

# „WENIGER IST MEHR.“

---

EIGENPAARFILTERUNG AM BEISPIEL HERMITESCH POSITIV  
DEFINITER EIGENWERTPROBLEME

**Thorsten Matthias Lucke**  
Mathematik an der Technischen Universität Berlin

*Dem Institut für Mathematik  
der Technischen Universität Berlin  
vorgelegte Bachelorarbeit von*  
Thorsten Matthias Lucke.

*Betreuer und Erstgutachter:*  
Prof. Dr. Jörg Liesen

*Zweitgutachter:*  
Prof. Dr. Christian Mehl

*Datum der Einreichung:*  
Freitag, der 13. Oktober 2017

# Erklärung

Die selbständige und eigenhändige Anfertigung versichert an Eides statt

Berlin, den 13. Oktober 2017

---

Unterschrift



*Meiner Familie,  
der Thesisselbsthilfegruppe,  
der Hängemathe und dem 1 Café*

# Danksagung

Obschon die vorliegende Arbeit der eigenen Feder entstammt, hätte dieses Werk kaum vollendet werden können, wären nicht einige Hürden beiseite geschafft worden. So sehe ich es als meine Pflicht, den Helferinnen und Helfern des „Räumungsdienstes“ ein paar Worte der Ehrung zukommen zu lassen.

Der erste Dank gilt meinen Betreuern Prof. Dr. Jörg Liesen und Prof. Dr. Christian Mehl deren Sprechstunden stets eine Quelle der Inspiration waren. Immer hilfsbereit standen sie mir fachlich zur Seite und ließen mich von ihrer Erfahrung profitieren.

Ein weiterer Dank geht an die Mitarbeiterinnen und Mitarbeiter des studentischen Mathekaffees „1 Café“. Welche Qualen blieben mir durch die etlichen Teestündchen, Schokoriegel und zum Schlafen einladenden Sofas erspart! Im selben Atemzug danke ich den Mitgliedern der „Hängemathe“, deren kritisches Hinterfragen und Diskussionsfreudigkeit stets anregend war.

Schließlich richte ich meinen Dank an alle Lektoren, die akribisch jeden noch so hinterhältig versteckten Fehler entdeckt haben und damit das Lesen dieser Arbeit zu einem größeren Vergnügen machen.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Mathematische Grundlagen</b>	<b>5</b>
<b>3</b>	<b>Filbertechniken</b>	<b>9</b>
3.1	Rayleigh-Ritz Verfahren . . . . .	10
3.2	Konturintegration . . . . .	15
3.3	Illustration . . . . .	20
<b>4</b>	<b>Hilfsmittel: Rationale Funktionen</b>	<b>23</b>
4.1	Rayleigh-Ritz Iteration . . . . .	23
4.2	Gauß'sche Quadratur . . . . .	26
4.3	Approximation des Spektralprojektors . . . . .	28
4.4	FEAST-Algorithmus . . . . .	30
<b>5</b>	<b>Numerische Experimente</b>	<b>33</b>
5.1	Qualitätstests . . . . .	34
5.2	RKToolbox . . . . .	37
<b>A</b>	<b>Symbolverzeichnis</b>	<b>40</b>
<b>B</b>	<b>Sätze</b>	<b>41</b>
<b>C</b>	<b>Algorithmen</b>	<b>43</b>
	<b>Literatur</b>	<b>48</b>





# Kapitel 1

## Einleitung

Das Lösen von Eigenwertproblemen ist eine Standarddisziplin in der numerischen linearen Algebra. Die konkrete Problemstellung dabei lautet: Zu gegebenen Matrizen  $A, B \in \mathbb{C}^{n,n}$  sollen Paare  $(\lambda, x)$  mit  $\lambda \in \mathbb{C}$  und  $x \in \mathbb{C}^n \setminus \{0\}$  gefunden werden, welche der Gleichung

$$Ax = \lambda Bx$$

genügen. Solchen *Eigenwertgleichungen* begegnet man in ganz unterschiedlichen Kontexten. So sind sie beispielsweise bei der Bestimmung von Eigenfrequenzen oder dem Ermitteln von Fixpunkten beim Rotieren eines Fußballs<sup>1</sup> ebenso wie beim Untersuchen des PageRanks<sup>2</sup> einer Website von Bedeutung. Entsprechend strotzt der Kanon von angebotenen numerischen Lösungsmethoden vor Vielfalt und Virtuosität.<sup>3</sup>

Nun mag der Fall eintreten, da es notwendig wird, lediglich eine Teilmenge aus der Menge aller Eigenpaare eines gegebenen Problems zu untersuchen: Man stelle sich vor, man wolle das folgende Bild komprimieren, um den benötigten Speicherplatz zu reduzieren.



Abbildung 1.1: „Mind Games of Ozzy“ von Andy Prokh.

Dies lässt sich beispielsweise mit einer sogenannten *Singulärwertzerlegung*, welche von Numerikern auch als das *Schweizer Taschenmesser der numerischen linearen Algebra* bezeichnet wird, bewerkstelligen.

<sup>1</sup> Hier wird auf den bekannten *Satz vom Fußball* angespielt. Dieser besagt, dass auf einem Fußball zwei Punkte existieren, deren Raumkoordinaten zu Spielbeginn und zur Halbzeit unverändert sind – informell formuliert.

<sup>2</sup> Siehe [7, Abschnitt 2].

<sup>3</sup> Dies bestätigt sich beispielsweise bei einem Blick in das Inhaltsverzeichnis von [14].

Da es sich bei Abbildung 1.1 um ein Graustufenbild handelt, lässt sie sich als eine Matrix  $M$  auffassen, deren Einträge die Grauwerte der einzelnen Pixel repräsentieren. Mit einer Singulärwertzerlegung<sup>4</sup> lässt sich nun das Bild der Katze durch eine Folge von Matrizen niedrigeren Ranges approximieren. Man berechnet hierfür die Wurzeln der von Null verschiedenen Eigenwerte von  $M^H M$  und verwendet diese sogenannten *Singulärwerte*, um die Matrix  $M$  zu rekonstruieren. Wie genau die Singulärwerte für die Rekonstruktion verwendet werden, soll an dieser Stelle nicht weiter ausgeführt werden.

In unserem Beispiel hat  $M$  mehr als 700 Singulärwerte. Können wir uns von einigen Singulärwerten trennen und dennoch eine akzeptable Bildqualität aufrecht erhalten? Und welche Werte wollen wir behalten?

Dazu nehmen wir uns die  $n$  Singulärwerte  $\sigma_1, \dots, \sigma_n$  von  $M$  her. Ohne Einschränkung seien diese so nummeriert, dass  $\sigma_1 \geq \dots \geq \sigma_n$  gilt. Sollte dies nicht der Fall sein, nummerieren wir einfach um. Nun filtern wir nach eigenen Kriterien Teilmengen aus den Singulärwerten beziehungsweise den Eigenwerten von  $M^H M$  heraus und rekonstruieren mit deren Hilfe die Abbildung 1.1. Es sei an dieser Stelle darauf hingewiesen, dass die folgenden Beurteilungen der Bildqualität nicht ausschließlich mathematisch begründet sind. Die Eindrücke entstammen dem persönlichen Empfinden des Autors, sowie den Aussagen einer kleinen Testgruppe von Studierenden des Fachs Mathematik.

Beginnen wir mit einer mehr oder weniger willkürlichen Auswahl von Singulärwerten (SW).

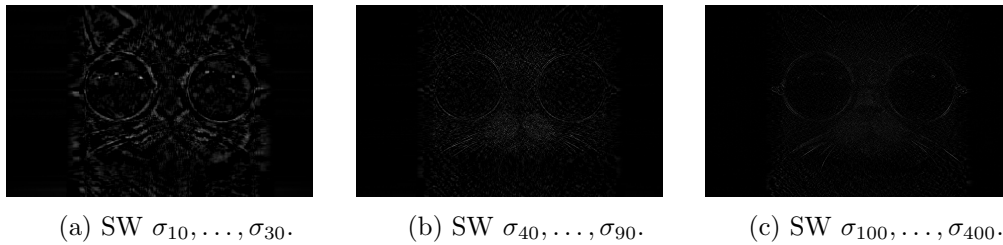


Abbildung 1.2: Rekonstruktionsversuch von Abbildung 1.1.

Die in Abbildung 1.2 gewählten Filtrierungen erscheint etwas unglücklich, da das ursprüngliche Bild kaum wieder zu erkennen ist. Daher gestalten wir die Auswahl von SW im nächsten Rekonstruktionsversuch etwas geschickter und machen uns eine wichtige Konsequenz der Singulärwertzerlegung zunutze. Sie ermöglicht es, Matrizen  $A_k$  mit der Eigenschaft

$$\|M - A_k\|_2 = \min_{\substack{B \in \mathbb{C}^{m,n} \\ \text{Rang}(B)=k}} \|M - B\|_2$$

zu finden.<sup>5</sup> Dieses Resultat wirkt unmittelbar auf die Bildqualität aus, wie in der folgenden Abbildung zu sehen ist.

---

<sup>4</sup> Eine Formulierung ist mit Satz B.1 im Anhang B gegeben.

<sup>5</sup> Diese Eigenschaft wird in Satz B.2 im Anhang B genauer formuliert.

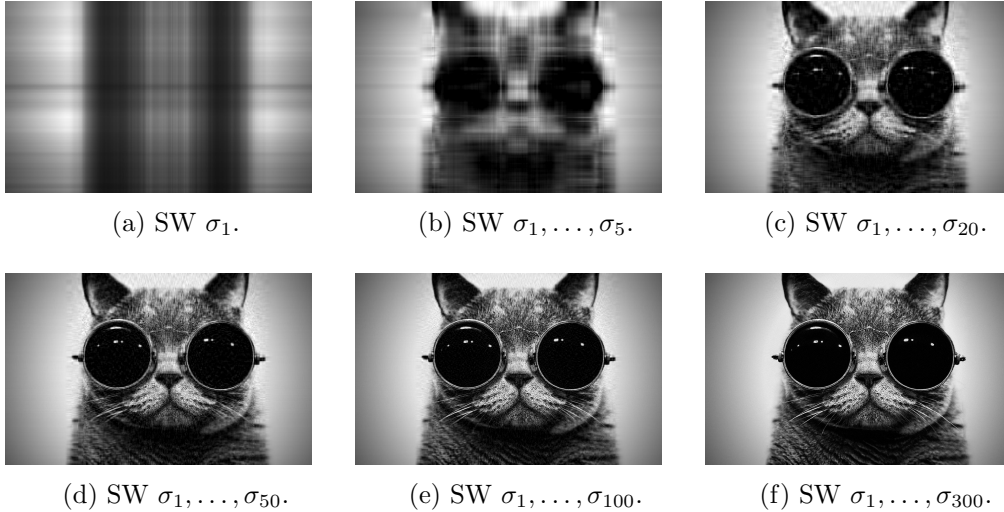


Abbildung 1.3: Erneuter Rekonstruktionsversuch von Abbildung 1.1.

Dieser Versuch wirkt optisch in der Tat überzeugender. Bereits im Bild 1.3e ist ein genauer Blick erforderlich, um die Einbüßungen bei der Komprimierung zu erkennen. Bei der Verwendung der ersten 300 SW ist es nahezu unmöglich, einen Unterschied zum Original festzustellen. Es ist also zu überlegen, ob man sich mit der Qualität von 1.3f zufrieden geben möchte und anstelle von Abbildung 1.1 abspeichert.

Nun ist das Approximieren von Bildern ein sehr spezieller Fall des Filterns von Eigenwerten und die Berechnung der Singulärwertzerlegung nicht immer zweckmäßig oder möglich. Daher setzt sich diese Arbeit mit Alternativen auseinander, die für Filtrierungen dienlich sind. Dabei werden neben den mathematischen Ideen dieser Alternativen auch Möglichkeiten der Implementation vorgestellt.

Im dritten Kapitel werden zunächst zwei Methoden präsentiert, die als Werkzeuge zum Filtern von Eigenpaaren dienlich sind. Daran wird sich nach einer kurzen analytischen Illustration dieser Verfahren eine Diskussion anschließen, bei der untersucht wird, wie es um die praktische Umsetzbarkeit der Verfahren bestellt ist. Wir werden feststellen, dass sich die vorgestellten Algorithmen kombinieren und modifizieren lassen und zu einem Verfahren führen, welches in der Literatur als FEAST-Algorithmus gehandelt wird. Zum Abschluss begeben wir uns in das „Numerik-Labor“ und werden die Verhaltensweisen der Algorithmen untersuchen.

Bevor es konkreter wird, erinnert der folgende Abschnitt an einige mathematische Grundlagen, die als helfende Handreichung das Lesen dieser Schrift mehr zur Freude, als zur Schikane machen soll.



## Kapitel 2

# Mathematische Grundlagen

Wie versprochen, widmet sich das hiesige Kapitel mit der Wiederholung und Einführung wichtigen Vokabulars, welches im Rahmen dieser Arbeit häufig zum Einsatz kommen wird. Von zentraler Bedeutung werden dabei Inhalte der linearen Algebra sein.

Um es nicht bei einer bloßen Auflistung von Definitionen zu belassen, werden zudem Resultate präsentiert, auf die wir später zurückgreifen werden. Obschon sich der Autor bemüht hat, in der Literatur gängige Notation zu benutzen, bittet er den verständnisvollen Leser bei Unklarheiten im Anhang A nachzuschlagen.

Mit den Buchstaben  $m$  und  $n$  werden wir – sofern nicht anders vermerkt – zwei natürliche Zahlen bezeichnen. Dabei wollen wir die Null aus den natürlichen Zahlen  $\mathbb{N}$  ausgeschlossen wissen. Falls die Null zugelassen ist, schreiben wir explizit  $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ . Des Weiteren bezeichnen wir mit  $\mathbb{R}$  und  $\mathbb{C}$  wie üblich die Mengen der reellen respektive komplexen Zahlen und entsprechend mit  $\mathbb{R}^{m,n}$  und  $\mathbb{C}^{m,n}$  die Mengen der reell- beziehungsweise komplexwertigen  $(m \times n)$ -Matrizen. Wir unterschlagen  $n$  im Fall  $n = 1$ .

In Anlehnung an die in MATLAB verwendete Syntax werden wir Gebrauch von der Notation  $i = m : n$  an Stelle von  $i = m, m + 1, \dots, n - 1, n$  machen. Diese Kurzschreibweise wird etwa bei der Einführung von Indizierungen zum Einsatz kommen. So lässt sich beispielsweise eine Menge  $\{x_1, x_2, \dots, x_n\}$  mit hier nicht näher bestimmten Elementen kurz durch  $\{x_k\}_{k=1:n}$  ausdrücken.

Es sei nun  $A$  eine quadratische, komplexwertige Matrix, also  $A \in \mathbb{C}^{n,n}$ . Diese wird als *hermitesch* bezeichnet, falls sie die Identität  $A = A^H$  erfüllt und ist *positiv definit*, sofern für alle Vektoren  $x \in \mathbb{C}^n \setminus \{0\}$  die Abschätzung

$$x^H A x > 0$$

gilt. Folglich werden wir eine Matrix *hermitesch positiv definit* (HPD) nennen, wenn sie sowohl hermitesch als auch positiv definit ist.

Ist  $A$  solch eine HPD-Matrix, dann nennen wir eine Menge von Vektoren  $\{x_k\}_{k=1:m} \subseteq \mathbb{C}^n$  *orthonormal bezüglich  $A$*  oder schlicht:  *$A$ -orthonormal*, falls

$$x_i^H A x_j = \delta_{i,j} := \begin{cases} 1 & \text{wenn } i = j \\ 0 & \text{wenn } i \neq j \end{cases}$$

für alle  $i, j = 1 : m$  gilt. Allgemeiner heißt eine Matrix  $X \in \mathbb{C}^{n,n}$  *orthogonal bezüglich  $A$*  oder  *$A$ -orthogonal*, falls sie

$$X^H A X = I_n := [\delta_{i,j}]_{i,j=1:n}$$

erfüllt.<sup>1</sup> Aus der Hermitizität von  $A$  und  $I_n$  folgt natürlich ebenfalls  $X A X^H = I_n$ . Für den Fall  $A = I_n$  ignorieren wir in der Formulierung den Bezug zu  $A$  und sprechen lediglich von Orthogonalität beziehungsweise Orthonormalität.<sup>2</sup>

Sei im Folgenden  $A$  wieder eine beliebige komplexwertige Matrix. Neben dieser betrachten wir nun noch eine weitere Matrix  $B \in \mathbb{C}^{n,n}$ . Unter einem  *$n$ -dimensionalen Eigenwertproblem* wollen wir die Aufgabe verstehen, Paare  $(\lambda, x) \in \mathbb{C} \times (\mathbb{C}^n \setminus \{0\})$  zu finden, die der Gleichung

$$A x = \lambda B x \tag{2.1}$$

genügen. Wenn klar ist, dass von solch einem Problem die Rede ist, werden wir anstelle des Wortes „Eigenwertproblem“ die Notation  $(A, B)$  als Bezeichnung verwenden und außerdem auf die Angabe der Dimension verzichten. Den Spezialfall  $(A, I_n)$  bezeichnen wir als *gewöhnliches Eigenwertproblem*.

