



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

NUMERICKÁ APROXIMACE VLASTNÍCH ČÍSEL MATICE

NUMERICAL APPROXIMATION OF MATRIX EIGENVALUES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ PRES

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. PETR TOMÁŠEK, Ph. D.

BRNO 2024

Zadání bakalářské práce

Ústav: Ústav matematiky
Student: **Jiří Pres**
Studijní program: Matematické inženýrství
Studijní obor: bez specializace
Vedoucí práce: **doc. Ing. Petr Tomášek, Ph.D.**
Akademický rok: 2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Numerická aproximace vlastních čísel matice

Stručná charakteristika problematiky úkolu:

Vlastní vektory a vlastní čísla matic hrají důležitou roli v oblasti řešení systémů diferenciálních a diferenčních rovnic, rovněž v analýze stability takovýchto systémů. K nalezení vlastních čísel pak lze využít několika více či méně efektivních přístupů.

Cíle bakalářské práce:

Rešerše numerických metod pro výpočet vlastních čísel a vlastních vektorů matic.
Komparace těchto metod.
Realizace vybraných algoritmů v Matlabu.

Seznam doporučené literatury:

GOLUB, G. H. a C. F. VAN LOAN. Matrix computations. 3rd ed. Baltimore: Johns Hopkins University Press, 1996, xxvii, 694 s. ISBN 0-8018-5414-8.

HOUSEHOLDER, A. S. Principles of numerical analysis. New York: McGraw-Hill Book Company, 1953.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

doc. Mgr. Petr Vašík, Ph.D.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá numerickými metodami pro aproximaci vlastních čísel matic, což je klíčový problém v mnoha oblastech vědy a techniky. Jsou zde uvedeny a analyzovány základní numerické algoritmy pro výpočet vlastních čísel a vlastních vektorů matic, jako jsou modifikace mocninné metody a QR metody. Všechny diskutované metody jsou realizovány ve formě skript v prostředí MATLAB a přiloženy k této práci. Testování na mnoha vybraných maticích vedlo i k závěrečnému přehledu zkoumaných metod vzhledem k jejich efektivnímu využití.

Summary

This work deals with numerical methods for the approximation of matrix eigenvalues, which is a key problem in many fields of science and technology. Main goal of the thesis is to get to know with several basic algorithms, such as power methods and the QR decomposition. Further to show these algorithms on examples and to compare achieved results. All described algorithms are processed in MATLAB scripts and attached in the appendix. After testing described methods, summary was created regarding their effective usage.

Klíčová slova

vlastní číslo, vlastní vektor, matice

Keywords

eigenvalue, eigenvector, matrix

Prohlašuji, že jsem bakalářskou práci *Numerická aproximace vlastních čísel matice* vypracoval samostatně pod vedením doc. Ing. Petra Tomáška, Ph. D. s použitím materiálů uvedených v seznamu literatury.

Jiří Pres

Rád bych vyjádřil vděčnost všem, kteří mi pomohli při psaní této práce. Zvláštní díky patří mému vedoucímu práce, panu doc. Ing. Petru Tomáškoví, Ph. D. za jeho cenné připomínky, trpělivost a motivaci po celou dobu psaní této práce. Děkuji také své rodině a přátelům za neustálou podporu a povzbuzení během celého procesu.

Jiří Pres

Obsah

1	Úvod	12
2	Matice a vlastní čísla	13
2.1	Matice	13
2.2	Problém vlastních čísel	14
2.2.1	Násobnost vlastních čísel	15
2.2.2	Lokalizace vlastních čísel	16
2.2.3	Závislost na transformacích	17
3	Metody výpočtu	18
3.1	Mocninná metoda	18
3.1.1	Normalizovaná mocninná metoda	19
3.1.2	Inverzní iterace	19
3.1.3	Inverzní iterace s posunem	19
3.1.4	Redukce matice	20
3.2	LR a QR transformace	20
3.2.1	Metoda LR rozkladu	20
3.2.2	Metoda QR rozkladu	21
3.3	Jacobiho metoda	23
3.4	Metoda bisekce	23
3.5	Lanczosova metoda	24
4	Příklady	26
4.1	Ukázka jednotlivých metod	26
4.2	Porovnání metod	29
5	Závěr	31
6	Seznam příloh	33

1 Úvod

Základy teorie matic byly položeny v 19. století. Matice, jako samostatný pojem, byl vytvořen v matematice, aby se usnadnilo zápis a manipulace s lineárními rovnicemi. Mezi první matematiky, kteří začali systematicky používat matice, patří Arthur Cayley a William Rowan Hamilton. Arthur Cayley je často považován za otce moderní teorie matic. V 19. století provedl rozsáhlý výzkum v oblasti lineární algebry a rozvinul teorii matic jako samostatnou matematickou disciplínu. V roce 1858 definoval matice a začal studovat jejich vlastnosti. Jeho práce¹ sehrála klíčovou roli v rozvoji teorie matic a položila základy pro další rozšíření a aplikace této matematické disciplíny. Teorie matic se stala klíčovým nástrojem při řešení lineárních rovnic, lineárních transformací a mnoha dalších problémů [6].

Nalezení vlastních čísel matic je klíčovým problémem v mnoha oblastech. Ve fyzice jsou základem pojmem kvantové mechaniky, kde reprezentují měřitelné vlastnosti fyzikálních systémů. Dále se využívají v analýze genetických dat, analýze finančních trhů a ekonomických modelů. V matematice určují stabilitu systémů diferenčních rovnic a tuhost počátečních problémů obyčejných diferenciálních rovnic. Celkově jsou vlastní čísla nedílnou součástí matematiky a poskytují klíčové nástroje pro analýzu a řešení široké škály problémů. Vzhledem k tomu, že analytické řešení je pro velké matice často neproveditelné, musíme k výpočtu použít numerických metod. Cílem této práce je představit několik numerických algoritmů pro aproximaci vlastních čísel matic.

Ve druhé kapitole uvedeme základní pojmy a tvrzení z teorie matic, které dále v práci využijeme. Nejprve si představíme výpočet pomocí mocninných metod pro získání právě jednoho vlastního čísla. Okomentujeme modifikace a jejich využití. Dále se zabýváme metodami LR a QR rozkladu, které můžeme použít pro široké spektrum matic. Nakonec popíšeme metody pro výpočet vlastních čísel symetrických matic a také jednu metodu pro nesymetrické matice. Každé metodě je věnován základní příklad pro ukázkou náročnosti, rychlosti konvergence a dosaženého výsledku. Dále jsou zde metody testovány na rozsáhlejších maticích.

¹Memoir on the Theory of Matrices (1858) - základ teorie matic

2 Matice a vlastní čísla

2.1 Matice

Definice 2.1. *Bud' X neprázdná množina skalárů a m, n přirozená čísla. Matice $A = (a_{ij})$ typu $m \times n$ nad množinou X je schéma složené z m krát n prvků množiny X zapsaných do tabulky o m řádcích a n sloupcích. Takovou matici zapisujeme ve tvaru*

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

nebo jen krátce $A = (a_{ij})$, kde a_{ij} je prvek na i -tém řádku a j -tém sloupci.

Poznámka 2.2. *Matice značíme velkým tučným písmenem např. A, B . Prvky matic mohou být i komplikované objekty, například algebraické výrazy nebo funkce. Množina skalárů bývá ale obvykle číselná. V dalším budeme předpokládat, že X je množina reálných čísel \mathbb{R} .*

Pro lepší orientaci jsou zavedeny a pojmenovány speciální případy matic, pojmenované podle tvaru, vlastností nebo toho, jaké obsahují prvky.

