wykorzystanie jej w zagadnieniu obliczania efektywnej rezystancji w obwodach elektrycznych przedstawionych w formalizmie grafów nieskierowanych oraz w problemie predykcji połączeń, który odgrywa znaczącą rolę w algorytmach propozycji znajomych na portalach społecznościowych.

Rozdział 5. zawiera implementację w języku Python algorytmów i przykładów zaprezentowanych w pracy.

Rozdział 2

Motywacja

Rozważmy liniowe algebraiczne równanie różniczkowe (DEA) postaci

$$E\dot{x} = Ax + f(t),$$

gdzie $E, A \in \mathbb{C}_{n,n}$ oraz $f \in \mathcal{C}(I, \mathbb{C}^n)$, wraz z warunkiem początkowym $x(t_0) = x_0$. Równania tej postaci pojawiają się naturalnie w wielu zagadnieniach związanych z linearyzacją układów autonomicznych w matematyce, fizyce i teorii sterowania.

Jeżeli E jest macierzą odwracalną, to otrzymujemy jawną postać rozwiązania powyższego równania

$$x(t) = e^{E^{-1}A(t-t_0)}x_0 + \int_{t_0}^t e^{E^{-1}A(t-s)}f(s)ds.$$

Jednakże w przypadku nieodwracalności E potrzebujemy nowego obiektu pełniącego funkcję pseudoodwrotności.

Definicja 2.1 (Pary równoważne)

Dwie pary macierzy (E_i, A_i) , $E_i, A_i \in \mathbb{C}_{m,n}$, $i = 1, 2$ nazywamy równoważnymi, gdy istnieją odwracalne macierze $P \in \mathbb{C}_{m,m}$ oraz $Q \in \mathbb{C}_{n,n}$, takie że

$$E_2 = PE_1Q, \quad A_2 = PA_1Q.$$

Relację tę zapisujemy $(E_1, A_1) \sim (E_2, A_2)$.

Lemat 2.2

Relacja z Definicji 2.1 jest relacją równoważności.

Dowód:

Zwrotność otrzymujemy biorąc $P = I_m$ oraz $Q = I_n$.

Symetryczność uzyskujemy z odwracalności P i Q .

Przechodniość wynika z odwracalności iloczynu macierzy odwracalnych.



Definicja 2.3 (Para regularna)

Niech $E, A \in \mathbb{C}_{m,n}$. Parę macierzy (E, A) nazywamy regularną jeżeli $m = n$ oraz wielomian charakterystyczny p zdefiniowany jako

$$p(\lambda) = \det(\lambda E - A)$$

nie jest wielomianem zerowym.

Lemat 2.4

Każda para macierzy równoważna regularnej parze macierzy jest regularna.

Dowód:

Istnieją odwracalne P i Q takie że $E_2 = PE_1Q$, $A_2 = PA_1Q$.

Wtedy

$$\tilde{p}(\lambda) = \det(\lambda E_2 - A_2) = \det(\lambda PE_1Q - PA_1Q) = \det(P) \cdot \det(\lambda E_1 - A_1) \cdot \det(Q) = c \cdot p(\lambda),$$

dla pewnego $c \neq 0$. ■

Twierdzenie 2.5

Niech $E, A \in \mathbb{C}_{n,n}$ oraz (E, A) niech będzie regularna. Wtedy

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix} \right),$$

gdzie J jest macierzą w kanonicznej postaci Jordana, N jest macierzą nilpotentną w kanonicznej postaci Jordana, a także bloki N i I mogą być nieobecne.

Dowód:

Para (E, A) jest regularna więc istnieje $\lambda_0 \in \mathbb{C}$, takie że $\det(\lambda_0 E - A) \neq 0$ co oznacza, że $\lambda_0 E - A$ jest odwracalna. Zatem

$$(E, A) \sim (E, A - \lambda_0 E + \lambda_0 E) \sim ((A - \lambda_0 E)^{-1} E, I - \lambda_0 (A - \lambda_0 E)^{-1} E).$$

$(A - \lambda_0 E)^{-1} E$ przekształcamy do postaci kanonicznej Jordana $\text{diag}(\tilde{J}, \tilde{N})$, gdzie \tilde{J} jest odwracalna oraz \tilde{N} jest nilpotentna ściśle górnorójkątna. Otrzymujemy więc

$$(E, A) \sim \left(\begin{bmatrix} \tilde{J} & 0 \\ 0 & \tilde{N} \end{bmatrix}, \begin{bmatrix} \tilde{I} + \lambda_0 \tilde{J} & 0 \\ 0 & \tilde{I} + \lambda_0 \tilde{N} \end{bmatrix} \right).$$

Korzystając z formy $\tilde{N}, I + \lambda_0 \tilde{N}$ jest odwracalną górnorójkątną macierzą. Otrzymujemy

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & (I + \lambda_0 \tilde{N})^{-1} \tilde{N} \end{bmatrix}, \begin{bmatrix} \tilde{J}^{-1} + \lambda_0 I & 0 \\ 0 & I \end{bmatrix} \right),$$

gdzie $(I + \lambda_0 \tilde{N})^{-1} \tilde{N}$ jest także ściśle górnorójkątną macierzą, a przez to nilpotentną. Ponowna transformacja nietrywialnych bloków do kanonicznej postaci Jordana daje tezę. ■

Definicja 2.6 (Indeks pary)

Rozważamy regularną parę (E, A) . Jeżeli w formie kanonicznej z Twierdzenia 2.5 występuje blok N , to liczba v spełniająca warunek

$$N^v = 0, \quad N^{v-1} \neq 0$$

nazywamy indeksem pary (E, A) . Jeżeli blok N nie występuje to przyjmujemy $v = 0$. Indeks pary będziemy oznaczać $\text{ind}(E, A)$.