Ist ein passendes Paar  $(\lambda, x)$  gefunden, welches die *Eigenwertgleichung* (2.1) löst, so nennen wir dieses *Eigenpaar von  $(A, B)$*  oder kurz: *Eigenpaar*, falls klar ist, von welchem Eigenwertproblem die Rede ist. Dabei heißt  $\lambda$  *Eigenwert von  $(A, B)$*  und  $x$  *Eigenvektor zum Eigenwert  $\lambda$  von  $(A, B)$* . Auch hier werden wir auf die Angabe des Eigenwertproblems verzichten, wenn der Kontext dies gestattet. Beim gewöhnlichen Eigenwertproblem sehen wir von der Paar-Schreibweise ab und sprechen direkt von Eigenpaaren, Eigenwerten und Eigenvektoren von  $A$ . Die Menge aller Eigenwerte von  $(A, B)$ , das sogenannte *Spektrum*, bezeichnen wir mit  $\Lambda(A, B)$  im allgemeinen und mit  $\Lambda(A)$  im gewöhnlichen Fall.

Wie auf der Titelseite angekündigt, wird sich diese Arbeit überwiegend mit der Behandlung *hermitesch positiv definiter Eigenwertprobleme* (HPD-Eigenwertprobleme) befassen. Damit seien fürderhin Eigenwertprobleme  $(A, B)$  bezeichnet, bei denen wir uns mit einer hermiteschen Matrix  $A$  sowie einer HPD-Matrix  $B$  konfrontiert sehen.

---

<sup>1</sup> Wir werden Matrizen in ihrer komponentenweisen Darstellung konsequent durch eckige Klammern, also durch „[“ und „]“, bezeichnen.

<sup>2</sup> Streng genommen sollte hier zwischen *orthogonal* und *unitär* unterschieden werden. Wir werden jedoch im Rahmen dieser Arbeit auf diese Unterscheidung verzichten und die Worte äquivalent gebrauchen.

Eigenwertprobleme dieser Art besitzen eine Reihe nützlicher Eigenschaften, an denen wir uns zu einem späteren Zeitpunkt in der Arbeit bedienen werden. Von besonderer Bedeutung wird dabei das folgende Resultat sein.

**Satz 2.1.** Ist  $(A, B)$  ein HPD-Eigenwertproblem der Dimension  $n$ , so sind alle zugehörigen Eigenwerte reell. Außerdem existiert eine  $B$ -orthogonale Matrix  $X \in \mathbb{C}^{n,n}$  sowie eine Diagonalmatrix  $\Lambda \in \mathbb{R}^{n,n}$  mit

$$A = BX\Lambda X^{-1}. \quad (2.2)$$

*Beweis.* Sei  $(\lambda, x)$  ein Eigenpaar von  $(A, B)$ . Der positiven Definitheit von  $B$  wegen gilt dann  $x^H Bx > 0$ . Aus

$$\lambda(x^H Bx) = x^H Ax = x^H A^H x = (Ax)^H x = \bar{\lambda}(Bx)^H x = \bar{\lambda}(x^H Bx)$$

folgt daher die Gleichheit  $\lambda = \bar{\lambda}$  und somit  $\lambda \in \mathbb{R}$ . Um die Existenz der im Satz angegebenen Faktorisierung (2.2) zu zeigen, ziehen wir den Beweis aus [8, Theorem 15.3.2] unterstützend zurate und ergänzen diesen durch weitere Argumente.

Zunächst nutzen wir die Hermitizität und die positive Definitheit von  $B$  aus: Mit diesen beiden Eigenschaften garantieren uns der Spektralsatz für hermitesche Matrizen<sup>3</sup> und der Satz von der Existenz der Cholesky-Zerlegung<sup>4</sup> die Faktorisierung

$$B = X_B \Lambda_B^2 X_B^H. \quad (2.3)$$

Dabei ist  $\Lambda_B \in \mathbb{R}^{n,n}$  eine diagonale Matrix mit echt positiven Einträgen und  $X_B \in \mathbb{C}^{n,n}$  orthogonal. Wir definieren nun eine weitere Matrix  $M := \Lambda_B^{-1} X_B^H A X_B \Lambda_B^{-1}$ , welche wegen

$$M^H = (\Lambda_B^{-1} X_B^H A X_B \Lambda_B^{-1})^H = (\Lambda_B^{-1})^H X_B^H A^H (X_B^H)^H (\Lambda_B^{-1})^H = \Lambda_B^{-1} X_B^H A X_B \Lambda_B^{-1} = M$$

hermitesch ist. Nutzen wir erneut den Spektralsatz, so finden wir eine orthogonale Matrix  $X_M \in \mathbb{C}^{n,n}$  und eine Diagonalmatrix  $\Lambda_M \in \mathbb{R}^{n,n}$ , sodass

$$M = X_M \Lambda_M X_M^H$$

gilt. Setzen wir nun  $X := X_B \Lambda_B^{-1} X_M$ , erhalten wir die Identitäten

$$\begin{aligned} X^H A X &= X_M^H \Lambda_B^{-1} X_B^H A X_B \Lambda_B^{-1} X_M = X_M^H M X_M = \Lambda_M \\ X^H B X &= X_M^H \Lambda_B^{-1} X_B^H B X_B \Lambda_B^{-1} X_M \stackrel{(2.3)}{=} X_M^H X_M = I_n. \end{aligned}$$

Also ist  $X$  schon  $B$ -orthogonal. Schließlich folgt Gleichung (2.2) aus

$$X^{-1} B^{-1} A X = X^{-1} B^{-1} (X^H)^{-1} X^H A X = \Lambda_M.$$

durch Umstellen. □

<sup>3</sup> Siehe Satz B.3 im Anhang B.

<sup>4</sup> Siehe Satz B.4 im Anhang B.

Die Existenz der Faktorisierung (2.2) werden wir später nutzen, um bei hermitesch positiv definiten Eigenwertproblem  $(A, B)$  anstelle von (2.1) das Lösen von

$$AX = BXA \quad (2.4)$$

verlangen zu können. Diese Notation wird häufig bei der Ausformulierung von Algorithmen in Erscheinung treten.

Fahren wir fort mit einer beliebig gewählten Matrix  $A \in \mathbb{C}^{m,n}$ . Diese induziert bekannterweise eine lineare Abbildung  $\mathcal{A}: \mathbb{C}^n \rightarrow \mathbb{C}^m$  vermöge  $\mathcal{A}(x) := Ax$ . Wir werden fortan nicht mehr zwischen der Abbildung  $\mathcal{A}$  und der zugehörigen Matrix unterscheiden und wollen  $A$  sowohl als Element des Vektorraumes  $\mathbb{C}^{m,n}$  als auch als lineare Abbildung von  $\mathbb{C}^n$  nach  $\mathbb{C}^m$  begreifen. Demnach sind Notationen wie  $\text{Bild}(A)$  und ähnliche abbildungsrelevante Schreibweisen wohldefiniert.

Nehmen wir uns zu guter Letzt einen  $m$ -dimensionalen Unterraum  $\mathcal{U} \subseteq \mathbb{C}^n$  her und bemühen erneut eine Matrix  $A \in \mathbb{C}^{n,n}$ . Der eben eingeführte Unterraum heißt *A-invariant*, falls die Inklusion  $A\mathcal{U} \subseteq \mathcal{U}$  gilt. Ist dann  $U \in \mathbb{C}^{n,m}$  eine Matrix mit  $\text{Bild}(U) = \mathcal{U}$ , so existiert folglich eine Matrix  $V \in \mathbb{C}^{m,m}$  mit  $AU = UV$ .

Ausgestattet mit den eben besprochenen Definitionen und Resultaten können wir nun mit dem Hauptteil dieser Arbeit beginnen.



## Kapitel 3

# Filtertechniken

In der Einleitung dieser Arbeit wurde bereits häufiger vom *Filtern* gesprochen. Bevor wir uns mit Techniken des Filterns beschäftigen können, wollen wir formal festhalten, was unter einem Filter zu verstehen ist.

**Definition 3.1** (Filter). *Sei  $M$  eine nicht leere Menge. Eine Abbildung der Gestalt*

$$\mathfrak{F}: M \rightarrow \{0, 1\}, \quad m \mapsto \mathfrak{F}(m)$$

*heißt Filterfunktion oder Filter auf der Menge  $M$ . Die Zuordnung  $m \mapsto \mathfrak{F}(m)$  heißt Filtern von  $m$  aus  $M$ .*

Damit können wir uns nun in diesem Kapitel der Frage widmen, ob und wie man aus der Menge aller Eigenpaare eines Eigenwertproblems mit der zugehörigen Eigenwertgleichung

$$Ax = \lambda Bx$$

eine gewünschte Teilmenge auswählen kann. Dabei wollen wir an die Matrizen  $A$  und  $B$  zunächst keine weiteren Anforderungen stellen.

Ist  $M$  entweder die Potenzmenge des Spektrums oder die Menge aller Eigenpaare von  $(A, B)$  und  $\mathfrak{F}$  eine Filterfunktion auf  $M$ , so wollen wir vereinbaren, dass wir stets auf der Suche nach der Urbildmenge  $\mathfrak{F}^{-1}(\{1\})$  sind. Wollen wir etwa die Menge  $G$  aller ganzzahligen Eigenwerte eines gewissen Eigenwertproblems finden, so gelte  $\mathfrak{F}^{-1}(\{1\}) = G$ . In den meisten Fällen werden wir zwar nicht von einer konkreten Filterfunktion der oben definierten Art sprechen, wollen sie aber als latent vorhanden wissen.

Die Methoden zur Bestimmung von Teilmengen von Eigenpaaren oder Eigenwerten sind ebenso vielfältig wie die Möglichkeiten, das gesamte Spektrum und entsprechende Eigenvektoren zu bestimmen. Wir wollen aber im Rahmen dieser Arbeit aus dem Katalog der Verfahren lediglich zwei diskutieren: *Rayleigh-Ritz Verfahren* und *Konturintegration*.

### 3.1 Rayleigh-Ritz Verfahren

Die Idee dieser Klasse von Verfahren ist die Approximation des von den gesuchten Eigenvektoren aufgespannten Unterraums. Bevor wir uns mit beliebigen Eigenwertproblemen  $(A, B)$  auseinandersetzen, betrachten wir den Spezialfall  $B = I_n$  und wenden uns der entsprechenden Eigenwertgleichung

$$Ax = \lambda x$$

zu. Sei im Folgenden  $\mathcal{S}_m \subseteq \mathbb{C}^n$  ein  $m$ -dimensionaler Unterraum. Dieser zunächst nicht näher bestimmte *Suchraum* wird als Grundlage für das Verfahren gewählt. Gesucht sind nun Paare  $(\tilde{\lambda}, \tilde{x}) \in \mathbb{C} \times (\mathcal{S}_m \setminus \{0\})$ , welche wir als approximierten Lösungen des Eigenproblems verstehen wollen, die die Eigenschaft

$$s^H(A\tilde{x} - \tilde{\lambda}\tilde{x}) = 0 \quad (3.1)$$

für alle  $s \in \mathcal{S}_m$  erfüllen. Das *Residuum*  $(A\tilde{x} - \tilde{\lambda}\tilde{x})$  soll also orthogonal auf dem Suchraum stehen. Paare, die diesem Anliegen nachkommen, werden auch *Ritz-Paare von A bezüglich des Suchraums  $\mathcal{S}_m$*  genannt.

Wir wollen nun annehmen, dass mit der Menge  $\{s_i\}_{i=1:m} \subseteq \mathbb{C}^n$  eine Basis orthonormaler Vektoren (ONB) des Unterraums  $\mathcal{S}_m$  gegeben ist. Setzen wir dann  $S_m := [s_i]_{i=1:m} \in \mathbb{C}^{n,m}$ , so muss wegen  $\tilde{x} \in \mathcal{S}_m$  ein Vektor  $y \in \mathbb{C}^m$  mit  $S_m y = \tilde{x}$  existieren. Die Forderung (3.1) ist dann zu der Gleichung

$$S_m^H(A S_m y - \tilde{\lambda} S_m y) = 0$$

äquivalent. Als direkte Konsequenz dieser Umformulierung erhalten wir unter Ausnutzung der Orthogonalität der Spalten von  $S_m$  mit

$$(S_m^H A S_m) y = \tilde{\lambda} y \quad (3.2)$$

die Eigenwertgleichung zum Eigenwertproblem  $(S_m^H A S_m, I_m)$ . Jede Lösung  $(\tilde{\lambda}, y)$  von (3.2) liefert dann nach Konstruktion mit  $(\tilde{\lambda}, S_m y)$  ein Ritz-Paar des gewöhnlichen Eigenwertproblems bezüglich des Suchraums  $\mathcal{S}_m$ .

In Abhängigkeit von der Wahl des Suchraumes variiert natürlich die Güte der Approximation. Ist etwa  $\mathcal{S}_m$  durch Eigenvektoren aufgespannt, so ist jedes Ritz-Paar schon ein Eigenpaar. Wir werden später aus den Untersuchungen des Eigenwertproblems für beliebiges  $B$  folgern, dass bereits im Falle der  $A$ -Invarianz von  $\mathcal{S}_m$  jedes Ritz-Paar von  $A$  bezüglich  $\mathcal{S}_m$  bereits ein Eigenpaar von  $A$  ist.

Die eben skizzierte Vorgehensweise zur Berechnung von Ritz-Paaren des gewöhnlichen Eigenwertproblems lässt sich algorithmisch wie folgt zusammenfassen.

---

**Algorithmus 1** Berechnung von Ritz-Paaren (Vgl. [11, Algorithmus 4.5, S. 98])

---

**Input:** Matrix  $A$  und Suchraum  $\mathcal{S}_m$

**Output:** Ritz-Paare von  $A$  bzgl.  $\mathcal{S}_m$

- 1: Berechne eine ONB  $\{s_i\}_{i=1:m}$  von  $\mathcal{S}_m$  und setze  $S_m \leftarrow [s_i]_{i=1:m}$ .
  - 2: Setze  $\tilde{A} \leftarrow S_m^H A S_m$ .
  - 3: Finde Eigenpaare von  $\tilde{A}$  und wähle die gewünschten  $k$  Paare  $(\tilde{\lambda}_i, y_i)_{i=1:k}$ ,  $k \leq m$  aus.
  - 4: Gib Ritz-Paare  $(\tilde{\lambda}_i, S_m y_i)_{i=1:k}$  aus.
- 

Die Idee dieser Methode ist also simpel: Transformiere das Eigenwertproblem mit einer gewissen Matrix in ein anderes Eigenwertproblem und benutze dessen Lösungen, um Ritz-Paare des ursprünglichen Problems zu erhalten.

Doch wozu die Mühe, das ursprüngliche Eigenwertproblem in ein anderes Eigenwertproblem zu überführen? Zwar gelingt es, aus dem transformierten Problem (3.2) Ritz-Paare zu extrahieren, aber wäre nicht auch denkbar, sämtliche Eigenpaare von  $A$  zu approximieren und die zum Unterraum  $\mathcal{S}_m$  korrespondierende Teilmenge direkt auszuwählen?

Dies mag in Einzelfällen in der Tat sinnvoller sein. Sprechen wir allerdings von Matrixdimensionen hinreichender Größe, ist eine vollständige Berechnung aller Eigenpaare mitunter ein sehr zeitintensives Vergnügen, wie wir im fünften Kapitel sehen werden.

Bei genauerer Betrachtung der Gleichung (3.2) fällt auf, dass die Matrix  $(S_m^H A S_m) \in \mathbb{C}^{m,m}$  im Falle  $n \gg m$  ein deutlich kleineres Format hat als die Matrix  $A$  im ursprünglichen Problem. Man darf hier also erwarten, dass die benötigte Laufzeit zur Bestimmung der Ritz-Paare mit dem Algorithmus 1 geringer ist als beim Approximieren sämtlicher Eigenpaare von  $A$ . Auch das wird im fünften Kapitel anhand ausgewählter Beispiele vorgeführt.