Definice 2.3. *Speciální případy matic*

- a) *Je-li $m = n$, matice se nazývá **čtvercová**, v obecném případě $m \neq n$ **obdélníková**.*
- b) *Matice, která má všechny prvky nulové se nazývá **nulová matice** 0 .*
- c) ***Jednotková matice** I je čtvercová matice s jedničkami na hlavní diagonále, ostatní prvky jsou nulové.*
- d) *Matice A se nazývá **symetrická**, jestli se každý prvek $a_{ij} = a_{ji}$, $\forall i, j$.*
- e) *Matice A se nazývá **horní trojúhelníková**, pokud pod hlavní diagonálou jsou samé nuly, tj. $a_{ij} = 0$ pro každé dva indexy $i > j$.*
- f) *Matice A se nazývá **dolní trojúhelníková**, pokud nad hlavní diagonálou jsou samé nuly, tj. $a_{ij} = 0$ pro každé dva indexy $i < j$.*
- g) *Transpozicí matice A nazveme takovou matici B , která splňuje vlastnost $(b_{ji}) = (a_{ij})$, $\forall i, j$. Transponovanou matici značíme A^T .*
- h) *Je-li determinant matice A nenulový, pak existuje její inverze A^{-1} , pro kterou platí $AA^{-1} = A^{-1}A = I$.*
- i) *Pokud je matice čtvercová, její transponovaná matice je současně maticí inverzní a jejíž řádky (respektive sloupce) tvoří soustavu ortogonálních vektorů, nazveme ji maticí **ortogonální**, tedy platí $Q^T Q = Q Q^T = I$.*
- j) *Pokud pro symetrickou reálnou matici $A \in \mathbb{R}^{n \times n}$ a každý nenulový reálný vektor $x \in \mathbb{R}^n$ platí:*
 - $x^T A x > 0$, *potom se A nazývá **pozitivně definitní**,*
 - $x^T A x \geq 0$, *potom se A nazývá **pozitivně semidefinitní**,*
 - $x^T A x < 0$, *potom se A nazývá **negativně definitní**,*
 - $x^T A x \leq 0$, *potom se A nazývá **negativně semidefinitní**.*

2.2 Problém vlastních čísel

Definice 2.4. *Vlastním číslem* λ matice A nazveme takové číslo, pro které existuje nenulový vektor x a platí následující rovnice:

$$Ax = \lambda x. \quad (1)$$

Tento vektor x nazveme **vlastním vektorem** příslušný vlastnímu číslu λ .

Důležitá vlastnost při hledání vlastního čísla je, že pokud ho odečteme od hlavní diagonály, bude determinant této matice s posunem λ roven nule.

Poznámka 2.5. Číslo λ je vlastním číslem matice A , pokud je řešením rovnice:

$$\det|A - \lambda I| = 0. \quad (2)$$

Definice determinantu je založena na permutacích prvků matic, neboli pořadí jejich sloupcových indexů. Pro určení znaménka zavádíme lichost (resp. sudost) permutace. Nastane-li v permutaci dvou čísel i a j k prohození oproti jejich vzestupnému pořadí, hovoříme o inverzi v permutaci. Pokud je počet inverzí lichý, bude mít permutace záporné znaménko a pokud je sudý, pak je permutace sudá a má kladné znaménko. Například permutace $p_1 = (1, 4, 2, 5, 3)$ je lichá, jelikož obsahuje inverze $(4, 2)$, $(4, 3)$ a $(5, 3)$, $\text{sgn}(p_1) = -1$. Naproti tomu $p_2 = (1, 2, 5, 3, 4)$ je permutací sudou a $\text{sgn}(p_2) = 1$.

Definice 2.6. *Determinant* čtvercové matice $A = (a_{ij})$ řádu n je roven součtu permutací prvků a_{ij}

$$\det|A| = \sum_{p \in P(n)} \text{sgn}(p) \cdot a_{1p_1} \cdot a_{2p_2} \cdot \dots \cdot a_{np_2}. \quad (3)$$

Pro $n = 2$ bude determinant matice A vypadat následovně:

$$\det|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = \sum_{p \in P(2)} \text{sgn}(p) a_{1p_1} a_{2p_2} = \text{sgn}(1, 2) a_{11} a_{22} + \text{sgn}(2, 1) a_{12} a_{21} = a_{11} a_{22} - a_{12} a_{21}.$$

Se zvětšujícím n se zásadním způsobem zvyšuje i počet možných permutací, $P(n) = n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1$. Pro $n = 3$ bude mít $\sum_{p \in P(3)}$ šest členů.

$$\begin{aligned} \det|A| &= \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \sum_{p \in P(3)} \text{sgn}(p) a_{1p_1} \cdot a_{2p_2} \cdot a_{3p_3} = \\ &= a_{11} a_{22} a_{33} + a_{12} a_{23} a_{31} + a_{13} a_{21} a_{32} - (a_{13} a_{22} a_{31} + a_{12} a_{21} a_{33} + a_{11} a_{23} a_{32}) \end{aligned}$$

Pro determinant z rovnice (2) budou diagonální prvky $a_{ii}^* = a_{ii} - \lambda$, pro $i = 1, 2, \dots, n$. Proto budou permutace prvků matice $(A - \lambda I)$ polynomy λ . Jelikož uvažujeme matici A nad množinou reálných čísel, bude mít polynom reálné koeficienty $c_n \in \mathbb{R}$.

$$p(\lambda) = c_n \lambda^n + c_{n-1} \lambda^{n-1} + \dots + c_1 \lambda + c_0 = 0.$$

Tento polynom plynoucí z rovnice (2) nazveme **charakteristickým** polynodem matice A . Aby bylo číslo λ vlastním číslem, musí být kořenem tohoto polynomu.

Rovnici (2) chápeme jako soustavu lineárních rovnic s nulovou pravou stranou. Jelikož předpokládáme nenulové x , musí být matice $(A - \lambda I)$ singulární. Při výpočtu soustavy získáme polynom stupně n , který odpovídá velikosti matice A , ten nazýváme charakteristický. Zde je důležité připomenout, že polynom s reálnými koeficienty může mít komplexní kořeny.

Věta 2.7. Základní věta algebry

Je-li $f(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_1 x + c_0$ polynom řádu n s reálnými koeficienty c_n , pak jej lze vyjádřit ve tvaru

$$f(x) = c_n (x - a_1)^{b_1} (x - a_2)^{b_2} \dots (x - a_k)^{b_k}, \quad (4)$$

kde $a_1, \dots, a_k \in \mathbb{C}$, $b_1, \dots, b_k \in \mathbb{N}$ a $b_1 + b_2 + \dots + b_k = n$. Pak a_i jsou kořeny polynomu $f(x)$ a mají násobnost b_i .

Determinant v rovnici (2) nahradíme charakteristickým polynomem

$$p(\lambda) = c_n \lambda^n + c_{n-1} \lambda^{n-1} + \dots + c_1 \lambda + c_0 = 0.$$

Z věty 2.7 víme, že matice A řádu n bude mít n vlastních čísel λ , která jsou obecně komplexní. Navíc platí, že pokud najdeme komplexní vlastní číslo $\lambda_1 = \alpha + \beta i$ s nenulovým β , pak bude vlastním číslem i číslo komplexně sdružené $\lambda_2 = \alpha - \beta i$.

2.2.1 Násobnost vlastních čísel

V obecném případě známe pouze počet kořenů charakteristického polynomu, ale nikoliv, zda se vlastní čísla opakují nebo ne. U vlastních čísel rozlišujeme dva typy násobností:

Algebraická násobnost odpovídá násobnosti kořene charakteristické rovnice. Pokud je algebraická násobnost daného čísla λ rovna jedné, říkáme, že vlastní číslo je *jednoduché*. Je-li větší než jedna, jedná se o *násobné* vlastní číslo.

Geometrická násobnost vlastního čísla je počet lineárně nezávislých vlastních vektorů, které můžeme dopočítat z rovnosti (1). Geometrická násobnost je vždy menší nebo rovna algebraické násobnosti.

Pomocí násobnosti vysvětlujeme další pojmy. Vlastní číslo nazveme *defektní*, jestliže je jeho geometrická násobnost menší než algebraická. Pokud jsou si rovny, hovoříme o *nedefektním vlastním čísle*.

Matici nazýváme defektní, pokud obsahuje alespoň jedno defektní vlastní číslo. V opačném případě se jedná o matici nedefektní. Defektní matice je užitečná při analýze vlastností grafu, jako je například určení počtu hran nebo sousedů každého uzlu, a může být využita v různých aplikacích, včetně síťového plánování, analýzy sociálních sítí, elektrických obvodů a dalších oblastí.

Pokud je matice A řádu n nedefektní, tj. všechny vlastní čísla jsou jednoduchá, má právě n lineárně nezávislých vlastních vektorů \mathbf{x}_i příslušných vlastním číslům λ_i , tj. existuje úplný systém vlastních vektorů $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$. Označme matici $\mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, pak $\mathbf{AX} = \mathbf{XD}$ a $\mathbf{A} = \mathbf{XD}\mathbf{X}^{-1}$, tzv. spektrální rozklad matice A . Z tohoto vztahu vidíme, že matice A je podobná matici \mathbf{D} , tedy pomocí obnásobení maticí vlastních vektorů můžeme matici A převést na diagonální tvar. Tato vlastnost se nazývá *diagonalizovatelnost* a je předpokladem pro některé metody pro výpočet vlastních čísel. Vidíme tedy, že pokud je matice A nedefektní, pak je také diagonalizovatelná.

2.2.2 Lokalizace vlastních čísel

Ve speciálních případech víme o vlastních číslech mnohé. Pokud je matice symetrická, potom jsou její vlastní čísla reálná a navíc odpovídají definitnosti této matice [4]. V případě obecné matice nám s lokalizací vlastních čísel pomůže následující věta.