W celu dobrej określoności indeksu pary musimy wykazać, że nie zależy on od transformacji do postaci kanonicznej.

Lemat 2.7

Załóżmy, że para (E, A) , gdzie $E, A \in \mathbb{C}_{n,n}$ ma dwie postacie kanoniczne

$$(E, A) \sim \left(\begin{bmatrix} I & 0 \\ 0 & N_i \end{bmatrix}, \begin{bmatrix} J_i & 0 \\ 0 & I \end{bmatrix} \right), i = 1, 2$$

gdzie $d_i, i = 1, 2$ jest rozmiarem bloku J_i . Wtedy $d_1 = d_2$ i ponadto $N_1^v = 0$ oraz $N_1^{v-1} = 0$ wtedy i tylko wtedy gdy $N_2^v = 0$ oraz $N_2^{v-1} = 0$.

Dowód: Dla wielomianów charakterystycznych dwóch postaci kanonicznych mamy

$$\rho_i(\lambda) = \det \begin{pmatrix} \lambda I - J_i & 0 \\ 0 & \lambda N_i - I \end{pmatrix} = (-1)^{n-d_i} \det(\lambda I - J_i).$$

Zatem p_i jest wielomianem stopnia d_i . Ponieważ postaci normalne są silnie równoważne p_1 i p_2 różnią się jedynie o stały czynnik, co wynika z Lematu 2.4. Zatem $d_1 = d_2$, więc rozmiary bloków są takie same. Ponadto z silnej równoważności form kanonicznych otrzymujemy istnienie odwracalnych macierzy

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

takich że

$$P \begin{bmatrix} I & 0 \\ 0 & N_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & N_1 \end{bmatrix} Q$$

oraz

$$P \begin{bmatrix} J_2 & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} J_1 & 0 \\ 0 & I \end{bmatrix} Q.$$

Zatem otrzymujemy zależności $P_{11} = Q_{11}, P_{12}N_2 = Q_{12}, P_{21} = N_1Q_{21}, P_{22}N_2 = N_1Q_{22}, P_{11}J_2 = J_1Q_{11}, P_{12} = J_1Q_{12}, P_{21} = J_1Q_{12}, P_{22} = Q_{22}$.

Z powyższych równości $P_{21} = N_1P_{21}J_2$ i poprzez kolejne podstawienia pod P_{21} otrzymujemy $P_{21} = 0$ dzięki nilpotentności N_1 , zatem P_{11} i P_{22} muszą być nieosobliwe. W szczególności J_1 i J_2 oraz N_1 i N_2 są podobne. Zatem postaci Jordana N_1 i N_2 zawierają taką samą liczbę nilpotentnych bloków Jordana. ■

Wróćmy teraz do algebraicznych równań różniczkowych. Przyjmijmy $A = I$, $f \equiv 0$, oraz E nieodwracalna. Wtedy równanie przyjmuje postać

$$E\dot{x} = x$$

Rozważmy parę (E, I) . Jest ona regularna (wystarczy wziąć $\lambda_0 = 0$ by przekonać się, że $\det(\lambda_0 E - I) = \det(-I) \neq 0$ zatem możemy ją sprowadzić do postaci z Twierdzenia 2.5. Rozpocznijmy od transformacji macierzy E do kanonicznej postaci Jordana

$$E = T^{-1}JT, \quad \text{gdzie } J = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}.$$

Przekształcając wyjciowe równanie

$$\begin{aligned} E\dot{x} &= x \\ TET^{-1}T\dot{x} &= TT^{-1}Tx \end{aligned}$$

Dla wygody połóżmy $y := Tx$

$$\begin{aligned} \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ \begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} &= \begin{bmatrix} C^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \end{aligned}$$

Z nilpotentności N otrzymujemy

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} e^{C^{-1}(t-t_0)} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} y_1(t_0) \\ y_2(t_0) \end{bmatrix}.$$

Definiując $J^D := \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix}$ możemy przepisać powyższy wynik w postaci

$$y(t) = e^{J^D(t-t_0)}y(t_0)$$

Wracając do zmiennej x

$$x(t) = T^{-1}e^{J^D(t-t_0)}Tx(t_0) = e^{E^D(t-t_0)}x(t_0),$$

gdzie $E^D := T^{-1}J^DT$.