Nun, da wir uns mit der Idee der Rayleigh-Ritz Methode vertraut gemacht haben, erweitern wir die obige Theorie auf Probleme der Art  $(A, B)$  für beliebiges  $B$  und betrachten entsprechend die Eigenwertgleichung

$$Ax = \lambda Bx. \quad (3.3)$$

Dabei gehen wir ganz analog zum gewöhnlichen Eigenwertproblem vor und betrachten wieder einen  $m$ -dimensionaler Suchraum  $\mathcal{S}_m \subseteq \mathbb{C}^n$ . Gefunden werden sollen dieses Mal Paare  $(\tilde{\lambda}, \tilde{x}) \in \mathbb{C} \times (\mathcal{S}_m \setminus \{0\})$ , die der Orthogonalitätsbedingung

$$A\tilde{x} - \tilde{\lambda}B\tilde{x} \perp \mathcal{S}_m \quad (3.4)$$

genügen.

Durch die Wahl eines geeigneten Vektors  $y \in \mathbb{C}^m$  kann die Näherungslösung  $\tilde{x}$  wie zuvor durch das Produkt  $S_m y$  ersetzt werden, wobei die Spalten von  $S_m$  erneut eine ONB des Suchraums  $\mathcal{S}_m$  bilden. Damit lässt sich die zu (3.4) äquivalente Forderung

$$S_m^H (AS_m y - \tilde{\lambda} BS_m y) = 0.$$

aufstellen. Wie bereits beim gewöhnlichen Eigenwertproblem lässt sich nun aus jeder Lösung  $(\tilde{\lambda}, y)$  von

$$(S_m^H AS_m)y = \tilde{\lambda} (S_m^H BS_m)y. \quad (3.5)$$

mit  $(\tilde{\lambda}, S_m y)$  ein Ritz-Paar für das Eigenwertproblem  $(A, B)$  gewinnen.

Da mit der Matrix  $B$  ein weiterer Darsteller berücksichtigt werden muss, zieht dies als Konsequenz eine Anpassung des Algorithmus 1 nach sich. Wir können diesen in der folgenden Manier abwandeln.

---

**Algorithmus 2** Berechnung von Ritz-Paaren

---

**Input:** Matrizen  $A, B$  und Suchraum  $\mathcal{S}_m$

**Output:** Ritzpaare von  $(A, B)$  bzgl.  $\mathcal{S}_m$

- 1: Berechne ONB  $\{s_i\}_{i=1:m}$  von  $\mathcal{S}_m$  und setze  $S_m \leftarrow [s_i]_{i=1:m}$ .
  - 2: Setze  $\tilde{A} \leftarrow S_m^H AS_m$ ,  $\tilde{B} \leftarrow S_m^H BS_m$ .
  - 3: Finde Eigenpaare von  $(\tilde{A}, \tilde{B})$  und wähle die gewünschten  $k$  Paare  $(\tilde{\lambda}_i, y_i)_{i=1:k}$  für  $k \leq m$  aus.
  - 4: Gib Ritz-Paare  $(\tilde{\lambda}_i, S_m y_i)_{i=1:k}$  aus.
- 

Bei den beiden vorgestellten Algorithmen ist anzumerken, dass zunächst nicht von einer Eigenpaarfilterung im Sinne der Definition 3.1 gesprochen werden kann. Schließlich erhalten wir unter Umständen kein exaktes Eigenpaar, sondern eben nur einen Ersatz. Um eine zufriedenstellende Qualität dieses Ersatzes gewährleisten zu können, ist es erforderlich, die Wahl des Suchraums nicht dem Geschick alleine zu überlassen, sondern eine gegebenenfalls weniger geschickte Wahl algorithmisch kompensieren zu können.

Dies kann zum Beispiel durch das Einführen einer Iterationsvorschrift gelingen, die den Suchraum in jedem Iterationsschritt ändert. Damit lassen sich zu berücksichtigende Toleranzen, beispielsweise eine maximale euklidische Abweichung der Ritz-Werte zu den Eigenwerten oder Vorgaben bezüglich des Winkels zwischen Eigenraum und dem Unterraum der Ritz-Vektoren, einhalten. Wir werden diese Idee zu einem späteren Zeitpunkt in dieser Arbeit weiterverfolgen.

Wir wollen uns nun vorerst von der Algorithmik verabschieden und einige Beobachtungen festhalten. Es wurde behauptet, dass die Invarianz des Suchraumes  $\mathcal{S}_m$  unter  $A$  dazu führt, dass jedes Ritz-Paar bezüglich  $\mathcal{S}_m$  ein Eigenpaar von  $A$  ist. Doch warum ist das so? Dies zu beantworten verpflichtet sich der folgende Satz.

**Satz 3.2.** Neben zwei Matrizen  $A, B \in \mathbb{C}^{n,n}$  – wobei wir Hermitizität und positive Definitheit von  $B$  fordern – sei ein  $m$ -dimensionaler Unterraum  $\mathcal{S}_m \subseteq \mathbb{C}_n$  gegeben. Ist dieser invariant unter  $B^{-1}A$ , so ist jedes Ritz-Paar von  $B^{-1}A$  bezüglich  $\mathcal{S}_m$  auch ein Eigenpaar von  $(A, B)$ .

*Beweis.* Beginnen wir mit einer ONB  $\{s_i\}_{i=1:m} \subseteq \mathcal{S}_m$  des Unterraums  $\mathcal{S}_m$  und setzen wie bisher  $S_m := [s_i]_{i=1:m} \in \mathbb{C}^{n,m}$ . Aufgrund der Invarianz von  $\mathcal{S}_m$  unter  $B^{-1}A$  muss eine Matrix  $V_m \in \mathbb{C}^{m,m}$  existieren, welche die Gleichung

$$B^{-1}AS_m = S_mV_m \quad (3.6)$$

erfüllt. Insbesondere folgt unter Ausnutzung der Orthogonalität der Spalten von  $S_m$  die Identität  $S_m^H B^{-1}AS_m = V_m$ . Sind nun  $\lambda \in \mathbb{C}$  und  $y \in \mathbb{C}^m \setminus \{0\}$  so gewählt, dass  $(\lambda, S_my)$  ein Ritz-Paar von  $B^{-1}A$  bezüglich  $\mathcal{S}_m$  ist, so ist  $(\lambda, y)$  ein Eigenpaar von  $V_m$ . Es folgt daher mit (3.6) die Gleichung

$$B^{-1}AS_my = S_mV_my = \lambda S_my$$

und schließlich auch die Behauptung durch Umstellen.  $\square$

Aus dem eben bewiesenen Resultat können wir nun für das gewöhnliche Eigenwertproblem unmittelbar das folgende Korollar ableiten.

**Korollar 3.3.** Ist  $A \in \mathbb{C}^{n,n}$  und  $\mathcal{S}_m \subseteq \mathbb{C}^n$  ein  $m$ -dimensionaler  $A$ -invarianter Unterraum, so ist jedes Ritz-Paar von  $A$  bezüglich  $\mathcal{S}_m$  ein Eigenpaar von  $A$ .

*Beweis.* Betrachte für  $B := I_n$  das Eigenwertproblem  $(A, B)$  mit der zugehörigen Eigenwertgleichung

$$Ax = \lambda Bx = \lambda x.$$

Die Aussage folgt dann aus dem vorigen Satz.  $\square$

Wenden wir uns einem weiteren Resultat zu. Zwischen dem Rayleigh-Ritz Verfahren und dem Konzept der Projektion besteht ein enger Zusammenhang. Um zu dieser Einsicht zu gelangen, bedarf es ein wenig Vorbereitung.

**Satz 3.4.** Es sei  $B \in \mathbb{C}^{n,n}$  eine HPD-Matrix und  $\mathcal{S}_m \subseteq \mathbb{C}^n$  ein  $m$ -dimensionaler Unterraum. Sei weiter  $\{s_i\}_{i=1:m} \subseteq \mathcal{S}_m$  eine Basis  $B$ -orthonormaler Vektoren, das heißt, die Matrix  $S_m := [s_i]_{i=1:m} \in \mathbb{C}^{n,m}$  erfülle die Gleichung  $S_m^H BS_m = I_m$ . Dann ist die von der Matrix  $P := S_m S_m^H B \in \mathbb{C}^{n,n}$  induzierte lineare Abbildung

$$p: \mathbb{C}^n \rightarrow \mathbb{C}^n, x \mapsto S_m S_m^H Bx$$

eine  $B$ -orthogonale Projektion auf den Unterraum  $\mathcal{S}_m$ . Außerdem gilt für alle  $x \in \mathbb{C}^n$  die Identität

$$\|x - p(x)\|_B = \min_{y \in \mathcal{S}_m} \|x - y\|_B.$$

*Beweis.* Nach Konstruktion gilt  $p(\mathbb{C}^n) \subseteq \mathcal{S}_m$ . Aus der  $B$ -Orthogonalität von  $S_m$  folgt für alle  $x \in \mathbb{C}^n$

$$p^2(x) = S_m S_m^H B S_m S_m^H B x = S_m S_m^H B x = p(x).$$

Ist nun  $x \in \mathbb{C}^n$  vorgegeben, dann folgt unter erneuter Ausnutzung von  $S_m^H B S_m = I_m$

$$S^H B(x - p(x)) = S^H B(I_n - S_m S_m^H B)x = (S^H B - S^H B S S^H B)x = 0$$

und somit  $x - p(x) \perp_B \mathcal{S}_m$ . Damit ist gezeigt, dass  $p$  eine  $B$ -orthogonale Projektion auf  $\mathcal{S}_m$  ist. Die Optimierungsaufgabe wird schließlich wegen

$$\begin{aligned} \|x - y\|_B^2 &= \|x - p(x) + p(x) - y\|_B^2 \\ &= \|x - p(x)\|_B^2 + \|p(x) - y\|_B^2 \\ &\geq \|x - p(x)\|_B^2 \end{aligned}$$

gelöst. Dabei ist bei der zweiten Gleichheit zu berücksichtigen, dass  $x - p(x) \in \mathcal{S}_m^{\perp_B}$  und  $p(x) - y \in \mathcal{S}_m$  gilt.  $\square$

Mit Hilfe der im Satz eingeführten Projektionsmatrix  $P$  lässt sich die Gleichung (3.5) umformulieren. Durch Linksmultiplikation mit  $B^H S_m$  und dem Einschub der Identität  $S_m^H B S_m$  gilt nämlich

$$B^H S_m (S_m^H A S_m) (S_m^H B S_m) y = \tilde{\lambda} B^H S_m (S_m^H B S_m) (S_m^H B S_m) y$$

und folglich

$$P^H A P S_m y = \tilde{\lambda} P^H B P S_m y.$$

Wenn wir uns an dieser Stelle erinnern, dass  $\tilde{x}$  durch  $S_m y$  ersetzt wurde, so erhalten wir

$$P^H A P \tilde{x} = \tilde{\lambda} P^H B P \tilde{x}.$$

Im Fall der gewöhnlichen Eigenwertgleichung erhalten wir speziell

$$P^H A P \tilde{x} = P A P \tilde{x} = \tilde{\lambda} \tilde{x}.$$

Die Behauptung, es würde ein Zusammenhang zu Projektionen bestehen, wurde also in der Tat zu Recht aufgestellt.

Nach diesen theoretischen Ausführungen fragt sich der anwendungsorientierte Leser zu Recht, welcher Nutzen aus den obigen Einsichten gezogen werden kann. Dafür wenden wir uns erneut dem Rayleigh-Ritz Verfahren zu und betrachten nun den Spezialfall eines hermitesch positiv definiten Eigenwertproblems  $(A, B)$ .

Angenommen, uns stünde mit dem Suchraum  $\mathcal{S}_m$  bereits ein von Eigenvektoren aufgespannter Unterraum zur Verfügung. Dann sichert uns Satz 2.1 die Existenz einer Matrix  $S_m := [s_i]_{i=1:m} \in \mathbb{C}^{n,m}$  mit  $B$ -orthonormalen Spalten aus Eigenvektoren von  $(A, B)$  und  $\text{Bild}(S_m) = \mathcal{S}_m$ , sowie die Existenz einer reellen Diagonalmatrix  $\Lambda_m := \text{diag}(\lambda_i)_{i=1:m} \in \mathbb{R}^{m,m}$  mit

$$AS_m = BS_m\Lambda_m.$$

Dabei entsprechen die Diagonaleinträge von  $\Lambda_m$  gerade den zu den Spalten von  $S_m$  korrespondierenden Eigenwerten von  $(A, B)$ . Folglich wäre das Bild des Spektralprojektors  $P = S_m S_m^H B$  nach Konstruktion wegen

$$B^{-1}AP = B^{-1}(AS_m)S_m^H B = B^{-1}(BS_m\Lambda)S_m^H B = S_m\Lambda S_m^H B = \left( \sum_{i=1}^m \lambda_i s_i s_i^H \right) \cdot B$$

ein  $(B^{-1}A)$ -invarianter Unterraum. Daher erscheint es wegen Satz 3.2 sinnvoll, den Spektralprojektor  $P$  in das Rayleigh-Ritz-Verfahren zu integrieren. Da jedoch der Unterraum  $\mathcal{S}_m$  im Allgemeinen nicht bekannt ist, stellt sich die Frage, ob es überhaupt sinnvoll ist nach dem Projektor zu suchen. Wir werden jedoch alsbald feststellen, dass die Matrix  $S_m S_m^H$  mit Methoden der Funktionentheorie berechnbar ist.

## 3.2 Konturintegration

Betrachten wir ein  $n$ -dimensionales Eigenwertproblem  $(A, B)$ . Zu einer gegebenen Teilmenge  $\Lambda_1$  vom Spektrum des besagten Problems wollen wir den Spektralprojektor konstruieren, welcher Vektoren auf den von den zu  $\Lambda_1$  korrespondierenden Eigenvektoren aufgespannten Unterraum  $\mathcal{S}_1$  projiziert. Dazu schränken wir uns vorerst auf das gewöhnliche Eigenwertproblem  $(A, I_n)$  ein und orientieren uns in den folgenden Ausführungen an der Herleitung von [18, Abschnitt 4.9]. Dabei nehmen wir uns die Freiheit, die Notation an die hiesige Arbeit anzupassen und gelegentlich von den vorgeschlagenen Schreibweisen abzuweichen.

Wir wollen annehmen, dass die Zahl  $m$  der Dimension von  $\mathcal{S}_1$  entspricht. Seien Matrizen  $A_1 \in \mathbb{C}^{m,m}$ ,  $A_2 \in \mathbb{C}^{n-m,n-m}$  und  $S \in \mathbb{C}^{n,n}$  derart vorgegeben, dass  $\Lambda(A_1) = \Lambda_1$  einerseits und  $\Lambda_2 := \Lambda(A_2)$  mit  $\Lambda_2 \cap \Lambda_1 = \emptyset$  andererseits gelten und eine Faktorisierung der Art

$$A = S \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} S^{-1} \quad (3.7)$$

existiert.<sup>1</sup> Dann können wir zwei Matrizen  $S_1 \in \mathbb{C}^{n,m}$  mit  $\text{Bild}(S_1) = \mathcal{S}_1$  und  $S_2 \in \mathbb{C}^{n,n-m}$  finden, welche die Partitionierung  $S = [S_1, S_2]$  erlauben.

<sup>1</sup> Diese Zerlegung kann beispielsweise mit Hilfe der Jordan'schen Normalform erreicht werden.

Durch Definition der Matrix

$$P_1 := S \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix} S^{-1} \quad (3.8)$$

erhalten wir dann eine Projektion auf  $\mathcal{S}_1$ : Nach Konstruktion gilt  $P_1 x \in \mathcal{S}_1$  für alle  $x \in \mathbb{C}^n$  und  $P_1^2 = P_1$ . Andererseits liefert die Matrix

$$P_2 := I_n - P_1$$

eine Projektion auf  $\mathcal{S}_2 := \text{Bild}(S_2)$ : Es gilt  $P_2 x \in \mathcal{S}_2$  für alle  $x \in \mathbb{C}^n$  und

$$P_2^2 = (I_n - P_1)^2 = I_n - 2P_1 + P_1^2 = I_n - P_1.$$

Da sich jeder Vektor  $x \in \mathbb{C}^n$  in

$$x = P_1 x + (I_n - P_1)x$$

dekomponieren lässt und trivialerweise  $P_2 x \perp \mathcal{S}_2^\perp$  gilt, können wir schließen, dass  $P_1$  orthogonal bezüglich  $\mathcal{S}_2^\perp$  auf  $\mathcal{S}_1$  abbildet.