Věta 2.8. Geršgorinova věta

Vlastní čísla matice $\mathbf{A} = (a_{ij})$, kde $i, j = 1, 2, \dots, n$ jsou obsažena ve sjednocení kruhů v komplexní rovině se středem v bodě a_{ii} a poloměrem $\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$.

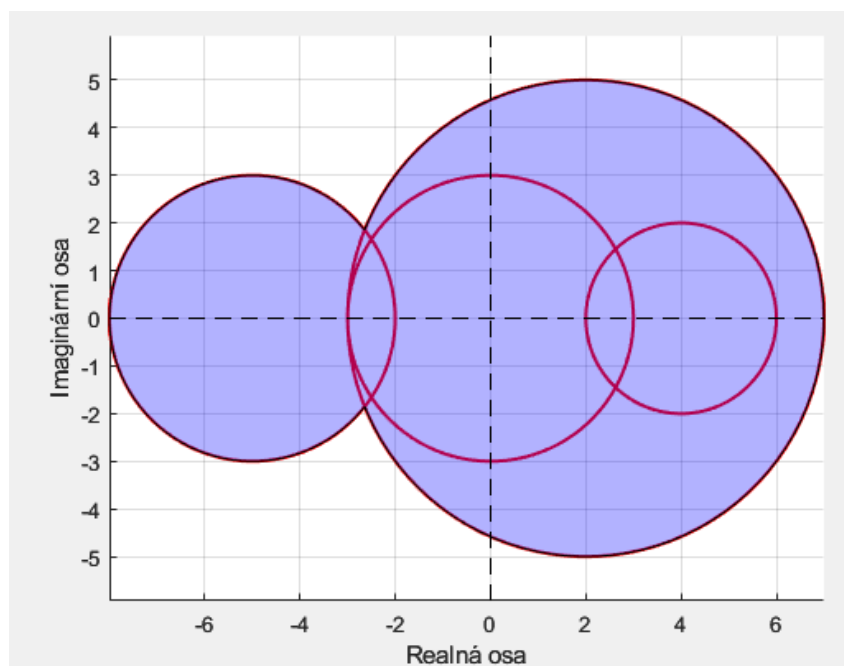
Ukažme si vizualizaci této věty na jednoduchém příkladě.

Příklad 2.9. Uvažujme matici

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 2 & 0 \\ -1 & 4 & 0 & 1 \\ 4 & 0 & 2 & 1 \\ -1 & -2 & 0 & -5 \end{pmatrix}$$

Podle Geršgorinovy věty budou vlastní čísla matice \mathbf{A} ležet v komplexní rovině ve sjednocení:

- Kruhu se středem $S_1 = [0, 0]$ a poloměrem $r_1 = |1| + |2| + |0| = 3$
- Kruhu se středem $S_2 = [4, 0]$ a poloměrem $r_2 = |-1| + |0| + |1| = 2$
- Kruhu se středem $S_3 = [2, 0]$ a poloměrem $r_3 = |4| + |0| + |1| = 5$
- Kruhu se středem $S_4 = [-5, 0]$ a poloměrem $r_4 = |-1| + |-2| + |0| = 3$



Obr. 1: Vizualizace Geršgorinovy věty na matici \mathbf{A}

Vlastní čísla tedy budou ležet v modré oblasti komplexní roviny na obrázku.

2.2.3 Závislost na transformacích

Uvažujme dvě matice:

- Matice A s vlastními čísly λ a vlastními vektory x
- Matice B s vlastními čísly μ a vlastními vektory y

Následující tabulka uvádí přehled základních transformací matice A a jejich dopad na vlastní čísla a vektory.

Transformace	Předpoklady	Vlastní čísla	Vlastní vektory
$B = A - cI$	$c \in \mathbb{C}$	$\mu = \lambda - c$	$y = x$
$B = A^{-1}$	$\det A \neq 0$	$\mu = \frac{1}{\lambda}$	$y = x$
$B = A^n$	$n \in \mathbb{N}$	$\mu = \lambda^n$	$y = x$
$B = p(A)$	$p(A) = a_n A^n + a_{n-1} A^{n-1} + \dots + a_1 A + a_0$	$\mu = p(\lambda)$	$y = x$
$B = T^{-1}AT$	$\det T \neq 0$	$\mu = \lambda$	$y = T^{-1}x$

V další kapitole budeme k nalezení vlastních čísel využívat posunu, inverze a hlavně podobnostní transformace. Ukažme si tedy, že tabulka odpovídá realitě.

Posun znamená odečtení konstanty od diagonálních prvků, tj. $B = (A - c \cdot I)$. Jestliže $Ax = \lambda x$ a c je komplexní číslo, pak $(A - cI)x = (\lambda - c)x$. Vlastní čísla matice B se oproti vlastním číslům matice A posunou o c , vlastní vektory se však nezmění.

Inverze. Je-li A regulární, tj. $\det|A| \neq 0$ a ani žádné $\lambda \neq 0$, a $Ax = \lambda x$, pak $A^{-1}x = \frac{1}{\lambda}x$. Tedy vlastní čísla matice A^{-1} jsou převrácenými hodnotami vlastních čísel původní matice a vlastní vektory zůstávají stejné.

Podobnost. Matice A a B jsou si podobné, pokud existuje regulární matice T a je splněna rovnost $B = T^{-1}AT$. Matici T nazýváme podobnostní maticí. Platí-li $By = \lambda y$, pak platí také $T^{-1}ATy = \lambda y$. Po vynásobení podobnostní maticí zleva můžeme psát $ATy = \lambda Ty$. Pro substituci $x = Ty$ tedy platí $Ax = \lambda x$. Vlastní čísla podobných matic jsou stejná a vlastní vektory můžeme dopočítat pomocí podobnostní matice.

3 Metody výpočtu

Již víme, že vlastní čísla jsou kořeny charakteristické rovnice matice A . Pro matici řádu n platí, že charakteristická rovnice bude řádu n . Řešení lineárního a kvadratického polynomu je všem známo. Existují i tzv. Cardanovy vzorce, které umožňují analyticky určit kořeny polynomu 3. a 4. stupně. Pro vyšší n můžeme pomocí *Hornerova schématu* pouze hádat řešení. Uvedme si tedy několik numerických metod pro výpočet vlastních čísel.

3.1 Mocinná metoda

Jedná se o jednu z nejjednodušších iteračních metod. Základním předpokladem pro mocninou metodu je, aby matice A řádu n měla také n lineárně nezávislých vlastních vektorů, neboli aby matice A byla nedefektní. Za tohoto předpokladu můžeme napsat libovolný vektor \mathbf{v} jako lineární kombinaci

$$\mathbf{v} = \sum_{i=1}^n \alpha_i \mathbf{x}_i, \quad (5)$$

kde \mathbf{x}_i jsou vlastní vektory matice A . Pokud uvažujeme λ_i vlastní číslo odpovídající vlastnímu vektoru \mathbf{x}_i , platí

$$A\mathbf{v} = \sum_{i=1}^n \alpha_i \lambda_i \mathbf{x}_i. \quad (6)$$

Dalším důležitý předpokladem pro konvergenci mocninné metody je, aby matice A měla jediné dominantní vlastní číslo λ_1 , tj. $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$.

Algoritmus 3.1. Mocinná metoda

- (1) zvolíme počáteční nenulovou aproximaci \mathbf{v}_0
- (2) **for** $k = 1, 2, \dots$ **do**
- (3) $\mathbf{v}_k = A\mathbf{v}_{k-1}$
- (4) **end**

Pokud platí předpoklady na diagonalizovatelnost matice A a platí, že má jediné dominantní vlastní číslo, můžeme ukázat, že \mathbf{v}_k konverguje k nějakému násobku vlastního vektoru \mathbf{x}_1 odpovídající v absolutní hodnotě největšímu vlastnímu číslu λ_1 . Vektor \mathbf{v} si napíšeme podle (5) jako lineární kombinaci \mathbf{x}_i , pak

$$\begin{aligned} \mathbf{v}_k &= A\mathbf{v}_{k-1} = A^2\mathbf{v}_{k-2} = \dots = A^k\mathbf{v}_0 = A^k\mathbf{v} = A^k \sum_{i=1}^n \alpha_i \mathbf{x}_i = \\ &= \sum_{i=1}^n \alpha_i A^k \mathbf{x}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{x}_i = \lambda_1^k \left(\alpha_1 \mathbf{x}_1 + \sum_{i=2}^n \left(\frac{\lambda_i}{\lambda_1} \right)^k \alpha_i \mathbf{x}_i \right). \end{aligned}$$

Pro $i > 1$ je $\left| \frac{\lambda_i}{\lambda_1} \right| < 1$, a proto $\left| \frac{\lambda_i}{\lambda_1} \right|^k \rightarrow 0$. Na pravé straně zůstane pouze první člen a vidíme, že $\mathbf{v}_k = \lambda_1^k \alpha_1 \mathbf{x}_1$. Tedy \mathbf{v}_k je násobkem vlastního vektoru \mathbf{x}_1 . Vlastní číslo λ_1 můžeme následně spočítat pomocí *Rayleighova podílu*. Rychlost konvergence závisí na počáteční volbě \mathbf{v}_0 .