Widzimy więc, że E^D odpowiada E^{-1} w przypadku gdyby E było odwracalne. Zatem E^D pełni rolę pseudoodwrotności (zauważmy, że spełniony jest warunek $EE^D = E^DE$). Nazywamy ją odwrotnością Drazina, lecz w następnym rozdziale zdefiniujemy ją formalnie w sposób bardziej subtelny.

Rozdział 3

Odwrotność Drazina

Definicja 3.1 (Indeks macierzy)

Niech $A \in \mathbb{C}_{n,n}$. Indeks macierzy A nazywamy najmniejszą liczbę naturalną k , taką że $rk(A^{k+1}) = rk(A^k)$. Będziemy stosować oznaczenie $ind(A)$.

Uwaga 3.2

Przy powyższej definicji indeks macierzy jest poprawnie zdefiniowany. Załóżmy że nie istnieje taka liczba naturalna k , że $rk(A^{k+1}) = rk(A^k)$. Zdefiniujmy $j := rk(A)$. Wtedy $rk(A^j) = 0$ lecz również $rk(A^{j+1}) = 0$, gdyż $rk(AB) \leq \min(rk(A), rk(B))$. Zatem doszliśmy do sprzeczności.

Definicja 3.3 (Odwrotność Drazina)

Niech $A \in \mathbb{C}_{n,n}$ ma indeks k . Macierz $X \in \mathbb{C}_{n,n}$ spełniającą warunki

1. $AX = XA$
2. $XAX = X$
3. $XA^{k+1} = A^k$

nazywamy odwrotnością Drazina macierzy A i oznaczamy przez A^D .

Twierdzenie 3.4 (Jedyność odwrotności Drazina)

Niech $A \in \mathbb{C}_{n,n}$. Wtedy istnieje dla niej odwrotność Drazina i jest ona jedyna.

Dowód:

Dla macierzy A weźmy macierz $A^D = T^{-1}J^DT$ z końca poprzedniego rozdziału. Spełnia ona wymagane warunki z definicji odwrotności Drazina.

$$\begin{aligned} 1. \quad A^DA &= T^{-1}J^DTT^{-1}JT = T^{-1} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} T = T^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} T. \\ AA^D &= T^{-1}JTT^{-1}J^DT = T^{-1} \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} T = T^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} T. \end{aligned}$$

$$2. A^D A A^D = T^{-1} J^D T T^{-1} J T T^{-1} J^D T = T^{-1} J^D J J^D T =$$

$$T^{-1} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} T = T^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} C^{-1} & 0 \\ 0 & N \end{bmatrix} T =$$

$$T^{-1} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} T = A^D$$

$$3. v = \text{ind}(E, I) = \text{ind}(J, I) = \text{ind}\left(\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix}, \begin{bmatrix} C^{-1} & 0 \\ 0 & I \end{bmatrix}\right) = \text{ind}(N, I),$$

zatem $N^v = 0$ oraz $N^{v-1} \neq 0$.

$$A^D A^{v+1} = T^{-1} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} T T^{-1} \begin{bmatrix} C^{v+1} & 0 \\ 0 & N^{v+1} \end{bmatrix} = T^{-1} \begin{bmatrix} C^v & 0 \\ 0 & 0 \end{bmatrix} T.$$

$$A^v = T^{-1} \begin{bmatrix} C^v & 0 \\ 0 & N^v \end{bmatrix} T = T^{-1} \begin{bmatrix} C^v & 0 \\ 0 & N \end{bmatrix} T$$

Aby pokazać jedyność założymy, że X_1 i X_2 są dwiema odwrotnościami Drazina macierzy A . Korzystając z każdego z warunków z definicji

$$\begin{aligned} X_1 &\stackrel{2}{=} X_1 A X_1 \stackrel{1}{=} X_1 X_1 A \stackrel{2}{=} X_1 X_1 X_1 A X_1 \stackrel{1}{=} X_1^2 X_1 A^2 = \dots = X_1^v X_1 A^v = X_1^{v+1} A^v \\ &\stackrel{3}{=} X_1^{v+1} X_2 A^{v+1} \stackrel{1}{=} X_1^{v+1} A X_2 A^v = \dots = X_1^{v+1} A^{v+1} X_2 = X_1^v X_1 A A^v X_2 \\ &\stackrel{1}{=} X_1^v A X_1 A^v X_2 X = X_1^{v-1} X_1 A X_1 A^v X_2 \stackrel{2}{=} X_1^{v-1} X A^v X_2 = X_1^v A^v X_2 = \dots = X_1 A X_2 = \\ &\dots = X_1 A^{v+1} X_2^{v+1} \stackrel{3}{=} A^v X_2^{v+1} = A^{v-1} A X_2 X_2^v \stackrel{1}{=} A^{v-1} X_2 A X_2 X_2^{v-1} \stackrel{2}{=} A^{v-1} X_2^v = \dots \\ &= X_2 \end{aligned}$$