Der Projektor  $P_1$  aus (3.8) lässt sich mit Hilfe eines Integrals darstellen. Bevor wir dazu kommen, benötigen wir zusätzliches Vokabular.

**Definition 3.5** (Jordan-Kurve). *Es sei  $\gamma: \{x \in \mathbb{R}^2 : \|x\|_2 = 1\} \rightarrow \mathbb{R}^2$  eine Abbildung, die homöomorph auf ihr Bild abbildet. Dann heißt  $\Gamma := \text{Bild}(\gamma)$  Jordan-Kurve.*

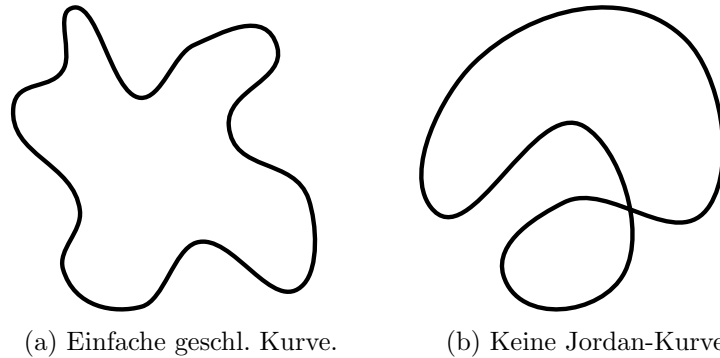


Abbildung 3.1: Skizze zweier Kurven.

Insbesondere sind Jordan-Kurven also geschlossene und überschneidungsfreie Kurven.

Da wir über solche Jordan-Kurven integrieren wollen, führen wir noch den Begriff der *orientierten Kurve* ein. Dabei wollen wir es jedoch der Einfachheit halber bei einer informellen Beschreibung belassen. Eine Jordan-Kurve heißt *positiv orientiert*, falls sie in ihrer Parametrisierung im mathematisch positiven Sinne „durchlaufen“ wird (Vgl. Abb. 3.2).



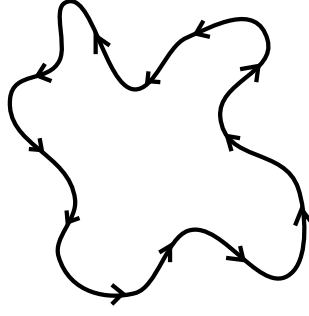


Abbildung 3.2: Skizze einer positiv orientierten Jordan-Kurve.

Wir können nun den Gedanken aus [18, Abschnitt 4.9] weiter folgen und betrachten für  $\Omega \subseteq \mathbb{C}^n$  die durch

$$G: \Omega \setminus \Lambda(A) \rightarrow \mathbb{C}^{n,n}, t \mapsto (tI_n - A)^{-1}$$

definierte *Resolventenfunktion* – auch *Green-Funktion* genannt (Vgl. [10]). Angenommen, die Menge  $\Lambda_1$  liegt im Inneren eines von einer positiv orientierten Jordan-Kurve  $\Gamma$  umschlossenen Gebiets und  $\Lambda_2$  im Inneren des Komplementes dieses Gebiets.<sup>2</sup> Dann gilt die Identität

$$P_1 = \frac{1}{2\pi\iota} \int_{\Gamma} G(t) dt \quad (3.9)$$

beziehungsweise unter Ausnutzung der Faktorisierung (3.7)

$$P_1 = \frac{1}{2\pi\iota} S \begin{bmatrix} \int_{\Gamma} (tI_m - A_1)^{-1} dt & 0 \\ 0 & \int_{\Gamma} (tI_{n-m} - A_2)^{-1} dt \end{bmatrix} S^{-1}. \quad (3.10)$$

Hierbei ist das Integral eintragsweise zu verstehen und  $\iota$  als imaginäre Einheit zu lesen. Da die Menge  $\Lambda_2$  nach Vorgabe außerhalb des umrandeten Gebietes liegt, folgt aus dem Cauchy'schen Integralsatz<sup>3</sup>

$$\int_{\Gamma} (tI_{n-m} - A_2)^{-1} dt = 0_{n-m}.$$

Um die Begründung, dass die Gleichheit (3.10) korrekt ist, fortsetzen zu können, benötigen wir die folgende Erkenntnis.

**Satz 3.6.** Ist  $M \in \mathbb{C}^{n,n}$  mit  $\|M\|_2 < 1$ , so ist  $I_n - M$  invertierbar und es gilt

$$(I_n - M)^{-1} = \sum_{j=0}^{\infty} M^j$$

*Beweis* (Vgl. [3, Abschnitt 2.3.4]). Zunächst zur Regularität: Angenommen, die Matrix  $I_n - M$  wäre nicht invertierbar. Dann muss ein von Null verschiedener Vektor  $x \in \mathbb{C}^n$  existieren, mit  $(I_n - M)x = 0$ . Wir erhalten insbesondere  $x = Mx$ . Da die induzierte 2-Norm für Matrizen bekanntermaßen submultiplikativ ist, gilt  $\|x\|_2 \leq \|M\|_2 \|x\|_2$ . Dann führt aber die Konsequenz  $\|M\|_2 \geq 1$  zum Widerspruch.

<sup>2</sup> Man beachte, dass im Folgenden stillschweigend die Identifikation  $\mathbb{R}^2 \cong \mathbb{C}$  benutzt wird.

<sup>3</sup> Siehe Satz B.5 im Anhang B.

Nun zur Potenzreihenentwicklung: Nutzen wir erneut die Submultiplikativität der 2-Norm, so erhalten wir für alle  $k \in \mathbb{N}$  die Abschätzung  $\|M^k\|_2 \leq \|M\|_2^k$ . Wegen  $\|M\|_2 < 1$  existiert daher der Grenzwert

$$\lim_{k \rightarrow \infty} M^k = 0_n.$$

Schließlich folgt die Behauptung des Satzes aus

$$\sum_{j=0}^{\infty} M^j (I_n - M) = \lim_{k \rightarrow \infty} \sum_{j=0}^k M^j (I_n - M) = \lim_{k \rightarrow \infty} I_n - M^{k+1} = I_n$$

durch einfaches Umstellen. □

**Korollar 3.7.** Mit den obigen Notationen folgt für alle  $t \in \mathbb{C}$  mit  $|t| > \|A_1\|_2$  die Identität

$$(tI_m - A_1)^{-1} = \frac{1}{t} \sum_{j=0}^{\infty} (t^{-1} A_1)^j.$$

*Beweis.* Man mache sich die Identität

$$(tI_m - A_1)^{-1} = \frac{1}{t} (I_m - t^{-1} A_1)^{-1}$$

klar und benutze den Satz 3.6. □

Wenden wir uns nun wieder der Gleichung (3.10) zu. Es bleibt zu zeigen, dass

$$\frac{1}{2\pi i} \int_{\Gamma} (tI_m - A_1)^{-1} dt = I_m \tag{3.11}$$

gilt. Dazu halten wir uns nach wie vor an die Ausführungen von [18, Abschnitt 4.9]. Wie von den Autoren vorgeschlagen, setzen wir

$$F(A_1, t) := \frac{1}{t} \sum_{j=1}^{\infty} (t^{-1} A_1)^j$$

und erhalten mit Hilfe des Korollars 3.7

$$(tI_m - A_1)^{-1} = \frac{1}{t} I_m + F(A_1, t). \tag{3.12}$$

Nach Konstruktion sind alle Eigenwerte von  $A_1$  beziehungsweise die Singularitäten von  $(tI_m - A_1)^{-1}$  im Inneren des von  $\Gamma$  umschlossenen Gebiets. Auf der linken Seite von Gleichung (3.11) ändert sich der Wert des Integrals nicht, wenn wir anstelle von  $\Gamma$  über eine andere Jordan-Kurve, die das von  $\Gamma$  umschlossene Gebiet enthält, integrieren. Daher können wir als Integrationskontur  $\hat{\Gamma}$  ebenso gut den Kreis um den Nullpunkt mit hinreichend großem Radius  $r > \|A_1\|_2$  wählen.

Für  $t \in [0, 2\pi]$  ist die Abbildung  $t \mapsto re^{it}$  eine Parametrisierung von  $\widehat{\Gamma}$ . Diese Kurve ist dann positiv orientiert und wir erhalten wegen

$$\int_{\widehat{\Gamma}} \frac{1}{t} dt = \int_0^{2\pi} \frac{1}{re^{it}} i r e^{it} dt = 2\pi i$$

und unter Ausnutzung von Gleichung (3.12)

$$\int_{\Gamma} (tI_m - A_1)^{-1} dt = \int_{\widehat{\Gamma}} \frac{1}{t} I_m dt + \int_{\widehat{\Gamma}} F(A_1, t) dt = 2\pi i \cdot I_m + \int_{\widehat{\Gamma}} F(A_1, t) dt.$$

Betrachten wir nun für  $t \in \widehat{\Gamma}$  den Ausdruck  $F(A_1, t)$  etwas genauer. Es gilt

$$\|F(A_1, t)\|_2 = \left\| \frac{1}{t} \sum_{j=1}^{\infty} \left( \frac{1}{t} A_1 \right)^j \right\|_2 \leq \frac{1}{|t|} \sum_{j=1}^{\infty} \left( \frac{\|A_1\|_2}{|t|} \right)^j = \frac{\|A_1\|_2}{|t|^2} \sum_{j=0}^{\infty} \left( \frac{\|A_1\|_2}{|t|} \right)^j.$$

Nach Konstruktion gilt  $\|A_1\|_2 < r \leq |t|$ . Mit der Kenntnis über den Grenzwert der geometrischen Reihe erhalten wir daher die Abschätzung

$$\|F(A_1, t)\|_2 \leq \frac{\|A_1\|_2}{r} \frac{1}{1 - \|A_1\|_2/r}.$$

Da diese Abschätzung für alle  $r > \|A_1\|_2$  gilt und die rechte Seite für  $r \rightarrow \infty$  den Wert Null annimmt, muss daher

$$\frac{1}{2\pi i} \int_{\Gamma} (tI_m - A_1)^{-1} dt = \frac{1}{2\pi i} \left( 2\pi i \cdot I_m + \int_{\widehat{\Gamma}} F(A_1, t) dt \right) = I_m$$

und somit auch Gleichung (3.9) gelten.

Welche Schlussfolgerungen lassen sich hieraus für ein Eigenwertproblem  $(A, B)$  ziehen, bei der eine beliebige Matrix  $B$  vorgegeben ist? Diese Frage soll zumindest für HPD-Eigenwertprobleme beantwortet werden.

Sei also  $(A, B)$  ein HPD-Eigenwertproblem. Dann ist  $B$  invertierbar und jede Singularität der Greenfunktion  $G_1(t) := (tI_n - B^{-1}A)^{-1}$  ist auch eine Singularität der Funktion  $G_2(t) := (tB - A)^{-1}$ . Ist wie eben  $\Lambda_1$  eine Teilmenge des Spektrums,  $\Gamma$  eine umschließende Jordan-Kurve und die Zerlegung

$$B^{-1}A = S \begin{bmatrix} B_1^{-1}A_1 & 0 \\ 0 & B_2^{-1}A_2 \end{bmatrix} S^{-1}$$

analog zu Gleichung 3.7 gegeben, so schließt man auf ähnliche Weise

$$\frac{1}{2\pi i} \int_{\Gamma} (tI_n - B^{-1}A)^{-1} dt = \frac{1}{2\pi i} \int_{\Gamma} (tB - A)^{-1} dt = P_1$$

für eine entsprechende Matrix  $P_1$ .

### 3.3 Illustration

Bevor wir uns vom laufenden Kapitel verabschieden, soll die Theorie der vorangegangenen Abschnitte an einem konkreten Beispiel vorgeführt werden. Dabei werden wir mit Hilfe der Konturintegration zunächst den Spektralprojektor berechnen. Ist dies erledigt, reduzieren wir wie im Rayleigh-Ritz Verfahren die Dimension des Eigenwertproblems und konstruieren ausgehend vom transformierten Problem Eigenpaare des ursprünglichen Problems.

Dafür wenden wir uns den beiden HPD-Matrizen  $A, B \in \mathbb{C}^{3,3}$  zu, welche durch

$$A := \text{diag}(3, 1, 4) \text{ und } B := \text{diag}(1, 5, 9)$$

gegeben sind. Durch einfaches Nachrechnen überprüft man, dass sich das Eigenwertproblem  $(A, B)$  durch Vektoren  $x_1 \in \text{Span}_{\mathbb{C}}\{e_1\}$  mit zugehörigem Eigenwert  $\lambda_1 = 3$ , Vektoren  $x_2 \in \text{Span}_{\mathbb{C}}\{e_2\}$  mit zugehörigem Eigenwert  $\lambda_2 = 1/5$ , sowie Vektoren  $x_3 \in \text{Span}_{\mathbb{C}}\{e_3\}$  mit zugehörigem Eigenwert  $\lambda_3 = 4/9$  lösen lässt.<sup>4</sup>

Vergessen wir für den Moment, dass uns die Eigenpaare bekannt sind und versuchen mit den Methoden, welche wir in den vorangegangenen Abschnitten besprochen haben, die Eigenwerte im Inneren des reellen Intervalls  $I = [-1, 1]$  zu bestimmen.<sup>5</sup> Gemäß dem Abschnitt 3.2 wählen wir daher als Integrationskontur  $\Gamma$  das Bild der Funktion

$$\gamma: [0, 2\pi[ \rightarrow \mathbb{C}, \theta \mapsto e^{i\theta} \quad (3.13)$$

und integrieren darüber die Green-Funktion

$$G(t) = \text{diag}(t - 3, 5t - 1, 9t - 4)^{-1} = \text{diag}\left(\frac{1}{t - 3}, \frac{1}{5t - 1}, \frac{1}{9t - 4}\right).$$

Nach Definition 3.5 ist  $\Gamma$  eine positiv orientierte Jordan-Kurve. Da nach Konstruktion keiner der Eigenwerte im Bild von  $\gamma$  liegt, ist  $G$  auf der gesamten Kurve wohldefiniert. Folgen wir also weiter dem Abschnitt 3.2, so erhalten wir

$$\frac{1}{2\pi i} \int_{\Gamma} G \, ds = \frac{1}{2\pi i} \int_0^{2\pi} G(\gamma(t)) \cdot \gamma'(t) \, dt = \text{diag}\left(0, \frac{1}{5}, \frac{1}{9}\right)$$

Diese Matrix lässt sich mit den  $B$ -orthonormalen Vektoren  $x_1 = 1/\sqrt{6} \cdot e_2$  und  $x_2 = 1/3 \cdot e_3$  wie gewünscht in der Art

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/6 & 0 \\ 0 & 0 & 1/9 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1/\sqrt{6} & 0 \\ 0 & 1/3 \end{bmatrix} \begin{bmatrix} 0 & 1/\sqrt{6} & 0 \\ 0 & 0 & 1/3 \end{bmatrix} = x_1 x_1^H + x_2 x_2^H =: X X^H$$

darstellen.

---

<sup>4</sup> Hier bezeichnen  $e_1, e_2, e_3 \in \mathbb{C}^3$  wie üblich die Einheitsvektoren.

<sup>5</sup> Man beachte, dass weder  $-1$  noch  $1$  im Spektrum von  $(A, B)$  liegen.

Nachdem dies geschafft ist, lassen wir uns von der in Abschnitt 3.1 diskutierten Vorgehensweise inspirieren und reduzieren das Problem zunächst mit Hilfe einer vollrangigen Matrix  $Z \in \mathbb{C}^{3,2}$  auf ein Problem kleinerer Dimension. Zu diesem Zwecke wählen wir

$$Z := \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

und benutzen die Matrix  $P := XX^H Z$  um das Ausgangsproblem vermöge  $\tilde{A} := P^H A P$  und  $\tilde{B} := P^H B P$  auf das Problem

$$\tilde{A}y = \begin{bmatrix} 1/36 & 0 \\ 0 & 1/9 \end{bmatrix} y = \mu \begin{bmatrix} 5/36 & 0 \\ 0 & 1/4 \end{bmatrix} y = \mu \tilde{B}y$$

der Dimension  $(2 \times 2)$  zu transformieren. Die Eigenwerte  $1/5$  und  $4/9$  lassen sich leicht ablesen und stimmen wie erwartet mit den sich auf  $[-1, 1]$  befindlichen Eigenwerten des eingangs formulierten Problems überein. Passende Eigenvektoren sind Elemente aus  $\text{Span}_{\mathbb{C}}\{[1 \ 0]^T\}$  und  $\text{Span}_{\mathbb{C}}\{[0 \ 1]^T\}$ .