Pokud je rozdíl mezi prvním a druhým největším vlastním číslem v absolutní hodnotě relativně malý, může mocninná metoda konvergovat velmi pomalu.

Rayleighův podíl pro matici \mathbf{A} a nenulový vektor \mathbf{y} je číslo $R(\mathbf{A}, \mathbf{y}) = \frac{\mathbf{y}^T \mathbf{A} \mathbf{y}}{\mathbf{y}^T \mathbf{y}}$.

Je-li \mathbf{y} vlastní vektor matice \mathbf{A} a λ je odpovídající vlastní číslo, pak

$$R(\mathbf{A}, \mathbf{y}) = \frac{\mathbf{y}^T \mathbf{A} \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{\mathbf{y}^T \lambda \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \lambda. \text{ Víme, že } \mathbf{v}_k \rightarrow \mathbf{x}_1, \text{ a tedy } R(\mathbf{A}, \mathbf{v}_k) \rightarrow \lambda_1.$$

3.1.1 Normalizovaná mocninná metoda

Při výpočtu na počítači může při postupném násobení v algoritmu 3.1 dojít k přetečení ($\lambda_1 > 1$) nebo podtečení ($\lambda_1 < 1$) paměti. Abychom se tomuto problému vyhnuli, můžeme vektor \mathbf{v}_k normovat. Navíc se při následném výpočtu Rayleighova podílu bude jmenovatel rovnat jedné.

Algoritmus 3.2. *Normalizovaná mocninná metoda*

- (1) zvolíme počáteční nenulovou aproximaci \mathbf{v}_0 , $\|\mathbf{v}_0\| = 1$
- (2) **for** $k = 1, 2, \dots$ **do**
- (3) $\mathbf{y}_k = \mathbf{A} \mathbf{v}_{k-1}$
- (4) $\mathbf{v}_k = \frac{\mathbf{y}_k}{\|\mathbf{y}_k\|}$
- (5) $\sigma_k = \mathbf{v}_k^T \mathbf{A} \mathbf{v}_k$
- (6) **end**

Výpočet zastavíme pomocí vhodného stop-kritéria, např. $\|\mathbf{v}_k - \mathbf{v}_{k-1}\| < \epsilon$. Z předchozího textu víme, že $\mathbf{v}_k \rightarrow \mathbf{x}_1$, $\|\mathbf{x}_1\| = 1$ a $\sigma_k = R(\mathbf{A}, \mathbf{v}_k) \rightarrow \lambda_1$.

3.1.2 Inverzní iterace

Poté, co jsme našli největší vlastní číslo matice \mathbf{A} , můžeme využít toho, že vlastní čísla matice \mathbf{A}^{-1} jsou převrácenými hodnotami matice původní. Takže mocninná metoda aplikovaná na matici \mathbf{A}^{-1} konverguje k vektoru příslušnému nejmenšímu vlastnímu číslu matice \mathbf{A} v absolutní hodnotě. Pro ulehčení místo řešení $\mathbf{y}_k = \mathbf{A}^{-1} \mathbf{v}_{k-1}$ využijeme *LU rozkladu* $\mathbf{A} = \mathbf{L} \mathbf{U}$ a řešíme soustavu rovnic $\mathbf{L} \mathbf{U} \mathbf{y}_k = \mathbf{v}_{k-1}$.

Algoritmus 3.3. *Inverzní normalizovaná mocninná metoda*

- (1) zvolíme počáteční nenulovou aproximaci \mathbf{v}_0 , $\|\mathbf{v}_0\| = 1$
- (2) vypočti *LU rozklad* matice \mathbf{A}
- (3) **for** $k = 1, 2, \dots$ **do**
- (4) vypočti \mathbf{y}_k řešením $\mathbf{L} \mathbf{U} \mathbf{y}_k = \mathbf{v}_{k-1}$
- (5) $\mathbf{v}_k = \frac{\mathbf{y}_k}{\|\mathbf{y}_k\|}$
- (6) $\sigma_k = \mathbf{v}_k^T \mathbf{A} \mathbf{v}_k$
- (7) **end**

Takže $\mathbf{v}_k \rightarrow \mathbf{x}_1$, $\|\mathbf{x}_1\| = 1$ a $\sigma_k \rightarrow \lambda_1$, kde $|\lambda_1| < |\lambda_j|$ pro $j = 2, 3, \dots, n$.

3.1.3 Inverzní iterace s posunem

Uvažujme situaci, že známe μ dostatečně dobrý odhad některého vlastního čísla λ_i . Stejně jako v algoritmu 3.3 využijeme *LU rozkladu*, tentokrát ale pro matici \mathbf{A} s posunem μ , tj. $(\mathbf{A} - \mu \mathbf{I})$.

Algoritmus 3.4. Inverzní normalizovaná mocninná metoda s posunem

- (1) zvolíme počáteční nenulovou aproximaci \mathbf{v}_0 , $\|\mathbf{v}_0\| = 1$
- (2) vypočti LU rozklad matice $(\mathbf{A} - \mu\mathbf{I})$
- (3) **for** $k = 1, 2, \dots$ **do**
- (4) vypočti \mathbf{y}_k řešením $\text{LU}\mathbf{y}_k = \mathbf{v}_{k-1}$
- (5) $\mathbf{v}_k = \frac{\mathbf{y}_k}{\|\mathbf{y}_k\|}$
- (6) $\sigma_k = \mathbf{v}_k^T \mathbf{A} \mathbf{v}_k$
- (7) **end**

Zde $\mathbf{v}_k \rightarrow \mathbf{x}_i$, příslušnému vlastnímu vektoru λ_i , kde $\|\mathbf{x}_1\| = 1$, a $\sigma_k \rightarrow \lambda_i$.

3.1.4 Redukce matice

Nyní si ukážeme, jak lze pomocí metody, která nám iteračně vypočítá pouze jedno vlastní číslo matice \mathbf{A} získat postupně všechna vlastní čísla. Pomocí normalizované mocninné metody jsme si vypočetli dominantní vlastní číslo λ_1 a odpovídající vlastní vektor \mathbf{x}_1 . Pro ulehčení budeme uvažovat, že matice \mathbf{A} je symetrická a má tedy pouze reálná vlastní čísla. Definujme matici $\mathbf{A}_1 = \mathbf{A} - \frac{\lambda_1 \mathbf{x}_1 \mathbf{x}_1^T}{\|\mathbf{x}_1\|^2}$. Díky vlastnosti posunu víme, že má vlastní čísla $0, \lambda_2, \lambda_3, \dots, \lambda_n$. Jestliže je číslo λ_2 dominantní vlastní číslo matice \mathbf{A}_1 , můžeme aplikací mocninné metody na matici \mathbf{A}_1 určit aproximaci σ_2 vlastního čísla λ_2 a pomocí inverzní iterace s posunem $\mu = \sigma_2$ vypočteme λ_2 . Tento postup můžeme opakovat a vypočítat několik dalším vlastních čísel λ_j matice \mathbf{A} a k nim odpovídající vlastní vektory \mathbf{x}_j . Je určitě dobře představitelné, že tento postup je zdlouhavý. Proto pro výpočet všech vlastních čísel používáme jiné metody, které si představíme v dalších kapitolách.

3.2 LR a QR transformace

Obě metody, které zde budeme popisovat, lze využít k výpočtu vlastních čísel libovolné matice. Převedením na Hessengergův tvar nebo na třídiagonální matici můžeme výpočet ještě zkrátit [1]. Cílem transformací je získat nějakou podobnou matici, u které můžeme vlastní čísla zjistit jednodušeji (například u trojúhelníkových rovnu získat z diagonály) než u původní matice. Výhodou oproti mocninným metodám je, že získáme všechny vlastní čísla.