■

Przykład 3.5

1. Jeżeli A jest odwracalna to jej odwrotnością Drazina jest A^{-1} .

2. Jeżeli A jest macierzą nilpotentną to $A^D = 0$.

3. Jeżeli A jest macierzą idempotentną to $A^D = A$.

$$4. A = \begin{bmatrix} 2 & 1 & -1 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}, \text{ wtedy } \text{ind}(A) = 2$$

$$A = T^{-1} \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} T, \text{ gdzie } T = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{4} & -\frac{3}{4} & \frac{1}{4} \end{bmatrix}$$

zatem

$$A^D = T^{-1} \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} T = \begin{bmatrix} \frac{3}{4} & \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix}.$$

Celem tej pracy nie jest dogłębna analiza problemu $E\dot{x} = Ax + f(t)$, służy on tu do przedstawienia motywacji zdefiniowania odwrotności Drazina, która naturalnie się pojawia przy analizie problemu. Z tego powodu następne twierdzenie ukazujące zastosowanie tej pseudoodwrotności do powyższego równania (choć niepraktyczne pod względem numerycznym) przedstawię bez dowodu.

Twierdzenie 3.6

Niech $E, A \in \mathbb{C}_{n,n}$ komutują oraz niech $\ker E \cap \ker A = 0$. Ponadto niech $f \in \mathcal{C}^v(I, \mathbb{C}^n)$, $v = \text{ind}(E)$ oraz $t_0 \in I$. Wtedy każde rozwiązanie $x \in \mathcal{C}^1(I, \mathbb{C}^n)$ problemu $E\dot{x} = Ax + f(t)$ ma postać

$$x(t) = e^{E^D A(t-t_0)} E^D E v + \int_{t_0}^t e^{E^D A(t-s)} E^D f(s) ds - (I - E^D E) \sum_{i=0}^{v-1} (EA^D)^i A^D f^{(i)}(t).$$

Uwaga 3.7

Przy założeniu regularności pary (E, A) możemy opuścić założenie o komutacji E i A wybierając $\tilde{\lambda} \in \mathbb{C}$ taką by $\tilde{\lambda}E - A$ było odwracalne. Wtedy macierze

$$\tilde{E} = (\tilde{\lambda}E - A)^{-1}E, \quad \tilde{A} = (\tilde{\lambda}E - A)^{-1}A$$

komutują.

Wymnażając więc równanie $E\dot{x} = Ax + f(t)$ obustronnie przez $(\tilde{\lambda}E - A)^{-1}$, otrzymujemy równanie dla którego możemy zastosować Twierdzenie 3.6.

Rozdział 4

Aspekty numeryczne

W celu numerycznego obliczenia odwrotności Drazina przede wszystkim potrzebujemy rozłożyć ją do postaci

$$A = T^{-1} \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix} T$$

Z powodu błędów numerycznych arytmetyki zmiennoprzecinkowej w praktyce niemożliwe jest doprowadzenie macierzy do takiej postaci, gdyż błąd zaokrąglenia uniemożliwia utworzenie się bloków Jordana (teoretyczne równe wartości własne będą się od siebie różnić). Zatem potrzebujemy alternatywnego sposobu obliczania odwrotności Drazina.

W tym celu użyjemy rozkładu Schura macierzy A , która zwróci nam postać

$$A = QZQ^{-1},$$

gdzie Q jest macierzą ortogonalną, Z macierzą górnotrójkątną z wartościami własnymi macierzy A na diagonalu.

W ogólności rozkład Schura nie jest jednoznacznie wyznaczony, gdyż wartości własne na diagonalu macierzy Z mogą być przepermutowane. W celu znalezienia C, N, T będziemy stosować rozkład Schura macierzy A dwukrotnie porządkując wartości własne na dwa różne sposoby.

Za pierwszym razem posortujemy je w taki sposób, by niezerowe wartości własne pojawiły się jako pierwsze na diagonalu Z . Następnie jeżeli k jest liczbą niezerowych wartości własnych to blok $k \times k$ z lewej górnej części macierzy tworzy nam macierz M , a k pierwszych kolumn Q tworzy k pierwszych kolumn macierzy T .

Za drugim razem ustawiamy zerowe wartości własne jako pierwsze na diagonalu Z . Następnie bloki $(n - k) \times (n - k)$ tworzą nilpotentną macierz N , a pierwsze $n - k$ kolumn Q tworzy ostatnie $n - k$ kolumn T . Następnie odwrotność Drazina otrzymujemy poprzez

$$A^D = T^{-1} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} T$$

Algorithm 1 Drazin inverse

```

1: procedure DRAZIN( $A, tol$ )
2:    $(n, n) \leftarrow \text{shape}(A)$ 
3:    $T_1, Z_1, k_1 \leftarrow \text{schur}(A, |x| > tol)$  ▷ Zerowe wartości własne na początku
4:    $T_2, Z_2, k_2 \leftarrow \text{schur}(A, |x| \leq tol)$  ▷ Zerowe wartości własne na końcu
5:    $U \leftarrow [Z_{1, :, k_1} | Z_{2, :, n-k_1}]$ 
6:    $U^{-1} \leftarrow \text{inverse}(U)$ 
7:    $C \leftarrow U^{-1}AU$  ▷ Wyłuskanie macierzy  $C$ 
8:    $Z \leftarrow 0_{n \times n}$ 
9:   if  $k_1 \neq 0$  then
10:      $C^{-1} \leftarrow \text{inverse}(C_{:, k_1, :k_1})$ 
11:      $Z_{:, k_1, :k_1} \leftarrow C^{-1}$ 
12:   return  $UZU^{-1}$ 

```

Uwaga 4.1

Powinniśmy pamiętać, że w algorytmie obracamy macierz U , zatem dla źle uwarunkowanej U algorytm może zwracać niepoprawny wynik.