Es bleibt die Rücktransformation der Eigenvektoren. Dazu wählen wir aus kosmetischen Gründen die Eigenvektoren

$$y_1 := \begin{bmatrix} \sqrt{6} \\ 0 \end{bmatrix} \text{ und } y_2 := \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

und setzen  $Y = [y_1 \ y_2]$ . Wir erhalten dann

$$PY = \begin{bmatrix} 0 & 0 \\ 1/6 & 0 \\ 0 & 1/9 \end{bmatrix} \begin{bmatrix} \sqrt{6} & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1/\sqrt{6} & 0 \\ 0 & 1/3 \end{bmatrix} = X.$$

Eine Bemerkung zum Abschluss: Erinnern wir uns an Satz 3.2, so ist es nicht verwunderlich, dass diese Rechnung exakte Eigenpaare liefert. Schließlich hat  $P$  vollen Rang und der Unterraum  $\text{Bild}(P)$  ist invariant unter  $B^{-1}A$ .



## Kapitel 4

# Hilfsmittel: Rationale Funktionen

Nachdem das vorangegangene Kapitel Ideen zum Filtern von Eigenpaaren theoretisch beleuchtet hat, werden wir uns nun mit der Frage der praktischen Umsetzbarkeit beschäftigen. Im Mittelpunkt wird dabei die Konstruktion geeigneter Suchräume stehen, welche im Rayleigh-Ritz-Verfahren zum Einsatz kommen sollen. Es wird sich zeigen, dass die Konturintegration hierbei ein nützliches Hilfsmittel darstellt.

### 4.1 Rayleigh-Ritz Iteration

Bei der Behandlung des Rayleigh-Ritz Verfahrens wurde angedeutet, dass das Einbinden einer geeigneten Iterationsvorschrift dabei helfen kann, die Güte von errechneten Ritz-Paaren zu verbessern. Hierbei ist „Güte“ natürlich in Abhängigkeit vom Kontext zu bewerten. Im Folgenden wollen wir uns genauer mit dieser ominösen Iterationsvorschrift auseinander setzen und beginnen die Herleitung mit der Betrachtung eines sehr einfach umsetzbaren Verfahrens zur Bestimmung von Eigenpaaren.

Ausgangspunkt ist ein gewöhnliches Eigenwertproblem mit einer von Null verschiedenen hermiteschen Matrix  $A \in \mathbb{C}^{n,n}$ . Bei der sogenannten *Potenzmethode* wird ausgehend von einem Startvektor  $y_{(0)} \in \mathbb{C}^n \setminus \{0\}$  in jeder Iteration der Vektor

$$y_{(k+1)} = \frac{1}{\|A^{k+1}y_{(0)}\|} A^{k+1}y_{(0)}$$

oder dazu äquivalent

$$y_{(k+1)} = \frac{1}{\|Ay_{(k)}\|} Ay_{(k)}$$

berechnet. Dieser Vorgang wird wiederholt, bis gewisse Abbruchkriterien erfüllt sind. Man kann zeigen, dass die Folge der Iterierten gegen einen zu einem betragsmäßig größten Eigenwert gehörenden Eigenvektor der Länge 1 konvergiert.<sup>1</sup>

---

<sup>1</sup> Eine exaktere Formulierung bietet Satz B.6 im Anhang B.

Da uns daran gelegen ist, nicht nur mit einzelnen Elementen des  $\mathbb{C}^n$  zu arbeiten, sondern mit Matrizen zu hantieren, müssen wir eine allgemeinere Form der Potenzmethode betrachten. Dazu wählen diesmal eine Startmatrix  $Y_{(0)} \in \mathbb{C}^{n,m}$  vollen Ranges und berechnen die  $k$ -te Iterierte mit

$$Y_{(k)} = A^k Y_{(0)}.$$

Bei der Normierung ist allerdings Vorsicht geboten: In seinen Abhandlungen über Unterraumiterationen merkt Y. Saad in [11, Abschnitt 5.1] an, dass ungeschicktes Normieren zum zunehmenden Verlust der linearen Unabhängigkeit der Spalten von  $Y_{(k)}$  führen kann.<sup>2</sup>

Anstatt jede Spalte von  $Y_{(k)}$  separat zu normieren, wird eine QR-Zerlegung<sup>3</sup> bemüht. Diese führt in der Tat zu einer Normierung: Sind  $Q := [q_i]_{i=1:n} \in \mathbb{C}^{n,n}$  und  $R \in \mathbb{C}^{n,m}$  so gewählt, dass  $Y_{(k)} = QR$  die QR-Zerlegung von  $Y_{(k)}$  ist, so gilt

$$\text{Bild}(Y_{(k)}) = \text{Bild}([q_i]_{i=1:m})$$

und  $\|q_i\|_2 = 1$  für  $i = 1 : m$ . Daher stellt folgender Algorithmus eine Verallgemeinerung der Potenzmethode dar.

---

**Algorithmus 3** Verallgemeinerte Potenzmethode (Vgl. [11, Algorithmus 5.1, S. 115])

---

**Input:** Hermitesche Matrix  $A$

**Output:** Matrix  $Y_{(k)}$  mit approximierten Eigenvektoren

- 1: Wähle linear unabhängige Vektoren  $\{y_i\}_{i=1:m}$  und setze  $Y_{(0)} \leftarrow [y_i]_{i=1:m}$  und  $k \leftarrow 1$ .
  - 2: **repeat**
  - 3:   Setze  $Y_{(k)} \leftarrow AY_{(k-1)}$  und berechne QR-Zerlegung  $Y_{(k)} = QR$ .
  - 4:   Setze  $Y_{(k)} \leftarrow Q$  und  $k \leftarrow k + 1$ .
  - 5: **until** Verfahren konvergiert.
- 

Saad weist darauf hin, dass die Kosten der Berechnung der QR-Zerlegung sehr hoch werden können. Da der von den Spalten von  $Y_{(k)}$  aufgespannte Unterraum gleich dem von den Spalten von  $A^k Y_{(0)}$  aufgespannten Unterraum ist, schlägt er daher folgende Abwandlung des eben vorgestellten Algorithmus vor.

---

**Algorithmus 4** Gebrauch variabler Exponenten (Vgl. [11, Algorithmus 5.2, S. 116])

---

**Input:** Hermitesche Matrix  $A$

**Output:** Matrix  $Y$  mit approximierten Eigenvektoren

- 1: Wähle linear unabhängige Vektoren  $\{y_i\}_{i=1:m}$ , setze  $Y \leftarrow [y_i]_{i=1:m}$  und wähle initialen Exponenten  $k$ .
  - 2: **repeat**
  - 3:   Setze  $S \leftarrow A^k Y$  und orthonormalisiere  $S$  zu  $\hat{S}$ .
  - 4:   Setze  $Y \leftarrow \hat{S}$  und wähle neuen Exponenten  $k$ .
  - 5: **until** Verfahren konvergiert.
- 

<sup>2</sup> Dies ist unmittelbar einzusehen, wenn man bedenkt, dass jede Spalte gegen einen dominanten Eigenvektor konvergiert.

<sup>3</sup> Eine Formulierung des Satzes über die Existenz der QR-Zerlegung ist im Anhang B zu finden. Siehe hierzu Satz B.7. Für weitere Ausführungen siehe auch [14, S. 55 ff.].



Auch hier ist zu beachten, dass im Falle der Wahl eines sehr großen Exponenten die Unabhängigkeit der Spalten von  $S$  nicht mehr gewährleistet werden kann. Wir wollen an dieser Stelle auf Konvergenz- und Laufzeitanalysen der eben vorgestellten Algorithmen verzichten und kommen schließlich zur vielfach angekündigten Iterationsvorschrift für das Rayleigh-Ritz Verfahren, welche sich aus der Potenzmethode ableitet.

---

**Algorithmus 5** Iteratives Rayleigh-Ritz Verfahren (Vgl. [11, Algorithmus 5.3, S. 118])

---

**Input:** Hermitesche Matrix  $A$

**Output:** Matrix  $Y$  mit approximierten Eigenvektoren

- 1: Wähle linear unabhängige Vektoren  $\{y_i\}_{i=1:m}$ , setze  $Y \leftarrow [y_i]_{i=1:m}$  und wähle initialen Exponenten  $k$ .
  - 2: **repeat**
  - 3:   Setze  $S \leftarrow A^k Y$ .
  - 4:   Orthonormalisiere die Spalten von  $S$  und setze  $\tilde{A} \leftarrow S^H A S$ .
  - 5:   Berechne Eigenvektoren  $\tilde{X} \leftarrow [\tilde{x}_i]_{i=1:m}$  von  $\tilde{A}$ .
  - 6:   Setze  $Y \leftarrow S \tilde{X}$  und wähle neuen Exponenten  $k$ .
  - 7: **until** Verfahren konvergiert.
- 

Zunächst ein Wort zur fünften Zeile. Hier wurde das Berechnen von Schurvektoren – so wie es in der oben zitierten Quelle vorgeschlagen wird – durch das Berechnen von Eigenvektoren ersetzt. Dies ist möglich, weil  $A$  nach Vereinbarung eine hermitesche Matrix ist.

Die Wurzeln des eben erarbeiteten Algorithmus sind deutlich zu erkennen. In den Zeilen vier bis sechs wird das Rayleigh-Ritz Verfahren benutzt. Anstelle von Ritz-Paaren werden allerdings lediglich Ritz-Vektoren berechnet. In jeder Iteration wird wie beim Potenzverfahren ein neuer Exponent festgelegt und somit ein neuer Suchraum  $\mathcal{S} = \text{Bild}(A^k Y)$  vorgegeben. Erläuterungen zum Konvergenzverhalten sind in [11, Abschnitt 5] zu finden. Zur Berechnung von Eigenpaaren eines HPD-Eigenwertproblems  $(A, B)$ , bei dem  $B$  nicht mehr der Identität entspricht, müssen die Zeilen vier bis sechs gemäß Algorithmus 2 angepasst werden.

Es ist möglich, den zuletzt eingeführten Algorithmus weiter abzuwandeln. Dazu betrachten wir wieder ein HPD-Eigenwertproblem  $(A, B)$ . Zu diesem Duo gesellt sich nun mit  $\rho(B^{-1}A)$  ein Polynom in  $B^{-1}A$ , welches wir benutzen, um das iterative Rayleigh-Ritz Verfahren, wie in Algorithmus 6 formuliert, zu ändern. Diese Methode ähnelt stark dem Algorithmus 5. Die Zeilen vier bis sechs entsprechen erneut dem Rayleigh-Ritz Verfahren, aber die Berechnung des Suchraums  $\mathcal{S} := \text{Bild}(P_{(k)})$  geht anders vonstatten.

---

**Algorithmus 6** Beschleunigte Rayleigh-Ritz Iteration (Vgl. [15, Algorithmus A])

---

**Input:** HPD-Eigenwertproblem  $(A, B)$

**Output:** Matrix  $Y_{(k)}$  mit approximierten Eigenvektoren

- 1: Wähle  $m$  linear unabhängige Vektoren  $Y_{(0)} \leftarrow [y_i]_{i=1:m} \in \mathbb{C}^{n,m}$ . Setze  $k \leftarrow 1$ .
  - 2: **repeat**
  - 3:   Approximiere den Unterraumprojektor:  $P_{(k)} \leftarrow \rho(B^{-1}A)Y_{(k-1)}$
  - 4:   Reduziere die Dimension:  $\tilde{A}_{(k)} \leftarrow P_{(k)}^H A P_{(k)}$ ,  $\tilde{B}_{(k)} \leftarrow P_{(k)}^H B P_{(k)}$ .
  - 5:   Löse das transformierte Problem  $\tilde{A}_{(k)} \tilde{X}_{(k)} = \tilde{B}_{(k)} \tilde{X}_{(k)} \tilde{\Lambda}_{(k)}$  in  $\tilde{X}_{(k)}$  und  $\tilde{\Lambda}_{(k)}$ .
  - 6:   Setze  $Y_{(k)} \leftarrow P_{(k)} \tilde{X}_{(k)}$  und  $k \leftarrow k + 1$ .
  - 7: **until** Abbruchkriterium ist erfüllt.
- 

Im Kontext dieses Algorithmus wird  $\rho$  auch als *Filter*<sup>4</sup> oder *Beschleuniger* bezeichnet. Von dessen Wahl hängt nämlich ab, ob und wie gut Eigenpaare approximiert werden: Sei  $[\lambda_1, \lambda_2]$  ein reelles Intervall, in dessen Inneren  $l \in \mathbb{N}$  Eigenwerte gefunden werden können. Ist nun  $\rho(B^{-1}A)$  der Spektralprojektor,  $m = l$  und hat die Matrix  $P_{(1)} = \rho(B^{-1}A)Y_{(0)}$  vollen Rang, so konvergiert der Algorithmus 6 in einer Iteration (Vgl. [15, S. 356]).

Da der Spektralprojektor in den meisten Fällen unbekannt sein dürfte, liegt die Idee nahe, ihn zu approximieren. Tang und Polizzi [15, S. 356] merken an, dass dies gut funktioniert, falls  $\rho$  eine durch *Gauß-Legendre-Quadratur* konstruierte rationale Funktion ist. Um den Gedankengang der Autoren nachvollziehen zu können, wird sich das folgende Intermezzo mit der Auffrischung des Konzeptes von Quadraturformeln beschäftigen. Dabei sehen wir von Beweisen und ausufernden Erläuterungen ab, da diese Thematik in den meisten Einführungsbüchern zur numerischen Mathematik ausführlich besprochen wird.<sup>5</sup> Im Anschluss werden wir mit der Konstruktion des Projektors fortfahren.

## 4.2 Gauß'sche Quadratur

Um ein Integral numerisch zu approximieren, bedient man sich sogenannter *Quadraturformeln*. Dazu betrachten wir eine stetige Funktion  $f: \mathbb{R} \rightarrow \mathbb{R}$ , welche wir auf einem gegebenen Intervall  $I := [a, b] \subset \mathbb{R}$  integrieren wollen.<sup>6</sup> Zu gegebenen Stützpunkten  $(x_i, f(x_i))_{i=0:n}$  auf  $I \times \mathbb{R}$  sei  $p_n$  das zugehörige *Interpolationspolynom* vom Grad  $n$ , also ein Polynom, welches  $p_n(x_i) = f(x_i)$  für alle  $i$  mit  $0 \leq i \leq n$  erfüllt. Dann bezeichnen wir die Näherung

$$Q_n(f) := \int_a^b p_n(x) \, dx = (b-a) \sum_{i=0}^n \omega_i f(x_i) \quad (4.1)$$

als *interpolatorische Quadraturformel*. Dabei gilt

$$\omega_k = \int_0^1 \prod_{j=0, j \neq k}^n \frac{t - t_j}{t_k - t_j} \, dt, \quad t_j = \frac{x_j - a}{b - a}.$$

---

<sup>4</sup> Dieser Filter ist nicht mit dem in Definition 3.1 eingeführten Begriff zu verwechseln.

<sup>5</sup> Siehe zum Beispiel [9, Abschnitt 6].

<sup>6</sup> Im Allgemeinen ist die Stetigkeit von  $f$  nicht zwingend erforderlich. Dennoch werden wir uns hier der Einfachheit halber auf stetige Funktionen einschränken.

Die Qualität der Approximation, also die Abweichung vom exakten Integral, hängt maßgeblich von der Wahl und Anzahl der Stützpunkte ab. Wollten wir beispielsweise das Integral einer konstanten Funktion berechnen, so erschiene es wenig plausibel, anstelle der direkten Berechnung ein Polynom vom Grad 69 auf 70 Stützstellen für die Approximation zu bemühen.

Bei der Anwendung von Gauß-Legendre-Quadraturen ergibt sich die Wahl der Stützpunkte durch die Berechnung von Nullstellen von Polynomen, die in einer Orthogonalitätsbeziehung zueinander stehen. Wir werden in Kürze sauber formulieren, wie dies zu verstehen ist.

Ausgangspunkt für die Integration ist nun eine stetige Funktion  $f$ , die eine Faktorisierung in zwei stetige Funktionen  $\omega$  und  $g$  der Art

$$f = \omega \cdot g$$

besitzt, wobei  $\omega$  auf dem Integrationsintervall  $[a, b]$  positiv sein soll. Die Funktion  $\omega$  wird auch als *Gewichtsfunktion* bezeichnet. Ziel ist also nun die Berechnung von

$$\int_a^b \omega(x)g(x) \, dx, \quad (4.2)$$

wobei wir zusätzlich fordern, dass (4.1) und (4.2) für alle Polynome bis zum Grad  $(2n-1)$  übereinstimmen.