3.2.1 Metoda LR rozkladu

První metoda, LR transformace, je snadněji proveditelná, ale nemusí být vždy numericky stabilní. Základem této metody je vytvoření posloupnosti matic $\{\mathbf{A}_k\}$ a následné rozložení na součin dolní trojúhelníkové matice \mathbf{L} a horní trojúhelníkové matice \mathbf{R} . Je-li matice \mathbf{A}_1 rozložena na součin $\mathbf{L}_1 \mathbf{R}_1$, pak má matice $\mathbf{A}_2 = \mathbf{R}_1 \mathbf{L}_1$ stejná vlastní čísla jako matice \mathbf{A}_1 . To plyne z rovnice

$$\mathbf{A}_2 = \mathbf{R}_1 \mathbf{L}_1 = \mathbf{L}_1^{-1} \mathbf{A}_1 \mathbf{L}_1 = \mathbf{R}_1 \mathbf{A}_1 \mathbf{R}_1^{-1}, \quad (7)$$

kteřá ukazuje, že \mathbf{A}_1 a \mathbf{A}_2 jsou podobné. Nyní můžeme znovu použít rozklad, tentokrát na matici \mathbf{A}_2 . Proces postupně opakujeme podle následujícího algoritmu.

Algoritmus 3.5. LR metoda

- (1) $A_1 := A$
- (2) **for** $k = 1, 2, \dots$ **do**
- (3) *proved' LR rozklad* $L_k R_k = A_k$
- (4) $A_{k+1} = R_k L_k$
- (5) **end**

Všechny matice A_k jsou si podobné, to plyne z (7). Položme $P_k = R_k R_{k-1} \dots R_1$ a $N_k = L_1 L_2 \dots L_k$. Pak P_k , resp. L_k je horní, resp. dolní trojúhelníková matice. Jsou-li všechny matice regulární, platí podobně jako v (7) rovnice

$$A_{k+1} = L_k^{-1} A_k L_k = R_k A_k R_k^{-1}, \quad (8)$$

odtud matematickou indukcí plyne, že

$$A_{k+1} = N_k^{-1} A_k N_k = P_k A_k P_k^{-1}. \quad (9)$$

Protože platí, $L_k R_k = R_{k-1} L_{k-1}$, můžeme ukázat

$$N_k P_k = L_1 \dots L_{k-1} L_k R_k R_{k-1} \dots R_1 = L_1 \dots L_{k-1} R_{k-1} L_{k-1} R_{k-1} \dots R_1, \quad (10)$$

opakujeme-li tento postup, dostaneme, že platí

$$N_k P_k = (L_1 R_1)^k = A_k. \quad (11)$$

Získali jsme konstrukci posloupnosti matic $\{A_k\}$. V dalším budeme předpokládat, že matice R_k má jedničky na hlavní diagonále. Lze totéž předpokládat pro matici L_k místo R_k , ale i tak dospějeme ke stejnému výsledku. Budeme se zabývat konvergencí posloupnosti $\{A_k\}$. Konvergence této řady závisí na konvergenci posloupnosti $\{P_k\}$ a platí následující věta.

Věta 3.6. *Konverguje-li posloupnost P_k pro $k \rightarrow \infty$ k regulární matici P_∞ , pak existuje také $\lim_{k \rightarrow \infty} A_k$ a je to dolní trojúhelníková matice.*

Důkaz konvergence $\{P_k\}$ není obsažen v použité literatuře, protože přesahuje rámec těchto knih. Můžeme říci, že konverguje pro velmi širokou třídu matic, která obsahuje všechny pozitivně definitní matice, mnohé matice s různými i mnohonásobnými vlastními čísly a mnohé matice s komplexními vlastními čísly [?]. Pokud má matice A reálná vlastní čísla, budou obsažena na diagonále L_∞ seřazena sestupně v absolutní hodnotě. Abychom se nemuseli zabývat konvergencí, častěji volíme metodu *QR rozkladu* [2].

3.2.2 Metoda QR rozkladu

Jedním z nejpoužívanějších postupů pro jednorázový výpočet všech vlastních čísel matice je *QR metoda*. Z názvu je jasné, že hlavním bodem metody je **QR rozklad** matice A . Matici A v každém kroku algoritmu rozkládáme na součin matice R , která je horní trojúhelníková, a ortogonální matice Q . Díky volbě tohoto rozkladu si nemusíme dělat starosti s konvergencí jako u LR metody. Dále platí, že QR rozklad matice není jednoznačný, ale existuje vždy [3].

Algoritmus 3.7. QR metoda

- (1) $A_0 := A$
- (2) **for** $k = 1, 2, \dots$ **do**
- (3) *proved' QR rozklad* $Q_k R_k = A_{k-1}$
- (4) $A_k = R_k Q_k$
- (5) **end**

Vezměme si matici A_k a z třetího řádku 3.7 vyjádříme R_k , poté můžeme ukázat, že

$$A_k = R_k Q_k = Q_k^T A_{k-1} Q_k = Q_k^T Q_{k-1}^T A_{k-2} Q_{k-1} Q_k = \dots = \hat{Q}_k^T A \hat{Q}_k,$$

kde $\hat{Q}_k = Q_1 Q_2 \dots Q_k$ je ortonormální matice a platí následující věta.

Věta 3.8. *Věta o QR algoritmu* Jestliže $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$, pak A_k konvergují k horní trojúhelníkové matici $T = Q^T A Q$, kde $Q = \lim_{k \rightarrow \infty} \hat{Q}_k$.

Z věty 3.8 víme, že metoda konverguje k horní trojúhelníkové matici, proto můžeme jako *stop-kritérium* nastavit, aby byly poddiagonální prvky v absolutní hodnotě menší než nějaká pevně zvolená tolerance ϵ . Po ukončení výpočtu známe všechna vlastní čísla matice A , protože se nám zobrazí na hlavní diagonále matice A_k . Pokud je navíc původní matice A symetrická, dostaneme vlastní vektory jako sloupce matice $Q = (x_1, x_2, \dots, x_n)$. Pokud je matice A obecně nesymetrická, k výpočtu vlastních vektorů můžeme použít například inverzní mocninnou metodu s posunem z algoritmu 3.4.

Musíme brát v potaz, že ve výpočtu pomocí QR rozkladu v každém kroku počítáme rozklad matice A_{k-1} na součin jiných dvou matic Q_k a R_k . Je asi jasné, že počet iterací bude závislý na velikosti matice, se kterou počítáme. Rozklad pro plně zaplněnou matici vyžaduje $O(n^3)$ operací. Je tedy běžné nejprve převádět matice na Hessenbergův tvar. Pomocí této transformace snížíme počet operací na $O(n^2)$. Pokud bychom dostali matici A_{k-1} do třídiagonálního tvaru, pak by se náročnost snížila dokonce na $O(n)$ iterací. Obě transformace jsou založeny na podobnosti matic a tudíž se při nich hodnoty vlastních čísel nemění.

Pro zrychlení konvergence QR metody využíváme Hessenbergovu transformaci, v MATLABu příkazem *hess* a vhodně zvolených posunů.

Algoritmus 3.9. QR metoda s posuny

- (1) *převědeme matici A na Hessenbergův tvar*
- (2) $A_0 := A$
- (3) **for** $k = 1, 2, \dots$ **do**
- (4) *vyber posun* μ_k
- (5) *proved' QR rozklad* $Q_k R_k = A_{k-1} - \mu_k I$
- (6) $A_k = R_k Q_k + \mu_k I$
- (7) **end**

Nejjednodušší volba pro posun μ_k je prvek $a_{nn}^{(k-1)}$ matice A_{k-1} , tzv. Wilkinsonův posun[5]. Pokud jsou všechny prvky v posledním řádku kromě $a_{nn}^{(k-1)}$ matice A_k dostatečně malé, považujeme $a_{nn}^{(k-1)}$ za dobrou aproximaci vlastního čísla. Zbývající vlastní čísla hledáme v submatici tvořené prvními $(n-1)$ řádky a sloupci. Opakováním toho algoritmu dostaneme všechna vlastní čísla.

3.3 Jacobiho metoda

Z předchozího textu víme, že pokud je matice A nedefektní, potom existuje její spektrální rozklad $Q^T A Q = D$, kde Q je ortogonální matice a D je diagonální s vlastními čísly matice A . Metoda spočívá v sestavení posloupnosti $\{S_k\}$ ortogonálních matic, pro kterou platí, že $\lim_{k \rightarrow \infty} S_1 \cdot S_2 \cdot \dots \cdot S_k = Q$.