Dla macierzy Hilberta $(H)_{ij} := \frac{1}{1+i+j}$ wymiaru już 4 powyższy algorytm zaimplementowany w języku Python korzystający z metody odwracania macierzy z biblioteki *scipy* zwraca niepoprawny wynik dla parametru tolerancji $tol = 0.0001$ (Rozdział 5).

Rozdział 5

Zastosowania

5.1 Efektywna rezystancja

Oprócz przykładu przedstawionego w Rozdziale 3. jesteśmy w stanie znaleźć inne zastosowania odwrotności Drazina.

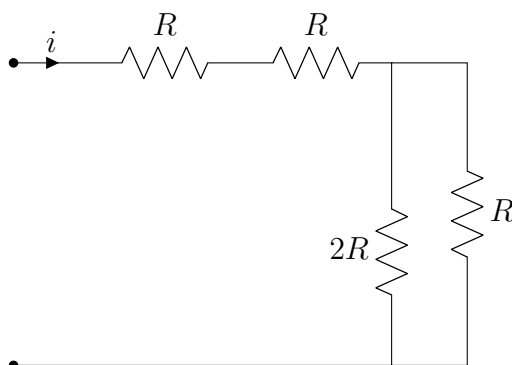
W teorii elektryczności i magnetyzmu klasycznym problemem jest obliczenie efektywnej rezystancji (wartości spadku napięcia przy stałym natężeniu prądu elektrycznego) w obwodzie elektrycznym między dowolnymi punktami. Standardowe metody są nieefektywne w bardziej skomplikowanych obwodach. Jednakże odwrotność umożliwia rozwiązanie tego problemu.

Metoda ta opiera się na reprezentacji obwodu elektrycznego za pomocą formalizmu grafów nieskierowanych, w których węzłom obwodu odpowiadają wierzchołki, a rezystorom krawędzie i wagi.

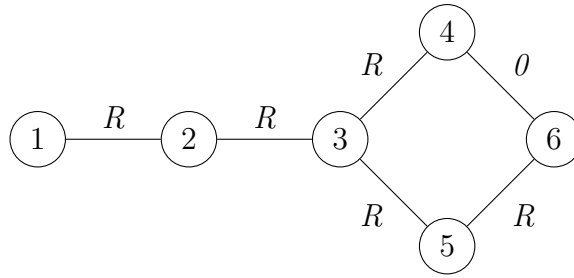
Na problem możemy również spojrzeć z perspektywy błędzenia losowego na grafie. Będzie to wtedy (z dokładnością do stałej) średni czas, który zajmuje błądzącemu przejście z jednego wierzchołka do drugiego i z powrotem.

Przykład 5.1

Układ rezystorów



możemy zaprezentować jako graf



Definicja 5.2 (Macierz laplasjanu)

Niech $\mathcal{G} = (\mathcal{N}, \mathcal{L}, \omega)$ będzie nieskierowanym spójnym grafem z wagami, o wierzchołkach w_1, \dots, w_n . Wtedy następująco zdefiniowana macierz

$$L_{ij} := \begin{cases} \deg(w_i) & \text{gdy } i = j \\ -1 & \text{gdy } i \neq j \text{ oraz } \{w_i, w_j\} \text{ jest krawędzią} \\ 0 & \text{w przeciwnym razie} \end{cases},$$

nazywamy macierzą laplasjanu. Formuła ta determinuje bijekcję między zbiorem macierzy laplasjanu, a zbiorem grafów \mathcal{G} .

Własność 5.3

Niech L będzie macierzą laplasjanu. Wtedy wprost z definicji wynikają następujące własności:

1. L jest symetryczna,
2. $\forall i \neq j : L_{ij} \leq 0$,
3. $\sum_j L_{ij} = \sum_i L_{ij} = 0$.

Własność 5.4

Niech L będzie macierzą laplasjanu.

1. L jest nieujemnie określona,
2. 0 jest wartością własną L i odpowiada jej wektor własny $[1, \dots, 1]^T$.

Dowód:

1. Weźmy dowolny $x \in \mathbb{R}^n$

$$x^T L x = \sum_{(i,j) \in \mathcal{L}} w_{ij} (x_i - x_j)^2 \geq 0.$$

2. Z Własności 5.3.3 otrzymujemy, że kolumny L są liniowo zależne, zatem L jest nieodwracalna, czyli 0 jest wartością własną L . Biorąc wektor złożony z samych jedynek i znów korzystając z Własności 5.3.3 jest to wektor własny odpowiadający wartości własnej 0 .

■

Z uwagi na długi dowód poniższe twierdzenie przedstawię bez dowodu. Można go znaleźć w pracy [2].