Dazu betrachten wir die Monombasis  $\{x^k\}_{k=0:(2n-1)}$  auf dem Raum der Polynome vom Grad  $(2n-1)$ . Dann landen wir unweigerlich bei dem Gleichungssystem

$$\sum_{j=1}^n \omega_j x_j^k = \int_a^b \omega(x) \cdot x^k \, dx \text{ mit } k = 0 : 2n-1.$$

Man kann zeigen, dass die Lösung dieses Systems durch Nullstellen eines Polynoms gegeben ist, welches durch ein Gram-Schmidt-Orthogonalisierungsverfahren bezüglich des Skalarproduktes

$$\langle p, q \rangle_\omega := \int_a^b p(x)q(x)\omega(x) \, dx$$

konstruiert wurde. Das heißt konkret: Ausgehend vom Polynom  $p_0 \equiv 1$  ist

$$p_n(x) := x^n - \sum_{j=0}^{n-1} \frac{\langle x^n, p_j \rangle_\omega}{\langle p_j, p_j \rangle_\omega} p_j(x)$$

gerade dasjenige Polynom  $n$ -ten Grades, durch dessen Nullstellen das obige Gleichungssystem gelöst wird.

Sind nun  $\{x_j\}_{j=1:n}$  die Nullstellen dieses  $n$ -ten Orthogonalitätspolynoms, so heißt die numerische Integrationsformel

$$Q_n(f) = \sum_{j=1}^n \omega_j f(x_j) \text{ mit } \omega_j = \langle L_j, 1 \rangle_\omega = \int_a^b L_j(x) \rho(x) dx$$

Gauß'sche Quadraturformel der  $n$ -ten Ordnung. Dabei ist  $L_j(x)$  durch

$$L_j(x) = \prod_{k \neq j, k=1}^n \frac{x - x_k}{x_j - x_k}$$

gegeben. **NOCHMAL NACHLESEN. IRGENDWAS STIMMT HIER NICHT.**

### 4.3 Approximation des Spektralprojektors

Sei  $(A, B)$  ein HPD-Eigenwertproblem und  $I := [\lambda_1, \lambda_2]$  ein reelles Intervall. Sei weiter  $\mathcal{X} \subseteq \mathbb{C}^n$  derjenige Unterraum, welcher von den Eigenvektoren aufgespannt wird, die zu den im Inneren von  $I$  befindlichen Eigenwerten korrespondieren. Wir wollen in diesem Abschnitt den Spektralprojektor approximieren, welcher den Vektorraum  $\mathbb{C}^n$  auf  $\mathcal{X}$  projiziert.

Die erste Ingredienz, die zum Gelingen dieser Approximation beiträgt, ist eine rationale Funktion  $\varrho: \mathbb{C} \rightarrow \mathbb{C}$  mit  $\varrho(\mathbb{R}) \subseteq \mathbb{R}$ , welche auf  $I$  näherungsweise der konstanten 1-Funktion entspricht, also  $\varrho(I) \subseteq ]1 - \varepsilon, 1 + \varepsilon[$  für ein  $\varepsilon > 0$ . Dafür bemühen wir die Cauchy'sche Integraldarstellung der Indikatorfunktion und wandeln diese mit Hilfe numerischer Quadraturformeln in die gewünschte rationale Funktion  $\varrho$  um.

Zunächst zur Indikatorfunktion: Ist  $c$  der Mittelpunkt des Intervalls  $I$  und  $r$  der Abstand des Mittelpunktes zum Rand des Intervalls, dann entspricht die Menge

$$\mathcal{C} := \{z \in \mathbb{C} : |z - c| = r\}$$

der Kugel mit Radius  $r$  um  $c$ . Mit dem Cauchy'schen Integralsatz lässt sich zeigen, dass im Falle  $z \notin \mathcal{C}$

$$\frac{1}{2\pi i} \int_{\mathcal{C}} \frac{1}{t - z} dt = \begin{cases} 1 & \text{falls } |z - c| < r \\ 0 & \text{falls } |z - c| > r \end{cases}$$

gilt. Um dieses Integral mit einer Gauß'schen Quadraturformel annähern zu können, benötigen wir eine Parametrisierung von  $\mathcal{C}$ . Wie in [15] vorgeschlagen wählen wir die Funktion

$$\gamma: [-1, 3] \rightarrow \mathbb{C}, x \mapsto c + r e^{i \frac{\pi}{2}(1+x)}$$

als Parametrisierung.

Wir erhalten dann für alle  $z \notin \mathcal{C}$  die Gleichung

$$\begin{aligned} \frac{1}{2\pi\iota} \int_{\mathcal{C}} \frac{1}{t-z} dt &= \frac{1}{2\pi\iota} \int_{-1}^3 \frac{\gamma'(x)}{\gamma(x)-z} dx \\ &= \frac{1}{2\pi\iota} \left( \int_{-1}^1 \frac{\gamma'(x)}{\gamma(x)-z} dx + \int_1^3 \frac{\gamma'(x)}{\gamma(x)-z} dx \right) \\ &= \frac{1}{2\pi\iota} \left( \int_{-1}^1 \frac{\gamma'(x)}{\gamma(x)-z} dx + \int_{-1}^1 \frac{\gamma'(2-x)}{\gamma(2-x)-z} dx \right) \\ &= \frac{1}{2\pi\iota} \int_{-1}^1 \left( \frac{\gamma'(x)}{\gamma(x)-z} + \frac{\overline{\gamma'(x)}}{\overline{\gamma(x)}-z} \right) dx \end{aligned}$$

wobei  $\overline{\gamma(x)}$  und  $\overline{\gamma'(x)}$  die komplexen Konjugationen von  $\gamma(x)$  beziehungsweise  $\gamma'(x)$  bezeichnen.

Es seien nun  $(\omega_j, x_j)_{j=1:m}$  die für die Gauß'sche Quadraturformel benötigten Gewichte und Stützstellen. Dann setzen wir

$$\rho(z) := \frac{1}{2\pi\iota} \sum_{j=1}^m \left( \frac{\omega_j \cdot \gamma'(x_j)}{\gamma(x_j)-z} - \frac{\omega_j \cdot \overline{\gamma'(x_j)}}{\overline{\gamma(x_j)}-z} \right)$$

und erhalten nach der Substitution  $\gamma(x_j) := \gamma_j$  und  $\sigma_j := \omega_j \gamma'(x_j)/(2\pi\iota)$  die gewünschte rationale Funktion

$$\varrho: \mathbb{C} \rightarrow \mathbb{C}, z \mapsto \sum_{j=1}^m \left( \frac{\sigma_j}{\gamma_j - z} + \frac{\overline{\sigma_j}}{\overline{\gamma_j} - z} \right)$$

als Approximation der Indikatorfunktion. Hierbei ist bemerkenswert, dass die rationale Funktion bereits in Partialbruchzerlegung vorliegt. Setzen wir schließlich  $B^{-1}A$  in die rationale Funktion ein, so erhalten wir

$$\begin{aligned} \varrho(B^{-1}A) &= \sum_{j=1}^m \sigma_j (\gamma_j I - B^{-1}A)^{-1} + \sum_{j=1}^m \overline{\sigma_j} (\overline{\gamma_j} I - B^{-1}A)^{-1} \\ &= \sum_{j=1}^m \sigma_j (\gamma_j B - A)^{-1} B + \sum_{j=1}^m \overline{\sigma_j} (\overline{\gamma_j} B - A)^{-1} B \end{aligned}$$

und folglich

$$\varrho(B^{-1}A)Y = \sum_{j=1}^m \sigma_j (\gamma_j B - A)^{-1} B Y + \sum_{j=1}^m \overline{\sigma_j} (\overline{\gamma_j} B - A)^{-1} B Y$$

für eine Matrix  $Y \in \mathbb{C}^{n,m}$ . Sind  $A, B$  und  $Y$  reellwertig, so lässt sich dies zu

$$\varrho(B^{-1}A)Y = 2 \sum_{j=1}^m \Re \left( \sigma_j (\gamma_j B - A)^{-1} B Y \right) \quad (4.3)$$

vereinfachen. Wie man diese rationale Funktion konkret in einem Algorithmus verwenden kann, besprechen wir im anschließenden Abschnitt.

## 4.4 FEAST-Algorithmus

Im Jahr 2009 stellte E. Polizzi in [10] ein Verfahren vor, welches sich die Theorie der vorigen Abschnitte zu Nutze macht. Dieser „[...] *fast and stable algorithm for solving the symmetric eigenvalue problem* [...]“ [10, Abstract] wurde seither weiterentwickelt und zahlreichen Analysen unterzogen.<sup>7</sup>

Polizzi beginnt seine Ausführungen mit einem Blick in die Physik und motiviert das Lösen von Eigenwertproblemen anhand der Schrödinger-Eigenwertgleichung

$$\mathbf{H}\Psi = E\mathbf{S}\Psi$$

welche die Fragestellung modelliert, ob gewisse Quantenobjekte, die von einer Wellenfunktion  $\Psi$  beschrieben werden, kinetische Energie besitzen oder nicht. Hierbei ist  $\mathbf{H}$  eine hermitesche Matrix und  $\mathbf{S}$  eine symmetrisch positiv definite Matrix.

Der Autor merkt an, dass das Lösen solcher Systeme, insbesondere dann, wenn sie sehr groß sind, enorme Anforderungen an die Numeriker stellt. Die Frage ist stets, wie Eigenpaare effizient berechnet werden können und welche Genauigkeit man erwarten darf. Polizzi bewirbt seinen Algorithmus mit hoher Geschwindigkeit, Robustheit und guter Skalierbarkeit.

Ausgehend von einer Ausführung über das Konzept der Konturintegration zur Bestimmung von Eigenpaaren, baut Polizzi seinen Algorithmus auf, welcher auf  $n$ -dimensionale Eigenwertprobleme  $(A, B)$  mit hermiteschem oder reell symmetrischem  $A$  und symmetrisch positiv definitem  $B$  ausgelegt ist und im Inneren eines reellen Intervalls  $[\lambda_1, \lambda_2]$  die  $m \in \mathbb{N}$  Eigenpaare finden soll. Sind  $(\omega_j, x_j)_{j=1:k}$  die für die Gauß'sche Quadraturformel benötigten Gewichte und Stützstellen, so lässt sich der FEAST-Algorithmus wie in Algorithmus 7 notieren.

Dieser Algorithmus stellt eine Umsetzung der beschleunigten Rayleigh-Ritz Iteration dar. Die Zeilen drei bis acht erfüllen den Zweck des Berechnens von  $\rho(B^{-1}A)Y$ , wobei man sich hier einer ähnlichen Umformulierung bedient wie in Gleichung (4.3).<sup>8</sup>

Es ist zu beachten, dass sich hier durch die Verwendung einer negativ orientierten Kurve, was aus der vierten und fünften Zeile herauszulesen ist, Vorzeichenwechsel ergeben. Da bei einer entsprechenden Parametrisierung auf der rechten Seite der Identität (3.9) der zusätzliche Faktor  $-1$  benötigt wird, muss dies entsprechend bei der Konstruktion der rationalen Funktion  $\rho$  berücksichtigt werden.

---

<sup>7</sup> Siehe etwa [5],[6] und [15].

<sup>8</sup> Die entsprechende Anpassung von Algorithmus 6 ist in [15, S. 365] zu finden.

**Algorithmus 7** FEAST-Algorithmus (Vgl. [10, Abschnitt III])

**Input:** Eigenwertproblem  $(A, B)$  wie oben beschrieben, Intervall  $[\lambda_1, \lambda_2]$ , Gewichte und Stützstellen  $(\omega_j, x_j)_{j=1:k}$  **Output:** Approximierte Eigenpaare  $(\tilde{\lambda}_i, \xi_i)_{i=1:m}$

- 1: Wähle  $M > m$  linear unabhängige Vektoren  $Y \leftarrow [y_i]_{i=1:M} \in \mathbb{C}^{n,M}$ , setze  $Q \leftarrow 0_{n,M}$  und  $r \leftarrow (\lambda_1 - \lambda_2)/2$ .
- 2: **repeat**
- 3:   **for**  $j=1:k$
- 4:     Berechne  $\theta_{(j)} \leftarrow -(\pi/2)(x_j - 1)$ .
- 5:     Berechne  $t_{(j)} \leftarrow (\lambda_2 + \lambda_1)/2 + r e^{i\theta_{(j)}}$ .
- 6:     Löse  $(t_{(j)}B - A)Q_{(j)} = Y$  in  $Q_{(j)}$ .
- 7:     Setze  $Q \leftarrow Q - (\omega_j/2) \cdot \Re(r e^{i\theta_{(j)}} Q_{(j)})$ .
- 8:   **endfor**
- 9:   Reduziere die Dimension:  $\tilde{A} \leftarrow Q^H A Q$ ,  $\tilde{B} \leftarrow Q^H B Q$ .
- 10:   Löse das transformierte Problem  $\tilde{A}\tilde{X} = \tilde{B}\tilde{X}\tilde{\Lambda}$  in  $\tilde{X}$  und  $\tilde{\Lambda} = \text{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_M)$ .
- 11:   Setze  $X \leftarrow [\xi_i]_{i=1:M} = Q\tilde{X}$ .
- 12:   Gilt  $\tilde{\lambda}_i \in [\lambda_1, \lambda_2]$ , so gib Eigenpaar  $(\tilde{\lambda}_i, \xi_i)$  aus.
- 13: **until** Abbruchkriterium ist erfüllt.

Die weiteren Abschnitte des Papers beschäftigen sich vornehmlich mit numerischen Experimenten. Polizzi demonstriert anhand elektronenstrukturell bezogenen Berechnungen auf Kohlenstoffnanoröhren die numerische Stabilität, Robustheit und Skalierbarkeit des FEAST-Algorithmus. Dabei vergleicht er FEAST mit ARPACK – einer Bibliothek auf FORTAN77 basierenden Methoden zum Lösen hochdimensionaler Eigenwertprobleme.

Abschließend hebt der Autor einige Merkmale seines Algorithmus heraus, von denen an dieser Stelle eine bescheidene Auswahl vorgestellt wird. Zunächst einmal kommt FEAST gänzlich ohne Orthogonalisierungsmethoden aus, wie es etwa bei der unbeschleunigten Rayleigh-Ritz Iteration nötig ist. Des Weiteren weist Polizzi auf das Potential bezüglich paralleler Implementierbarkeit hin.

In Papern aus den vergangenen zwei Jahren, wie etwa in [6, Abschnitt 4.1], wird auf dieses Potential genauer eingegangen. Inzwischen wurden auch Versuche unternommen, FEAST auf nicht hermitesche Eigenwertprobleme zu verallgemeinern.<sup>9</sup>

<sup>9</sup> Siehe [6].





## Kapitel 5

# Numerische Experimente

Nachdem wir uns der Thematik des Filterns von verschiedenen Seiten genähert haben, ist es nun an der Zeit, die Kompetenz der besprochenen Methoden auf die Probe zu stellen. Dabei wollen wir Kriterien wie benötigte Rechenzeit, Konsistenz und Genauigkeit am Beispiel von Eigenwertproblemen unterschiedlicher Dimension berücksichtigen.

Alle verwendeten Algorithmen wurden in MATLAB (Version 2016b) implementiert und können bei Bedarf im Anhang C nachgeschlagen werden. Entsprechende Verweise sind im Fließtext zu finden. Bei der Durchführung der Experimente wurde eine Rechenmaschine mit folgenden Spezifikationen verwendet.

Bauart	Laptop
Hersteller	Packard Bell
Modell	EasyNote TSX66-HR-196GE
Betriebssystem	Microsoft Windows 7 Professional 64 Bit
Prozessor	Intel Core i5-2410M CPU, 2.30 GHz
Hauptspeicher	1000 GB HDD
Festplatte	IDE ATA/ATAPI-Controller
Grafikkarte	NVIDIA GeForce GT 540M, 2 GB VRAM

Zu Beginn wollen wir eine kurze Laufzeitanalyse der build-in Funktion `eig` von MATLAB durchführen und im Anschluss den FEAST-Algorithmus unter die Lupe nehmen. Dieser bietet sich an, da er sowohl das Rayleigh-Ritz Verfahren als auch die Konturintegration benutzt. Dabei werden wir als Hilfsmittel die von S. Güttel und M. Berfjala entwickelte *RKToolbox* [13] verwenden.