Algoritmus 3.10. *Jacobiho metoda*

- (1) $A_0 := A$
- (2) *for* $k = 1, 2, \dots$ *do*
- (3) $A_{k+1} = J_{k+1}^T A_k J_{k+1}$
- (4) *end*,

kde matice J_{k+1} je ortonormální Jacobiho matice rovinné rotace zvolená tak, aby se vynuloval jeden pár symetricky položených prvků $a_{p_k q_k} = a_{q_k p_k}$ matice A_k . Matici $J_{k+1} := J(p_k, q_k, \theta_k)$ definujeme jako jednotkovou, ale prvky $j_{p_k p_k}$ a $j_{q_k q_k}$ jsou rovny $\cos \theta_k$ a prvky $j_{p_k q_k}$, $j_{q_k p_k}$ nejsou rovny nule, ale rovné $\pm \sin \theta_k$. Označme prvky matice J na pozicích (p, p) , (p, q) , (q, p) a (q, q) postupně čísla $c, s, -s$ a c . Dále prvky matice A_k označme $a = a_{pp}$, $b = a_{pq} = a_{qp}$ a $d = a_{qq}$. Koeficienty c a s určíme tak, aby matice

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} a & b \\ b & d \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} c^2 a - 2csb + s^2 d & c^2 b + cs(a-d) - s^2 b \\ c^2 b + cs(a-d) - s^2 b & c^2 d + 2csb + s^2 a \end{pmatrix}$$

byla diagonální, tj. $c^2 b + cs(a-d) - s^2 b = 0$. Dostaneme $t^2 + 2\tau t - 1 = 0$, kde $\tau = \frac{d-a}{2b}$ a $t = \frac{s}{c} = \tan \theta$, takže $c = \frac{1}{\sqrt{1+t^2}}$ a $s = ct$.

Za t volíme menší kořen kvadratické rovnice v absolutní hodnotě $t = \frac{\text{sgn}(\tau)}{|\tau| + \sqrt{1+\tau^2}}$.

Protože matice A_k je rovna součinu $A_k = \hat{J}_k^T A \hat{J}_k$, kde $\hat{J}_k = J_1 \cdot J_2 \cdot \dots \cdot J_k$, jsou si matice A_k a A podobné, tedy mají stejná vlastní čísla. Výpočtem v algoritmu 3.10 postupně nulujeme mimodiagonální prvky A_k a čísla na diagonále jsou dostatečně dobrými aproximacemi vlastních čísel. Jelikož musí být matice A pro tuto metodu symetrická, podobně jako u QR metody, i -tý sloupec matice \hat{J}_k je pak aproximací vlastního vektoru odpovídající aproximaci $a_{ii}^{(k)}$ vlastního čísla λ_i .

3.4 Metoda bisekce

Další z metod pro výpočet vlastních čísel symetrických matice je metoda bisekce. Nejprve si matici převedeme na třídiagonální tvar, tedy

$$A = \begin{pmatrix} a_1 & b_2 & & & \\ b_2 & a_2 & b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & b_n \\ & & & b_n & a_n \end{pmatrix},$$

a definujeme polynomy $p_0(x), p_1(x), \dots, p_n(x)$ předpisem:

$$p_0(x) = 1, p_r(x) = \det(A_r - xI) \text{ pro } r = 1, 2, \dots, n,$$

kde A_r je submatice tvořená prvními r řádky a sloupci matice A .

Je jasné, že takto vytvořené polynomy jsou charakteristickými polynomy. Postupným půlením intervalu můžeme hledat kořeny těchto polynomů. Je-li $a < b$ a $p_n(a) \cdot p_n(b) < 0$, pak v intervalu (a, b) leží vlastní číslo, které pro danou přesnost ϵ určíme pomocí algoritmu 3.11.

Algoritmus 3.11. *Metoda půlení intervalu*

- (1) **while** $b - a > \epsilon$ **do**
- (2) $x := \frac{a+b}{2}$
- (3) **if** $p_n(a) \cdot p_n(b) < 0$
- (4) $b := x$
- (5) **else**
- (6) $a := x$

I když je algoritmus výpočtu jednoduchý, musíme dodržet, aby ve zvoleném intervalu (a, b) leželo pouze jedno vlastní číslo, pouze jeden kořen charakteristického polynomu.

3.5 Lanczosova metoda

Jednou z metod pro výpočet vlastních čísel nesymetrické matice je Lanczosova metoda. Přístup k řešení v případě nesymetrických matic je analogický s dalšími metodami popsanými dříve. Smyslem je provést řadu podobnostních transformací matice A na jinou matici, která má stejná vlastní čísla. Budeme se snažit sestavit matici S takovou, aby výsledkem $T = S^{-1}AS$ byla třídiagonální matice T . Zkonstruujeme posloupnost vektorů $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ takových, že vezmeme-li je za sloupce matice S , budou matice A a T podobné. Dále budeme potřebovat posloupnost vektorů $\mathbf{y}_1, \dots, \mathbf{y}_n$ takovou, že $\{\mathbf{x}_i\}$ a $\{\mathbf{y}_i\}$ budou *biortogonální*, tj. platí $\mathbf{y}_j^T \mathbf{x}_i = 0$, pro každé $i \neq j$. Tyto dvě posloupnosti jsou dány rekurentním zápisem

$$\mathbf{x}_{k+1} = A\mathbf{x}_k - b_k\mathbf{x}_k - c_{k-1}\mathbf{x}_{k-1}, \quad (12)$$

$$\mathbf{y}_{k+1} = A^T\mathbf{y}_k - b_k\mathbf{y}_k - c_{k-1}\mathbf{y}_{k-1},$$

kde $k = 1, \dots, n-1$, $\mathbf{x}_0 = \mathbf{y}_0 = \mathbf{0}$ a \mathbf{x}_1 a \mathbf{y}_1 jsou nenulové libovolně zvolené vektory, pak čísla b_k, c_k jsou definovány takto:

$$b_k = \frac{\mathbf{y}_k^T A \mathbf{x}_k}{\mathbf{y}_k^T \mathbf{x}_k}, \quad (13)$$

$$c_{k-1} = \frac{\mathbf{y}_{k-1}^T A \mathbf{x}_k}{\mathbf{y}_{k-1}^T \mathbf{x}_{k-1}},$$

pro $k = 1, \dots, n-1$ a $c_0 := 0$.

Věta 3.12. *Za předpokladu $\mathbf{y}_j^T \mathbf{x}_j \neq 0$ pro $j = 1, \dots, n$, platí:*

1. Vektory \mathbf{x}_i pro $i = 1, \dots, n$ jsou lineárně nezávislé.
2. $\mathbf{x}_{n+1} = \mathbf{0}$

Důkaz plyne z biortogonality a druhý bod lze ukázat přímo z výpočtu (12). Ujijeme-li této větě, můžeme z první rovnice (12) psát:

$$Ax_1 = x_2 + b_1x_1, \quad (14)$$

$$Ax_k = x_{k-1} + b_kx_k + c_{k-1}x_{k-1},$$

pro $k = 2, \dots, n-1$, a také

$$Ax_n = b_nx_n + c_{n-1}x_{n-1}.$$

Je-li tedy S regulární matice, jejíž sloupce jsou vektory x_1, \dots, x_n , plyne z (14), že platí

$$AS = S \begin{pmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 \\ 1 & b_2 & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & \dots & 0 & 1 & b_n \end{pmatrix} = ST. \quad (15)$$

Matice T ze vzorce (15) je skutečně požadovanou třídiagonální maticí. Stačí pouze určit koeficienty b_k a c_k definované rekurencemi (13). Tyto koeficienty jsou závislé na libovolně zvolených vektorech x_1 a y_1 .

Charakteristický polynom matice T se dá vypočítat analogicky jako u metody bisekce. Zde máme:

$$f_{n-(i+1)}(\lambda) = (\lambda - b_{i+1})f_{n-i}(\lambda) - c_i f_{n-(i-1)}, \text{ pro } i = 1, \dots, n-1, \quad (16)$$

kde $f_n(\lambda) = 1$ a $f_{n-1}(\lambda) = -b_1 + \lambda$. Charakteristický polynom je pak $f_0(\lambda)$.

4 Příklady

4.1 Ukázka jednotlivých metod

Nejprve si na jednoduchých příkladech s matice malého řádu ukažme, co je výsledkem jednotlivých metod. S ukázkami začneme dle pořadí představených metod v předchozím textu. Za prvé ukážeme, jak si mocninné metody poradí s maticí řádu 3, která splňuje podmínku diagonalizovatelnosti a má jediné dominantní vlastní číslo.

Příklad 4.1. *Užití mocninných metod*

Uvažujme matici $A = \begin{pmatrix} 1 & 1 & 0.5 \\ 1 & 1 & 0.25 \\ 0.5 & 0.25 & 2 \end{pmatrix}$.