Twierdzenie 5.5

Niech L będzie macierzą Laplasjanu. Wtedy efektywna rezystancja między punktami i oraz j wynosi

$$\omega_{ij} = (e_i - e_j)^T L^D (e_i - e_j), \text{ gdzie } e_i \text{ oznacza } i\text{-ty wektor bazy kanonicznej.}$$

Macierz $(\Omega)_{ij} = \omega_{ij}$ nazywamy macierzą rezystancji.

Następny przykład jest szczególnie ważny dla autora, gdyż spotkał się z nim w szkole średniej i przyporzął mu znaczącą trudność przy korzystaniu z klasycznych metod.

Przykład 5.6

Rozważmy szkielet sześciangu z rezystorami o tej samej rezystancji ($R = 1$) na każdej krawędzi. Jesteśmy w stanie obliczyć efektywną rezystancję między dowolnymi wierzchołkami.

Macierz Laplasjanu tego układu wygląda następująco

$$L = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 3 & -1 & 0 & -1 \\ 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & -1 & 0 & -1 & 3 \end{bmatrix}.$$

Wtedy macierz rezystancji wynosi

$$\Omega = \frac{1}{12} \begin{bmatrix} 0 & 7 & 9 & 7 & 7 & 9 & 10 & 9 \\ 7 & 0 & 7 & 9 & 9 & 7 & 9 & 10 \\ 9 & 7 & 0 & 7 & 10 & 9 & 7 & 9 \\ 7 & 9 & 7 & 0 & 9 & 10 & 9 & 7 \\ 7 & 9 & 10 & 9 & 0 & 7 & 9 & 7 \\ 9 & 7 & 9 & 10 & 7 & 0 & 7 & 9 \\ 10 & 9 & 7 & 9 & 9 & 7 & 0 & 7 \\ 9 & 10 & 9 & 7 & 7 & 9 & 7 & 0 \end{bmatrix}.$$

Uwaga 5.7

W implementacji autor stosuje zadanie macierzy Laplasjanu poprzez macierz sąsiedztwa

$$(A)_{ij} := \begin{cases} 1 & \text{gdzie } i \neq j \text{ oraz } \{w_i, w_j\} \text{ jest krawędzią} \\ 0 & \text{w przeciwnym razie} \end{cases}$$

oraz macierz diagonalną postaci

$$D := \text{diag}(\sum_j (A)_{1j}, \dots, \sum_j (A)_{nj})$$

Wtedy

$$L = D - A$$

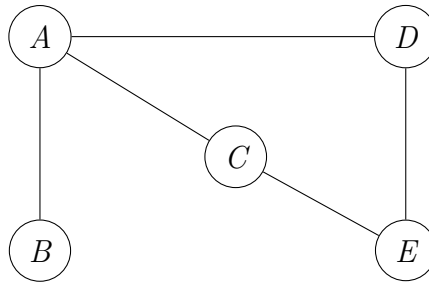
5.2 Predykcja połączeń

Predykcją połączeń jest problem przewidywania prawdopodobnych krawędzi dwóch niepołączonych wierzchołków w grafie. Naturalnym zastosowaniem tego konceptu jest problem propozycji znajomych w portalach społecznościowych. Taka sieć znajomych może być zaprezentowana jako graf, w którym wierzchołki oznaczają użytkowników portalu, a krawędzie status znajomego.

Możemy wykorzystać pojęcie efektywnej rezystancji jako metrykę między dwoma wierzchołkami, to znaczy mniejsza efektywna rezystancja między dwoma wierzchołkami wskazuje na większe prawdopodobieństwo utworzenia bezpośredniego połączenia między nimi.

Przykład 5.8

Rozważmy sieć 5-ciu znajomych:



Macierz Laplasjanu ma postać

$$L = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix},$$

a macierz rezystancji

$$\Omega = \begin{bmatrix} 0 & 1 & 0.75 & 0.75 & 1 \\ 1 & 0 & 1.75 & 1.75 & 2 \\ 0.75 & 1.75 & 0 & 1 & 0.75 \\ 0.75 & 1.75 & 1 & 0 & 0.75 \\ 1 & 2 & 0.75 & 0.75 & 0 \end{bmatrix}.$$

Aby uwzględnić obecne połączenia w grafie możemy wymnożyć po współrzędnych macierz (iloczyn Hadamarda "•") Ω przez $J_5 - A$, gdzie J_n oznacza $n \times n$ wymiarową macierz złożoną z samych jedynek, a A oznacza macierz dopasowania z Uwagi 5.7. Ta operacja wyzeruje nam pozycje, dla których obecne jest już połączenie.

$$\Omega \bullet (J_5 - A) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1.75 & 1.75 & 2 \\ 0 & 1.75 & 0 & 1 & 0 \\ 0 & 1.75 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \end{bmatrix}.$$

Zatem proces propozycji znajomych powinien wyglądać następująco (użytkownikowi po lewej stronie strzałki proponowany jest użytkownik po prawej stronie)

- $"A" \rightarrow "E"$
- $"B" \rightarrow "C" \text{ lub } "D"$
- $"C" \rightarrow "D"$
- $"D" \rightarrow "C"$
- $"E" \rightarrow "A"$