## 5.1 Qualitätstests

Im Verlauf der Arbeit wurde mehrfach behauptet, dass das Berechnen von Eigenpaaren bei hinreichend großer Dimension sehr zeitaufwendig werden kann. Um dies zu illustrieren, bemühen wir den Algorithmus C.1. Dieser berechnet für zufallsgenerierte Matrizen mit in Zehnerschritten aufsteigender Dimension – von 10 beginnend und bei 1500 endend – die Eigenpaare und misst die benötigte Rechenzeit.

Als Routine zur Berechnung der Eigenpaare wird die `eig`-Funktion von MATLAB verwendet. Diese gibt durch den Aufruf

$$[X, D] = \text{eig}(A)$$

eine Matrix  $X$  mit Eigenvektoren und eine Diagonalmatrix  $D$  mit den korrespondierenden Eigenwerten von  $A$  aus.<sup>1</sup> Zwischen den Dimensionen ist der Plot linear interpoliert. Um eine gewisse Konsistenz zu gewährleisten, durchläuft der Algorithmus eine Schleife, welche die eben beschriebene Berechnung drei Mal durchführt. Die folgende Grafik zeigt das Ergebnis der Berechnungen. Dabei ist jede Iteration mit einer eigenen Farbe gekennzeichnet.

Es sei noch darauf hingewiesen, dass die folgenden Diagramme aufgrund der geringen Größe der Stichproben in keinsten Weise repräsentativ sind. Sie dienen lediglich der Veranschaulichung und vermitteln höchstens eine Idee von Laufzeit- und Konvergenzverhalten. Wir werden die geplotteten Grafiken benutzen, um bekannte numerische Resultate sichtbar zu machen.

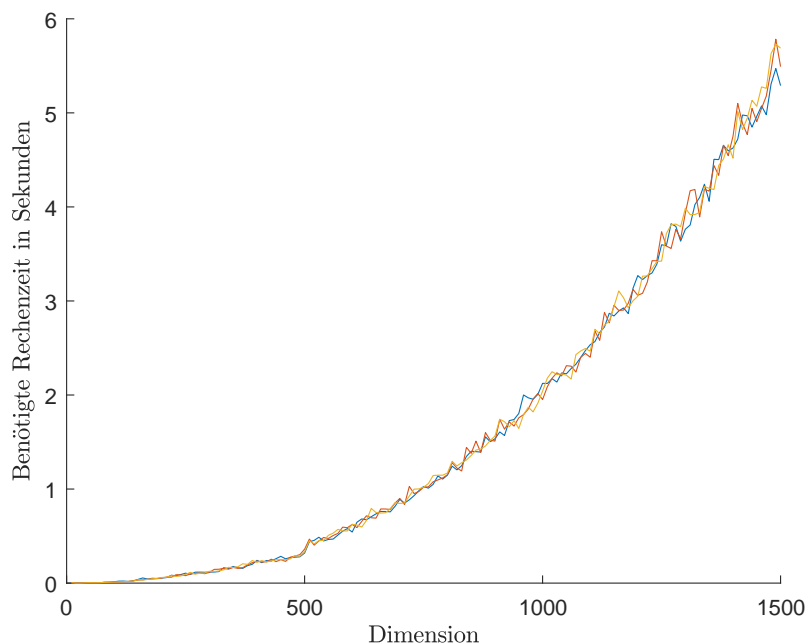


Abbildung 5.1: Laufzeitplot bei der Berechnung des Spektrums.

<sup>1</sup> Mit `eig(A, B)` lässt sich eine analoge Berechnung für beliebige Eigenwertprobleme  $(A, B)$  durchführen. Wir verweilen binnen dieses Abschnittes jedoch durchweg beim gewöhnlichen Eigenwertproblem  $(A, I)$ .

Die Abbildung deutet quadratisches Wachstum der Laufzeit in Abhängigkeit von der gewählten Dimension an. Erhöhen wir die Dimension noch weiter, so ergibt sich folgende Tabelle.

Dimension	2000	3000	4000	5000	10000
Rechenzeit in s	$\approx 12.2$	$\approx 33.3$	$\approx 75.6$	$\approx 133.1$	$\approx 1047.5$

Diese Werte vermitteln eine Intuition, von der Größenordnung von Laufzeiten bei hochdimensionalen Eigenwertproblemen, wie sie etwa bei Berechnungen von Netflix, Facebook oder Amazon vorkommen. Hier sprechen wir von Dimensionen im Millionenbereich.<sup>2</sup>

Die obigen Zahlen motivieren dazu einen Weg zu finden, beim Filtern von Eigenpaaren von einer Berechnung des gesamten Spektrums abzusehen. Wollen wir Eigenpaare durch Ritz-Paare zu einem gegebenen Unterraum annähern, so zeigt sich, dass das Rayleigh-Ritz Verfahren zu einer kürzeren Laufzeit führt. Das ist freilich nicht verwunderlich, da das Problem auf eine niedrigere Dimension reduziert wird. Der Algorithmus C.2 zeichnet entsprechend folgendes Bild bei der Berechnung von Ritz-Paaren.

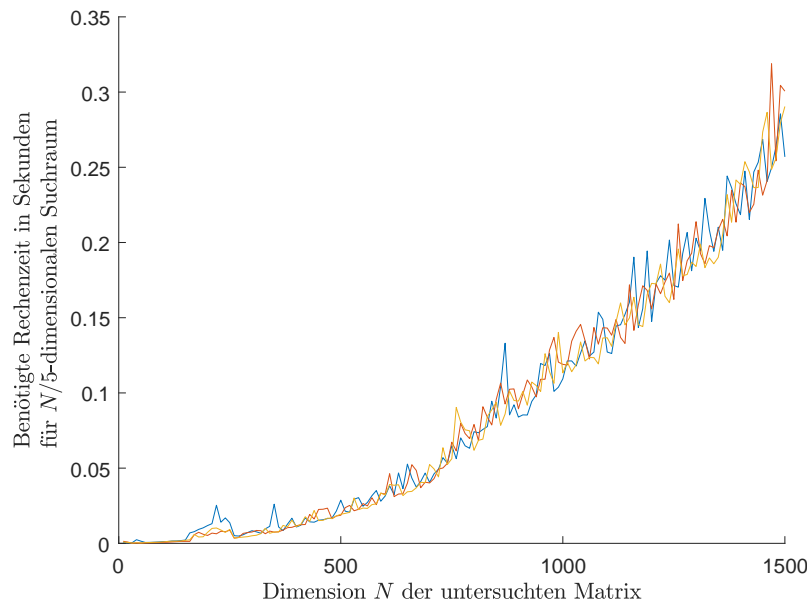


Abbildung 5.2: Laufzeitplot bei der Berechnung von Ritzpaaren.

Die Dimension des verwendeten Suchraums entsprach bei der Ausführung des Algorithmus' einem Fünftel der Dimension der untersuchten Matrix. Die Rechenzeit liegt für alle hier betrachteten  $N$  deutlich unterhalb der Sekundenmarke.

Es gibt jedoch die Kehrseite der Medaille: Die Willkürlichkeit bei der Wahl der Suchräume spiegelt sich in der Güte der Ritz-Paare wieder. Um dies einzusehen wählen wir als Gütekriterium den *minimalen kanonischen Winkel* zwischen dem von den Ritz-Vektoren aufgespannten Unterraum  $\mathcal{R}$  und dem von den Eigenvektoren aufgespannten Unterraum  $\mathcal{X}$ .

<sup>2</sup> Siehe [1], [2] und [12].

Diesen minimalen kanonischen Winkel definiert man folgendermaßen:

$$\theta_{\min}(\mathcal{R}, \mathcal{X}) = \min \left\{ \arccos \frac{|x^H r|}{\|x\| \|r\|} : r \in \mathcal{R} \setminus \{0\}, x \in \mathcal{X} \setminus \{0\} \right\}.$$

Sind  $R$  und  $X$  Matrizen mit  $\text{Bild}(R) = \mathcal{R}$  und  $\text{Bild}(X) = \mathcal{X}$ , so lässt sich  $\theta_{\min}(\mathcal{R}, \mathcal{X})$  mit Hilfe der *MATLAB*-Funktion

`subspace(R, X)`

näherungsweise bestimmen. Dies machen wir uns in Algorithmus C.3 zu Nutze. Diese Methode berechnet den kleinsten Winkel zwischen einem zweidimensionalen Suchraum und allen zweidimensionalen Unterräumen, die sich aus Eigenvektoren konstruieren lassen. Erneut wird die Berechnung dreimal durchgeführt, um ein wenig Konsistenz zu erhalten. Speziell bedient sich die Methode symmetrischer Matrizen.

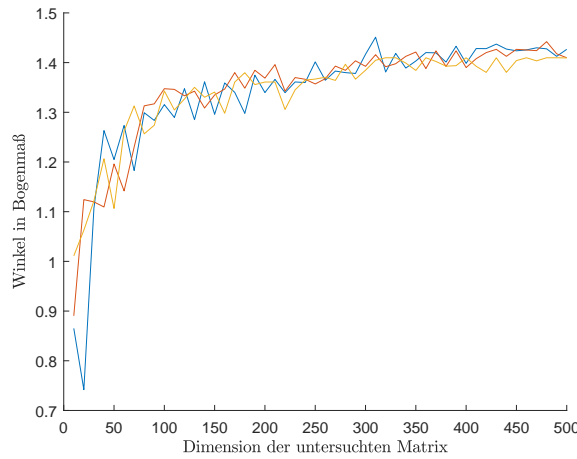


Abbildung 5.3: Berechnung des Winkels zwischen Ritz- und Eigenvektoren.

Nach Konstruktion gilt  $\theta_{\min}(\mathcal{R}, \mathcal{X}) \in [0, \pi/2]$ . Demnach lässt Abbildung 5.3 vermuten, dass diese Methode für gewisse Größenordnungen vollkommen unbrauchbar ist: Mit größer werdender Dimension nähert sich der Winkel  $\pi/2$  an. Hingegen liefert die iterative Version des Rayleigh-Ritz Verfahrens folgenden Plot.

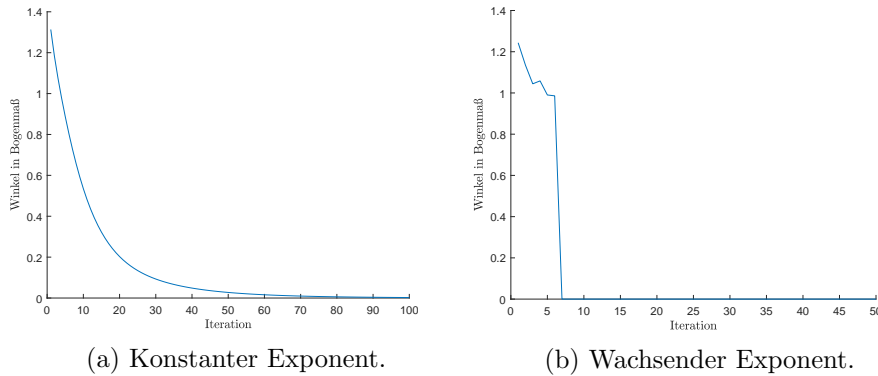


Abbildung 5.4: Unterraumwinkel bei iterativem Rayleigh-Ritz Verfahren.

Ausgehend von einer zufallsgenerierten Matrix, welche ebenfalls symmetrisch ist, wurde die Methode zunächst mit konstant bleibendem Exponenten  $k = 1$  und im Anschluss mit sich stets um 1 erhöhendem Exponenten durchgeführt. Im Gegensatz zur Prozedur ohne Iteration ist hier zumindest in dieser kleinen getesteten Stichprobe eine Konvergenz zu erkennen. Dies ist nicht weiter verwunderlich, da die Verwandtschaft zur Potenzmethode dazu führt, dass der zweidimensionale Suchraum gegen den zweidimensionalen Unterraum konvergiert, welcher von den zwei dominierenden Eigenvektoren aufgespannt wird.

Zum Abschluss wollen wir unsere Filterkriterien etwas konkreter angeben. Erscheint es bisher so, als hätte man nicht so recht in der Hand, welche Unterräume approximiert werden, wollen wir nun gezielt Eigenpaare gewisser Art aus dem Spektrum extrahieren. Dazu betrachten wir abschließend den FEAST-Algorithmus.

## 5.2 RKToolbox

Wir schränken uns nun nicht mehr auf das gewöhnliche Eigenwertproblem ein, sondern betrachten stattdessen allgemeiner das HPD-Eigenwertproblem  $(A, B)$  der Dimension  $n$ . Gesucht sind Eigenpaare, bei denen die Eigenwerte innerhalb eines gegebenen Intervalls  $[\lambda_{\min}, \lambda_{\max}]$  liegen sollen.

Wir wollen den FEAST-Algorithmus wie in Algorithmus 6 formuliert implementieren. Zur Erinnerung: Ersetzt man  $\rho(B^{-1}A)$  durch eine geeignete rationale Funktion, so entspricht der eben referenzierte Algorithmus dem Algorithmus 7.

Zur Konstruktion dieser Funktion bedienen wir uns an der von S. Güttel und M. Berfajala entwickelten *RKToolbox* [13]. Diese ermöglicht uns das Erstellen eines sogenannten `rkfun`-Objektes, mit dessen Hilfe diverse Filterfunktionen konstruiert werden können. Da so erstellte Filterfunktionen standardmäßig die Indikatorfunktion auf dem Intervall  $[-1, 1]$  approximieren, muss dem Kontext entsprechend eine passende Transformation angewendet werden.

Mit der *RKToolbox* können rationale Filterfunktionen `r` durch die Syntax

$$r = \text{rkfun}(\text{'step'}, k)$$

initialisiert werden. Dabei steuert der Parameter `'step'` welcher Funktionstyp zur Approximation benutzt werden soll und der Parameter  $k \in \mathbb{N}$  steuert den Ausschlag der Funktionen.<sup>3</sup>

---

<sup>3</sup> Für genauere Ausführungen wende sich der Leser an die Dokumentation der *RKToolbox* [13].

Zu Demonstrationszwecken setzen wir

```
r4 = rkfun('step',4), r5 = rkfun('step',5), r6 = rkfun('step',6).
```

Wie im folgenden Plot zu erkennen ist, approximieren die Funktionen die Indikatorfunktion auf  $[-1, 1]$  zumindest nach optischen Kriterien recht gut.

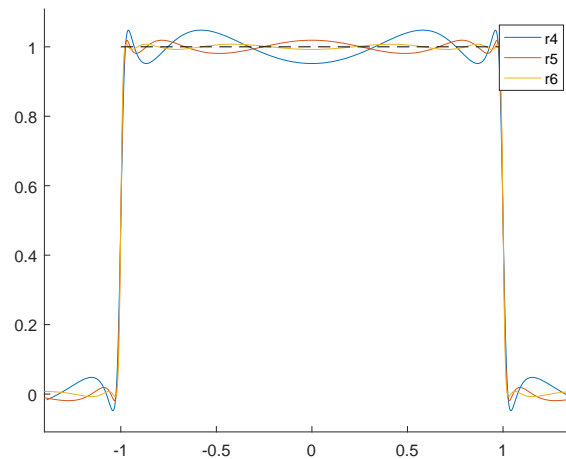


Abbildung 5.5: Plot der Filterfunktionen  $r_4$ ,  $r_5$  und  $r_6$ .

Das Einbinden dieser Filterfunktion in die beschleunigte Rayleigh-Ritz Iteration führt zu der in Algorithmus C.5 vorgestellten Umsetzung des FEAST-Algorithmus mit folgendem Plot.

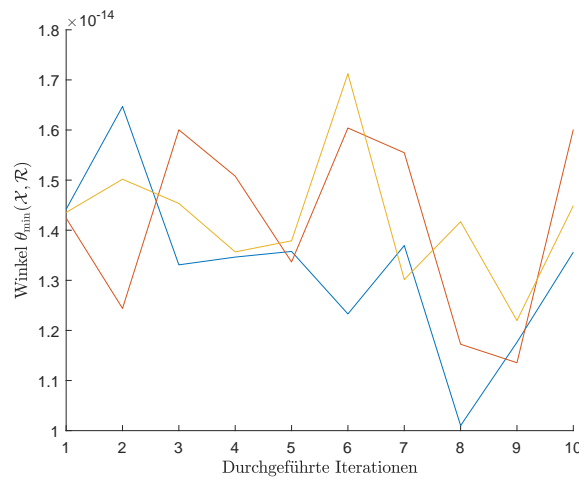


Abbildung 5.6: Plot von Algorithmus C.5.