Pomocí normalizované mocninné metody (algoritmus 3.2) vypočítáme největší vlastní číslo v absolutní hodnotě λ_1 a k němu příslušný vlastní vektor \mathbf{x}_1 . Zvolme počáteční odhad $\mathbf{v}_0 = (1, 1, 1)$ a požadovanou přesnost $\epsilon = 10^{-5}$.

index k-té iterace	vypočtený vlastní vektor	vypočtené vlastní číslo
1	(0.5754, 0.5179, 0.6330)	2.5248
2	(0.5578, 0.4952, 0.6660)	2.5325
3	(0.5471, 0.4814, 0.6848)	2.5352
4	(0.5407, 0.4732, 0.6955)	2.5361
⋮	⋮	⋮
18	(0.5315, 0.4615, 0.7103)	2.5365

Stejně tak můžeme pomocí inverzní normalizované mocninné metody (algoritmus 3.3) vypočítat nejmenší vlastní číslo v absolutní hodnotě.

index k-té iterace	vypočtený vlastní vektor	vypočtené vlastní číslo
1	(0.8321, -0.5547, 0)	0.0769
2	(-0.7202, 0.6893, 0.0942)	-0.0166
⋮	⋮	⋮
5	(0.7212, -0.6863, 0.0937)	-0.0166

Poslední vlastní číslo odhadneme hodnotou $\mu = 1.4$ a dosadíme do inverzní normalizované mocninné metody s posunem μ (algoritmus 3.4).

index k-té iterace	vypočtený vlastní vektor	vypočtené vlastní číslo
1	(0.6139, 0.6842, -0.3937)	1.6187
2	(0.4582, 0.5750, -0.6778)	1.4809
⋮	⋮	⋮
6	(0.4443, 0.5621, -0.6976)	1.4801

Z těchto tří výpočtů jsme získali všechna vlastní čísla a k nim příslušné vlastní vektory:

$$\begin{aligned}\lambda_1 &= 2.5365 \text{ a } \mathbf{x}_1 = (0.5315, 0.4615, 0.7103) \\ \lambda_2 &= 1.4801 \text{ a } \mathbf{x}_2 = (0.4444, 0.5622, -0.6975) \\ \lambda_3 &= -0.0166 \text{ a } \mathbf{x}_3 = (0.7212, -0.6863, 0.0937).\end{aligned}$$

U následujících metod nebudeme uvádět všechny mezivýsledky maticového počítání. Dříve bylo uvedeno, že LR metoda nemusí být vždy stabilní, a proto budeme počítat rovnou s QR metodou.

Příklad 4.2. Výpočet pomocí QR metody

$$\text{Uvažujme symetrickou matici } \mathbf{B} = \begin{pmatrix} 2.25 & -0.25 & -1.25 & 2.75 \\ -0.25 & 2.25 & 2.75 & 1.25 \\ -1.25 & 2.75 & 2.25 & -0.25 \\ 2.75 & 1.25 & -0.25 & 2.25 \end{pmatrix},$$

a požadujeme přesnost výpočtu $\epsilon = 10^{-4}$. Po 93 iterací získáváme:

$$\mathbf{B}_{93} = \begin{pmatrix} 5.49999 & -0.00009 & 0 & 0 \\ -0.00009 & 5.04508 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -0.54508 \end{pmatrix}$$

$$\hat{\mathbf{Q}}_{93} = \begin{pmatrix} -0.58842 & -0.37163 & 0.39223 & -0.60150 \\ 0.39211 & -0.60158 & 0.58835 & 0.37175 \\ 0.58827 & -0.37187 & -0.39223 & -0.60150 \\ -0.39236 & -0.60142 & -0.58835 & 0.37175 \end{pmatrix}.$$

Aproximace vlastních čísel leží na diagonále matice \mathbf{B}_{93} :

$$\lambda_1 \approx 5.5, \lambda_2 \approx 5.04508, \lambda_3 \approx -1 \text{ a } \lambda_4 \approx 0.54508$$

Jelikož je matice \mathbf{B} symetrická, aproximace odpovídajících vlastních vektorů jsou sloupce matice $\hat{\mathbf{Q}}_{93}$. Pokud bychom využili QR metodu s posuny, získáme výsledek už po 11 iteracích.

Vyzkoušejme nyní zadat obecně nesymetrickou matici.

Příklad 4.3. Vypočtete vlastní čísla QR metodou

$$\text{Vezměme si matici } \mathbf{C} = \begin{pmatrix} 2.5 & -2.5 & 3 & 0.5 \\ 0 & 5 & -2 & 2 \\ -0.5 & -0.5 & 4 & 2.5 \\ -2.5 & -2.5 & 5 & 3.5 \end{pmatrix}, \text{ a zvolme přesnost } \epsilon = 10^{-3}.$$

Po 37 iteracích dostaneme:

$$\mathbf{C}_{37} = \begin{pmatrix} 6 & -0.4463 & 3.1236 & -5.1259 \\ 0.0009 & 4.9996 & -0.8701 & -2.5418 \\ 0 & 0 & 3 & -3.5777 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Výsledné aproximace vlastních čísel jsou:

$$\lambda_1 \approx 6, \lambda_2 \approx 5, \lambda_3 \approx 3 \text{ a } \lambda_4 \approx 1$$

Vlastní vektory si musíme dopočítat jinou metodou, jelikož zadaná matice \mathbf{C} není symetrická. Metodou s posuny (algoritmus 3.9) bychom výpočet zkrátily na 27 kroků.

Příklad 4.4. Jacobiho metoda

Vezměme si matici \mathbf{B} z příkladu 4.2 a uvažujme i stejnou toleranci $\epsilon = 10^{-4}$.

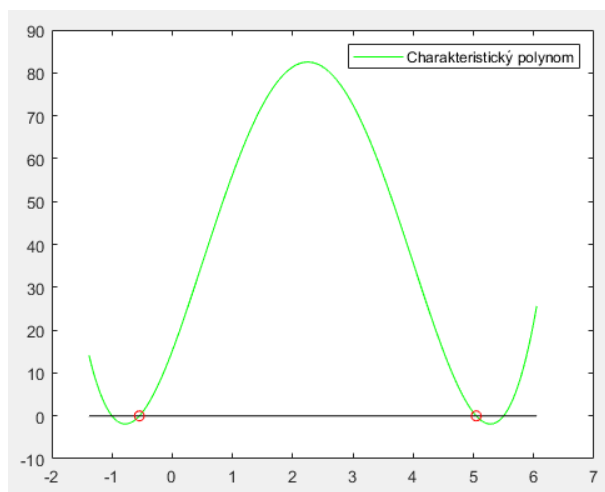
Potom po 9 iteracích dostaneme:

$$\mathbf{B}_9 = \begin{pmatrix} 5.04508 & 0 & 0 & 0 \\ 0 & 5.5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -0.54508 \end{pmatrix}$$

V tomto případě je Jacobiho metoda mnohem rychlejší než QR metoda i než metoda s posuny. Musíme si ale uvědomit, že lze použít pouze na symetrické matice.

Příklad 4.5. Ukázka metody bisekce

U metody bisekce je výhodou, že si můžeme zvolit, která vlastní čísla matice chceme spočítat. Mějme matici \mathbf{B} z předchozího příkladu a chceme znát λ_2 a λ_3 . U této metody indexy seřazují vlastní čísla podle velikosti, ne podle velikosti v absolutní hodnotě.



Obr. 2: Charakteristický polynom matice \mathbf{B}

Po výpočtu a vykreslení charakteristického polynomu nám vychází kořeny $\lambda_2 \approx 5.04506$ a $\lambda_3 \approx -0.54512$. Z obrázku jde vidět, že další kořeny se nachází v blízkosti hodnoty -1 a mezi čísly 5 a 6 .

Příklad 4.6. Lanczosova metoda

Vypočtete vlastní čísla matice $\mathbf{D} = \begin{pmatrix} 2 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 3 & -1 \end{pmatrix}$.

Za startovací vektory posloupnosti zvolte $\mathbf{x}_1 = \mathbf{y}_1 = (0, 0, 1)$ dle [4].

Pak z rovnic (12) a (13) vypočteme:

$$(\mathbf{A}\mathbf{x}_1)^T = (3, 1, -1), b_1 = -1, \mathbf{x}_2 = (3, 1, 0),$$

$$(\mathbf{A}\mathbf{x}_2)^T = (4, 4, 6), b_2 = \frac{8}{3}, \mathbf{x}_3 = (-4, \frac{4}{3}, 0),$$

$$(\mathbf{A}\mathbf{x}_3)^T = (-\frac{32}{3}, -\frac{8}{3}, 0), b_3 = \frac{1}{3}, \text{podobně}$$

$$(\mathbf{A}^T\mathbf{y}_1)^T = (1, 3, -1), \mathbf{y}_2^T = (1, 3, 0),$$

$$(\mathbf{A}^T\mathbf{y}_2)^T = (5, 1, 6), c_1 = 6, \mathbf{y}_3^T = (\frac{7}{3}, -7, 0),$$

$$(\mathbf{A}^T\mathbf{y}_3)^T = (-\frac{7}{3}, -\frac{35}{3}, 0), c_2 = -\frac{28}{9}, \text{takže}$$

$$\mathbf{T} = \begin{pmatrix} -1 & 6 & 0 \\ 1 & \frac{8}{3} & -\frac{28}{9} \\ 0 & 1 & \frac{1}{3} \end{pmatrix}$$

a pomocí rekurence (16) můžeme vypočítat charakteristický polynom

$$f_0(\lambda) = \lambda^3 - 2\lambda^2 - 5\lambda + 6,$$

jehož kořeny jsou $\lambda_1 = 3$, $\lambda_2 = -2$ a $\lambda_3 = 1$.