Rozdział 6

Implementacja

```
import numpy as np
from scipy import linalg as la
```

```
import csv
```

Pierwsza funkcja służy nam do obliczenia indeksu macierzy

```
def index(A, tol=1e-5):
    """Obliczanie indeksu macierzy A.

    Parameters:
        A ((n,n) ndarray): Macierz nxn.

    Returns:
        k (int): Indeks A.
    """

    # test na nieodwracalność
    if not np.isclose(la.det(A), 0):
        return 0

    n = len(A)
    k = 1
    Ak = A.copy()
    while k <= n:
        r1 = np.linalg.matrix_rank(Ak)
        r2 = np.linalg.matrix_rank(np.dot(A, Ak))
        if r1 == r2:
            return k
        Ak = np.dot(A, Ak)
        k += 1

    return k
```

Poniższa funkcja sprawdza, czy potencjalna macierz Ad jest odwrotnością Drazina macierzy A .

```
def is_drazin(A, Ad, k):
    """Weryfikacja czy macierz Ad jest odwrotnoscia Drazina macierzy A.

    Parameters:
        A ((n,n) ndarray): Macierz nxn.
        Ad ((n,n) ndarray): Kandydat na odwrotnosc Drazina A.
        k (int): Indeks macierzy A.

    Returns:
        (bool) True jezeli Ad jest odwrotnosci A, False w przeciwnym razie.
    """

    if np.allclose(A @ Ad, Ad @ A) and np.allclose(Ad @ A @ Ad, Ad) and
        np.allclose(Ad @ np.linalg.matrix_power(A, k+1),
                    np.linalg.matrix_power(A,k)):
        return True
    else:
        return False
```

Sprawdzenie poprawności wyniku z Przykładu 3.5.4.

```
In: A = np.array([[2,1,-1],[0,1,1],[1,2,1]])
    Ad = np.array([[0.75, 0.5, -0.25],[-0.25, 0, 0.25],[0, 0.25, 0.25]])
    k = index(A)

    is_drazin(A,Ad, k)
```

Out: True

Metoda obliczania odwrotności Drazina za pomocą rozkładu Schura.

```
f = lambda x: abs(x) > 0
T1,Z1,k = la.schur(A, sort=f)

def drazin_inverse(A, tol=1e-4):
    """Obliczanie odwrotności Drazina za pomoca rozkładu Schura A.

    Parameters:
        A ((n,n) ndarray): Macierz nxn.

    Returns:
        ((n,n) ndarray) Odwrotnosc Drazina A.
    """
```

```
n = np.shape(A)[0]
f1 = lambda x: abs(x) > tol
f2 = lambda x: abs(x) < tol
Z1, Q1, k1 = la.schur(A, sort=f1)
Z2, Q2, k2 = la.schur(A, sort=f2)
Q = np.concatenate((Q1[:, :k1], Q2[:, :n-k1]), 1)
Qinv = la.inv(Q)
V = Qinv @ A @ Q
Z = np.zeros((n, n))
if k1 != 0:
    M1 = la.inv(V[:, :k1])
    Z[:, :k1] += M1

return Q@Z@Qinv
```

Kontrola, czy funkcja zwróci ten sam wynik jak w Przykładzie 3.5.4.

```
In: A = np.array([[2, 1, -1], [0, 1, 1], [1, 2, 1]])
    drazin_inverse(A)
```

```
Out: array([[ 7.50000000e-01,  5.00000000e-01, -2.50000000e-01],
            [-2.50000000e-01,  3.46944695e-17,  2.50000000e-01],
            [-1.66533454e-16,  2.50000000e-01,  2.50000000e-01]])
```

Przykład macierzy Hilberta wymiaru 4 z Uwagi 4.1.

```
In: B = la.hilbert(4)
    Bd = drazin_inverse(B, 1e-4)

    print(is_drazin(B, Bd, index(B)))
```

```
Out: False
```

Metoda obliczania macierzy rezystancji podając jako argument macierz sąsiedztwa.

```
def effective_resistance(A):
    """Obliczanie efektywnej rezystancji kazdego wierzchołka grafu.

    Parameters:
        A ((n,n) ndarray): Macierz sasiedztwa A.

    Returns:
        ((n,n) ndarray) Macierz rezystancji.
    """
```

```
n = np.shape(A)[0]
D = np.zeros((n,n))
for i in range(n):
    D[i,i] += sum(A[i,:])

L = D - A

A_effective = np.zeros((n,n))
for i in range(n):
    for j in range(n):
        if i != j:
            L_tilde = L.copy()
            L_tilde[j,:] = np.eye(n)[j,:]
            A_effective[i,j] += drazin_inverse(L_tilde)[i,i]

return A_effective
```