4.2 Porovnání metod

V této část si porovnáme všechny metody, které nám umožňují počítat všechny vlastní čísla současně. Vynecháme tedy mocninné metody, protože díky nim vypočítáme vždy jen jedno vlastní číslo. Zároveň není vhodné porovnávat maticové výpočty s hledáním kořenů polynomu, a proto metodu bisekce a Lanczosovu metodu také vynecháme. Všem třem metodám byl již věnován příklad v předchozí kapitole. Aby byly rozdíly mezi dalšími metodami znatelnější, budeme počítat s maticemi vyššího řádu, v dalších příkladech bude rozměr matic $n = 20$ a klademe toleranci na přesnost výpočtu $\epsilon = 10^{-4}$. Všechny použité matice v příkladech jsou přímo dohledatelné v MATLABu pod knihovnou *gallery*.

Příklad 4.7. Porovnání metod na řídké symetrické matici

Vytvořme si vlastní matici \mathbf{A} řádu $n = 20$. Všechny diagonální prvky položme rovny jedné. Dále nechť jsou prvky na první naddiagonále rovny dvou. Nakonec položme prvky na druhé naddiagonále rovny třem. Stejně sestrojíme i poddiagonály. Výsledná matice je tedy pětdiagonální.

použitá metoda	QR	QR s posuny	Jacobiho
počet iterací	1054	33	494

Příklad 4.8. Výpočet vlastních čísel symetrické matice

Vezměme si matici $\mathbf{B} = \text{gallery}('moler', 20)$, která je symetrická a nemá žádné nulové prvky.

použitá metoda	QR	QR s posuny	Jacobiho
počet iterací	2133	33	484

Příklad 4.9. Ukázka výpočtu na symetrické matici

Budeme pracovat se symetrickou maticí $\mathbf{C} = \text{gallery}('lehmer', 20)$. Rozdíl oproti předchozímu příkladu spočívá v tom, že všechny prvky jsou kladné a $c_{ij} \leq 1$.

použitá metoda	QR	QR s posuny	Jacobiho
počet iterací	39	29	424

Příklad 4.10. Výpočet vlastních čísel nesymetrické matice

Vezměme si matici $\mathbf{D} = \text{gallery}('lesp', 20)$, která je nesymetrická a třídiagonální.

použitá metoda	QR	QR s posuny
počet iterací	306	24

Na uvedených případech lze pozorovat, že QR metoda s posuny konverguje rychleji než klasická QR metoda. Zároveň ale můžeme vidět, že Jacobiho metoda se vzhledem k počtu kroků, nachází někde mezi nimi. To je dáno tím, že Jacobiho metoda funguje na principu nulování prvků mimo diagonálu a QR metody se snaží sestrojít horní trojúhelníkovou matici. Tudíž podmínku na přesnost musí splňovat jen polovina prvků matice oproti výpočtu Jacobiho metodou. Všechny příklady jsou zpracovány v matlaboských live skriptech a přiloženy k práci v rámci příloh. Tyto skripty obsahují řešení nejen příkladů zde uvedených, ale i několika dalších. Uvedme přehled, kdy je vhodné použít dané metody.

Normalizovaná mocnná metoda konverguje pouze pro diagonalizovatelné matice a matice s jediným dominantním vlastním číslem. Jelikož vypočítá pouze největší vlastní číslo v absolutní hodnotě, je vhodné ji využít například pokud nás zajímá stabilita řešení systémů lineárních diferenčních rovnic nebo tuhost počáteční úlohy pro obyčejnou diferenciální rovnici.

Inverzní normalizovaná mocnná metoda bude mít stejné využití jako ta předchozí. Pokud chceme znát pouze nejmenší vlastní číslo v absolutní hodnotě, pak si vybereme pro výpočet právě tuto metodu.

Inverzní normalizovaná mocnná metoda s posunem využívá počátečního odhadu hodnoty vlastního čísla. Pokud si jinou z metod nemůžeme dopočítat vlastní vektory, využijeme právě této metody.

Pomocí **QR rozkladu** můžeme vypočítat všechna vlastní čísla téměř všech matic. Nejvíce efektivní bude pro matice s relativně malými prvky (viz příklad ??).

Metoda QR rozkladu s posunem je upravená oproti QR metodě tak, aby konvergovala rychleji. V drtivém počtu příkladů je lepší využít právě metodu s posuny.

Jacobiho metoda je hojně využívána pro výpočet vlastních čísel řídkých symetrických matic.

Metoda bisekce je další metodou, která se využívá u symetrických matic. Jelikož pracuje s charakteristickým polynomem, můžeme si zvolit, kterou část spektra matice chceme získat.

Lanczosovu metoda pracuje podobně jako metoda bisekce, ale používá se pro nesymetrické matice.

5 Závěr

Práce se zaměřila na představení a ukázkou vybraných numerických metod pro aproximaci vlastních čísel. Konkrétně byly popsány mocninné metody, metody LR a QR rozkladu, Lanczosova metoda pro nesymetrické matice a Jacobiho metoda a metoda bisekce pro výpočet vlastních čísel symetrických matic. Dále byly popsány metody porovnány na rozsáhlejších maticích. Z ukázkových příkladů lze pozorovat, že k výpočtu QR metodou s posuny je třeba nejmenšího počtu kroků. V poslední kapitole je uveden přehled, kdy je výhodné danou metodu využít. V obecném případě doporučujeme kombinaci představených metod.

Mimo popsány metody lze v literatuře [3, 4] najít i mnoho dalších způsobů výpočtu vlastních čísel. Mezi další metody patří Givensova metoda, Householderova metoda, metoda supertriangulace nebo Arnoldiho metoda.

Reference

- [1] GOLUB, Gene H. a VAN LOAN, Charles F. *Matrix computations*. 3rd ed. *Johns Hopkins studies in the mathematical sciences*. Baltimore: Johns Hopkins University Press, c1996. ISBN 0-8018-5414-8.
- [2] HOUSEHOLDER, A. S. *Principles of numerical analysis*. New York: McGraw-Hill Book Company, 1953.
- [3] STOER, J. a BULIRSCH, R. *Introduction to numerical analysis*. New York: Springer Verlag, 1993. ISBN 0-387-97878-x.
- [4] RALSTON, A. *Základy numerické matematiky*. Praha: Academia, 1973.
- [5] ČERMÁK, Libor a HLAVIČKA, Rudolf. *Numerické metody*. Vydání třetí. Brno: Akademické nakladatelství CERM, 2016. ISBN 978-80-214-5437-8.
- [6] FRANČŮ, Jan. *Matematika 1*. Brno: Akademické nakladatelství CERM, 2023. ISBN 978-80-214-6174-1.

6 Seznam příloh

Funkce pro výpočet vlastních čísel

<i>Bisekce.m</i>	výpočet metodou bisekce
<i>INMM.m</i>	výpočet nejmenšího vlastního čísla
<i>INMMsP.m</i>	výpočet pomocí dobrého odhadu
<i>Jacobi.m</i>	výpočet Jacobiho metodou
<i>NMM.m</i>	výpočet největšího vlastního čísla
<i>QRrozklad.m</i>	výpočet metodou QR rozkladu
<i>QRsP.m</i>	výpočet QR metodou s posunem

Příklady

<i>InvMocMetoda.mlx</i>	inverzní mocninná metoda, výpočet příkladů 4.1, 4.8
<i>JacobihoMetoda.mlx</i>	Jacobiho metoda, výpočet příkladů 4.4, 4.7, 4.8, 4.9
<i>MetodaBisekce.mlx</i>	metoda Bisekce, výpočet příkladů 4.1, 4.5
<i>NormMocMetoda.mlx</i>	normovaná mocninná metoda, výpočet příkladů 4.1, 4.8
<i>QRMetoda.mlx</i>	metoda QR rozkladu, výpočet příkladů 4.2, 4.3, 4.7, 4.8, 4.9, 4.10