Przykład 5.6.

```
In: B = np.array([[0,1,0,1,1,0,0,0],[1,0,1,0,0,1,0,0],
[0,1,0,1,0,0,1,0],[1,0,1,0,0,0,0,1],
[1,0,0,0,0,1,0,1],[0,1,0,0,1,0,1,0],
[0,0,1,0,0,1,0,1],[0,0,0,1,1,0,1,0]])
effective_resistance(B)
```

```
Out: array([[0.          , 0.58333333, 0.75          , 0.58333333, 0.58333333,
0.75          , 0.83333333, 0.75          ],
[0.58333333, 0.          , 0.58333333, 0.75          , 0.75          ,
0.58333333, 0.75          , 0.83333333],
[0.75          , 0.58333333, 0.          , 0.58333333, 0.83333333,
0.75          , 0.58333333, 0.75          ],
[0.58333333, 0.75          , 0.58333333, 0.          , 0.75          ,
0.83333333, 0.75          , 0.58333333],
[0.58333333, 0.75          , 0.83333333, 0.75          , 0.          ,
0.58333333, 0.75          , 0.58333333],
[0.75          , 0.58333333, 0.75          , 0.83333333, 0.58333333,
0.          , 0.58333333, 0.75          ],
[0.83333333, 0.75          , 0.58333333, 0.75          , 0.75          ,
0.58333333, 0.          , 0.58333333],
[0.75          , 0.83333333, 0.75          , 0.58333333, 0.58333333,
0.75          , 0.58333333, 0.          ]])
```

Klasa do operowania na sieciach z Rozdziału 5.2.

```
class LinkPredictor:
    """Predykcja polaczen wierzcholkow w sieci."""

    def __init__(self, filename):
        with open(filename, newline='') as f:
            reader = csv.reader(f)
            data = list(reader)
            self.data = data

            n = len(self.data)
            people = []
            for i in range(n):
                person = data[i][0]
                person2 = data[i][1]
                if person not in people:
                    people.append(person)
                if person2 not in people:
                    people.append(person2)

            self.names = people

            m = len(self.names)
            A = np.zeros((m,m))

            for k in range(n):
                pair = self.data[k]
                i = self.names.index(pair[0])
                j = self.names.index(pair[1])
                A[i,j] += 1
                A[j,i] += 1

            self.adjacent = A

            self.resistance = effective_resistance(self.adjacent)

    def predict_link(self, node=None):

        """Predykcja następnego polaczenia, albo dla calego grafu, albo dla
        pojedynczego wierzcholka.

        Parameters:
            node (str): Nazwa wierzcholka w sieci.
```

Returns:

```
node1, node2 (str): Nazwy wierzchołków mające być połączone
                     Zwrocone jeśli node równa się None
node1 (str): Nazwa wierzchołka przewidzianego do połączenia
              odpowiadająca node.
              Zwrocone jeśli node nie jest równy None.
```

Raises:

```
ValueError: Jeśli node nie jest w grafie.
```

```
"""
```

```
if node == None:
    L = []
    for ind, person in enumerate(self.names):
        L.append((person,
                  self.names[np.argmin(np.where(self.resistance * (1 -
                  self.adjacent) == 0, np.inf, self.resistance * (1 -
                  self.adjacent))[ind]))))
    return L
if node not in self.names:
    raise ValueError("Osoba {} nie znajduje się w
                     sieci.".format(node))

else:
    return self.names[np.argmin(np.where(self.resistance * (1 -
    self.adjacent) == 0, np.inf, self.resistance * (1 -
    self.adjacent))[self.names.index(node)])]
```

```
def add_link(self, node1, node2):
    """Dodanie połączenia w grafie między node1 oraz node2
    poprzez aktualizację macierzy sąsiedztwa oraz macierzy rezystancji
```

Parameters:

```
node1 (str): Nazwa wierzchołka w sieci.
node2 (str): TNazwa wierzchołka w sieci.
```

Raises:

```
ValueError: Jeżeli node1 lub node2 nie znajdują się w grafie.
```

```
"""
```

```
if node1 not in self.names:
    raise ValueError("Osoba {} nie znajduje się w
                     sieci.".format(node1))
```

```
if node1 not in self.names:
    raise ValueError("Osoba {} nie znajduje się w
                      sieci.".format(node2))

i = self.names.index(node1)
j = self.names.index(node2)
A = self.adjacent
A[i,j] += 1
A[j,i] += 1
self.adjacent = A
self.resistance = effective_resistance(self.adjacent)
```

Przykład 5.8.

In: net = LinkPredictor('net.csv')

```
print(net.resistance*(1 - net.adjacent))
print("-" * 30)
print(net.predict_link())
```

Out:

```
[[0.  0.  0.  0.  1. ]
 [0.  0.  1.75 1.75 2. ]
 [0.  1.75 0.  1.  0. ]
 [0.  1.75 1.  0.  0. ]
 [1.  2.  0.  0.  0. ]]
```

```
-----
[('A', 'E'), ('B', 'D'), ('D', 'C'), ('C', 'D'), ('E', 'A')]
```

Źródła

- [1] Peter Kunkel i Volker Mehrmann, *Differential-Algebraic Equations Analysis and Numerical Solution*. *European Mathematical Society*, 2006.
- [2] Karel Devriendt, *Effective resistance is more than distance: Laplacians, Simplices and the Schur complement*. *Linear Algebra and its Applications*, 639, 24--49.