Mit  $\mathcal{X}$  und  $\mathcal{R}$  ist hier der zum Intervall  $[1, 2.5]$  korrespondierende Eigenraum beziehungsweise der ermittelte approximierte Suchraum gemeint.

Es zeigt sich, dass bereits im ersten Durchlauf der Winkel zwischen den Unterraum verschwindend gering ist. Damit bestätigt sich die These, dass die beschleunigte Rayleigh-Ritz Iteration in einem Schritt konvergiert, falls  $\rho(B^{-1}A)$  der Spektralprojektor ist,  $m$  der Anzahl der Eigenwerte auf  $[1, 2.5]$  entspricht und  $\rho(B^{-1}A)Y_{(0)}$  vollen Rang hat – denn genau so wurden alle Werte bei der Umsetzung des FEAST-Algorithmus<sup>4</sup> gewählt.

Mit diesen Eindrücken verabschiedet sich der Verfasser der vorliegenden Arbeit von seinem Leser und bedankt sich für die Geduld und die investierte Zeit.

---

<sup>4</sup> Vgl. Seite 26 erster Absatz. Für weitere Demonstrationen siehe [16].

# Anhang A

## Symbolverzeichnis

$\mathbb{N}$	Menge der natürlichen Zahlen $\{1, 2, 3, \dots\}$
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{C}$	Menge der komplexen Zahlen
$\mathbb{R}^{m,n}$	Menge der reellen $(m \times n)$ -Matrizen
$\mathbb{C}^{m,n}$	Menge der komplexen $(m \times n)$ -Matrizen
$\mathbb{R}^n$	Elemente aus $\mathbb{R}^{n,1}$
$\mathbb{C}^n$	Elemente aus $\mathbb{C}^{n,1}$
$\text{Span}x$	Lineare Hülle $\{\lambda x \mid \lambda \in \mathbb{C}\}$ des Vektors $x$
$\delta_{i,j}$	Kronecker-Delta $\delta_{i,j} = 1$ , für $i = j$ und $\delta_{i,j} = 0$ , für $i \neq j$
$I_n$	Einheitsmatrix $[\delta_{i,j}]_{i,j=1:n}$
$0_n$	Nullmatrix des $\mathbb{C}^{n,n}$
$0_{m,n}$	Nullmatrix des $\mathbb{C}^{m,n}$
$A^H$	Komplex konjugiert Transponierte der Matrix $A$
$\gg$	Erheblich größer als
$\perp$	Steht orthogonal / unitär auf
$\perp_B$	Steht $B$ -orthogonal / $B$ -unitär auf
$\mathcal{U}^{\perp_B}$	$B$ -orthogonales Komplement des Unterraums $\mathcal{U}$
$\langle x, y \rangle_B$	Schreibweise für $x^H B y$ , für eine HPD-Matrix $B$
$\ \cdot\ _B$	Von der Matrix $B$ induzierte Norm: $\ x\ _B := \sqrt{x^H B x}$
$\ x\ _2$	Die euklidische Norm $\sqrt{x^H x}$
$\ A\ _2$	Induzierte 2-Norm für Matrizen (genauer)
$\Re$	Realteil komplexer Zahlen



# Anhang B

## Sätze

**Satz B.1** (Singulärwertzerlegung). Ist  $A \in \mathbb{C}^{m,n}$  eine  $r$ -rangige Matrix, so existieren unitäre Matrizen  $U \in \mathbb{C}^{m,m}$ ,  $V \in \mathbb{C}^{n,n}$  und eine Diagonalmatrix  $\Sigma_+^2 = \text{diag}(\sigma_1^2, \dots, \sigma_r^2) \in \mathbb{C}^{r,r}$  mit der Eigenschaft  $\sigma_1 \geq \dots \geq \sigma_r > 0$ , die eine Faktorisierung der Art

$$A = U \Sigma V^H$$

ermöglichen. Dabei ist

$$\Sigma = \begin{bmatrix} \Sigma_+ & 0_{r,n-r} \\ 0_{m-r,r} & 0_{m-r,n-r} \end{bmatrix}.$$

**Satz B.2** (Beste Rang- $k$  Approximierung). Ist  $A \in \mathbb{C}^{m,n}$  eine  $r$ -rangige Matrix und  $A = U \Sigma V^H$  mit  $U = [u_i]_{i=1:m}$  und  $V = [v_i]_{i=1:n}$  die Singulärwertzerlegung von  $A$  wie in Satz B.1, dann gilt

$$\left\| A - \sum_{i=1}^k \sigma_i u_i v_i^H \right\|_2 = \min_{\substack{B \in \mathbb{C}^{m,n} \\ \text{Rang}(B)=k}} \|A - B\|_2$$

für alle  $k = 1 : r$ .

**Satz B.3** (Spektralsatz für hermitesche Matrizen). Eine Matrix  $A \in \mathbb{C}^{n,n}$  ist genau dann hermitesch, wenn sie unitär diagonalisierbar ist und alle Eigenwerte von  $A$  reell sind.

**Satz B.4** (Existenz der Cholesky-Zerlegung). Ist  $A \in \mathbb{C}^{n,n}$  eine HPD-Matrix, so existiert eine eindeutig bestimmte untere Dreiecksmatrix  $L \in \mathbb{C}^{n,n}$  mit positiven Diagonalelementen, mit  $A = LL^H$ .

**Satz B.5** (Cauchy'scher Integralsatz). Ist  $\Omega \subseteq \mathbb{C}$  ein einfach zusammenhängendes Gebiet,  $f: \Omega \rightarrow \mathbb{C}$  eine holomorphe Funktion, sowie  $\Gamma$  eine Jordan-Kurve, so gilt

$$\oint_{\Gamma} f(z) \, dz = 0.$$

**Satz B.6** (Potenzmethode). Es sei  $A \in \mathbb{C}^{n,n}$  eine hermitesche Matrix mit Eigenpaaren  $(\lambda_i, x_i)_{i=1:n} \in \mathbb{R} \times \mathbb{C}^n \setminus \{0\}$  und  $n \geq 2$ . Es gelte weiter  $|\lambda_1| > |\lambda_j|$  für alle  $j > 1$ . Ist  $y_0 \in \mathbb{C}^n \setminus \{0\}$ , so konvergiert die Folge  $(y_k)_{k \in \mathbb{N}}$  mit

$$y_{(k+1)} = \frac{1}{\|A^{k+1}y_{(0)}\|} A^{k+1}y_{(0)}$$

gegen einen normierten Eigenvektor zum Eigenwert  $\lambda_1$

**Satz B.7** (QR-Zerlegung). Sei  $A \in \mathbb{C}^{n,m}$  mit  $n \geq m$ . Dann existiert eine unitäre Matrix  $Q \in \mathbb{C}^{n,n}$  und eine obere Dreiecksmatrix  $\tilde{R} = [r_{ij}]_{i,j=1:m} \in \mathbb{C}^{m,m}$  mit  $r_{ii} \geq 0$  für  $i = 1 : m$  mit

$$A = Q \begin{bmatrix} \tilde{R} \\ 0_{n-m,m} \end{bmatrix} =: QR.$$

# Anhang C

## Algorithmen

```
1 % Eingabe: -; Ausgabe: -
2 % Funktionsaufruf: eigenPlot
3
4 function eigenPlot
5
6 % Anpassung an Latexfont
7 set(0,'defaulttextinterpreter','latex');
8
9 % Beschriftung des Plots
10 figure; hold on;
11 xlabel('Dimension'); ylabel('Ben\{o\}tigte Rechenzeit in Sekunden');
12
13 % Initialisierung der x,y-Achsenwerte
14 xDimension = 10:10:1500; yTime = 10:10:1500;
15
16 % Beginn der Eigenpaarberechnung
17 for i=1:3
18     for N=10:10:1500 % zehnstufiger Dimensionszuwachs
19         tMatrix = tic; tEigenpairs = tic; % Timerinitialisierung
20         A = rand(N); A = A'*A; % Erzeugung einer symmetrischen Matrix
21         endMatrix = toc(tMatrix); % Dauer Matrixinitialisierung
22         [X, D] = eig(A); % Berechnung der Eigenpaare
23         endEigenpairs = toc(tEigenpairs);
24         yTime(N/10) = endEigenpairs-endMatrix; % Dauer Eigenpaarberechnung
25     end%for
26     plot(xDimension, yTime); % Erstellung des Diagramms
27 end%for
28 print -depsc eigLaufzeit; % Umwandlung in .eps Datei
29
30 end%function
```

Algorithmus C.1: Eigenpaarberechnung mit Zeiterfassung

```
1 % Eingabe: -; Ausgabe: -
2 % Funktionsaufruf: rayleighRitz
3
4 function rayleighRitz
5
6 % Anpassung an Latexfont
7 set(0,'defaulttextinterpreter','latex');
8
9 % Beschriftung des Plots
10 figure; hold on;
11 xlabel('Dimension  $N$  der untersuchten Matrix');
12 ylabel({'Ben\{"o\}tigte Rechenzeit in Sekunden'; ...
13   'f\{"u\}r  $N/5$ -dimensionalen Suchraum'});
14
15 % Initialisierung der x,y-Achsenwerte
16 xDimension = 10:10:1500; yTime = 10:10:1500;
17
18 % Beginn der Eigenpaarberechnung
19 for i=1:3
20     for N=10:10:1500 % zehnstufiger Dimensionszuwachs
21         tMatrix = tic; tRitzpairs = tic; % Timerinitialisierung
22         A = rand(N); A = A'*A; % Erzeugung einer symmetrischen Matrix
23         S = rand(N, N/5); % Suchraum der Dimension N/5
24         endMatrix = toc(tMatrix); % Dauer Matrixinitialisierung
25
26         % Rayleigh-Ritz Verfahren
27         S = orth(S); % Orthogonalisierung der Basisvektoren
28         A = S'*A*S; % Verringerung der Dimension
29         [X, D] = eig(A); S*X; % Berechnung der Ritz-Paare
30         endRitzpairs = toc(tRitzpairs);
31         yTime(N/10) = endRitzpairs-endMatrix; % Dauer Eigenpaarberechnung
32     end%for
33     plot(xDimension, yTime); % Erstellung des Diagramms
34 end%for
35 print -depsc rayleighRitzLaufzeit; % Umwandlung in .eps Datei
36
37 end%function
```

Algorithmus C.2: Rayleigh-Ritz Verfahren mit Zeiterfassung

```

1 % Eingabe: -; Ausgabe: -
2 % Funktionsaufruf: ritzPairAngle
3
4 function ritzPairAngle
5
6 % Anpassung an Latexfont
7 set(0,'defaulttextinterpreter','latex');
8
9 % Beschriftung des Plots
10 figure; hold on;
11 xlabel('Dimension der untersuchten Matrix');
12 ylabel({'Winkel in Bogenma{\ss}}');
13
14 % Initialisierung der x-Achse
15 xDimension = 10:10:500;
16
17 % Beginn der Eigenpaarberechnung
18 for i = 1:3
19     for N = 10:10:500 % zehnstufiger Dimensionszuwachs
20         %clearvars -except i N xDimension yMaxDist
21         [i, N]
22         A = rand(N); A = A'*A; % Erzeuge symmetrische Matrix
23         [X,D] = eig(A);
24         S = rand(N, 2); % Suchraum der Dimension 5
25
26         % Rayleigh-Ritz Verfahren
27         S = orth(S); % Orthogonalisierung der Basisvektoren
28         A2 = S'*A*S; % Verringerung der Dimension
29         [X2, D2] = eig(A2); X2 = S*X2; % Berechnung der Ritz-Paare
30
31         % Berechnung des minimalen Winkels
32         p = 1:N; P = nchoosek(p,2);
33         for k = 1:length(P) % = #Zeilen von P, wegen N >= 10
34             Theta(k) = subspace(X2, X(:,P(k,:)));
35         end
36         yTheta(N/10) = min(Theta);
37     end%for
38     plot(xDimension, yTheta); % Erstellung des Diagramms
39 end%for
40 print -depsec ritzPairAngle; % Umwandlung in .eps Datei
41
42 end%function

```

Algorithmus C.3: Berechnung des Winkels zwischen Ritz- und Eigenvektoren

```
1 % Eingabe: -; Ausgabe: -
2 % Funktionsaufruf: iterRayleighRitz
3
4 function iterRayleighRitz
5
6 % Anpassung an Latexfont
7 set(0,'defaulttextinterpreter','latex');
8
9 % Beschriftung des Plots
10 figure; hold on; xIteration = 1:50;
11 xlabel('Iteration'); ylabel('Winkel in Bogenma{\ss}');
12
13 %for i=1:3
14 A = rand(500); A=A'*A; [X, D] = eig(A);
15 Y1 = rand(500,2);
16 k1 = 1;
17
18 for j=1:100
19
20     % Rayleigh-Ritz Prozedur
21     S1 = A^k1*Y1; S1 = orth(S1);
22     A1 = S1'*A*S1;
23     [X1, D1] = eig(A1); Y1 = S1*X1;
24
25     % Berechnung des Winkels
26     p = 1:500; P = nchoosek(p,2);
27     for k = 1:length(P) % = #Zeilen von P
28         Theta(k) = subspace(Y1, X(:,P(k,:)));
29     end
30     yTheta(j) = min(Theta);
31 end%for
32 plot(xIteration, yTheta);
33
34 %print -depsc iterRitzAngle;
35
36 end%function
```

Algorithmus C.4: Berechnung des Winkels zwischen Ritz- und Eigenvektoren bei iterativem Rayleigh-Ritz Verfahren

```

1  % Eingabe: -; % Ausgabe: -
2  % Funktionsaufruf: feast
3
4  function feast
5
6  % Initialisierungen
7  n = 500; iter = 10; lmin = 1; lmax = 2.5;
8
9  % Initialisierung des Plots
10 set(0,'defaulttextinterpreter','latex');
11 figure; hold on;
12 xlabel('Durchgef\{"u\}hrte Iterationen');
13 ylabel('Winkel $\theta_{\min}(\mathcal{X}, \mathcal{R})$');
14
15 % Erzeuge und transformiere die Filterfunktionen r auf
16 % das Intervall [lmin,lmax].
17 x = rkfun(); t = 2/(lmax-lmin)*x - (lmin+lmax)/(lmax-lmin);
18 s = rkfun('step', 5); r = s(t);
19
20 for j=1:3
21     % Erzeuge HPD-Eigenwertproblem und skaliere die Matrizen
22     A = 10*rand(n); A=A'*A - 1;
23     B = 20*rand(n); B=B'*B;
24
25     % Berechne Eigenpaare auf ]lmin,lmax[ mit eig(A,B) als Referenz
26     [X, D] = eig(A,B);
27
28     for i=1:n
29         if lmin < D(i,i) < lmax
30             eigVecInterval(:,i) = X(:,i);
31             eigValInterval(i) = D(i,i);
32         end%if
33     end%for
34
35     Yk = rand(n, length(eigValInterval));
36     xAxis = 1:iter; M = B\A;
37
38     for k=1:iter
39         % Beschleunigtes Rayleigh-Ritz Verfahren
40         Pk = r(M, Yk); % Approximierung des Projektors
41         Ak = Pk'*A*Pk; Bk = Pk'*B*Pk;
42         [Xk, Dk] = eig(Ak, Bk);
43         Yk = Pk*Xk;
44         % Berechnung des Unterraumwinkels
45         theta(k) = subspace(Yk, eigVecInterval);
46     end%for
47     semilogy(xAxis, theta);
48 end%for
49 end%function

```

Algorithmus C.5: FEAST

# Literatur

- [1] *Anzahl der monatlich aktiven Facebook Nutzer weltweit vom 3. Quartal 2008 bis zum 2. Quartal 2017.* <https://de.statista.com/statistik/daten/studie/37545/umfrage/anzahl-der-aktiven-nutzer-von-facebook/>. Letzter Zugriff: 03.09.2017.
- [2] *Besucherzahlen der größten Online-Shops in Deutschland im Oktober 2014.* <https://de.statista.com/statistik/daten/studie/158229/umfrage/online-shops-in-deutschland-nach-besucherzahlen/>. Letzter Zugriff: 03.09.2017.
- [3] Gene H. Golub und Charles F. Van Loan. *Matrix Computations (3rd Ed.)* Baltimore, MD, USA: Johns Hopkins University Press, 1996. ISBN: 0-8018-5414-8.
- [4] Klaus Jänich. *Funktionentheorie: Eine Einführung.* Berlin New York: Springer, 2004. ISBN: 978-3-540-35015-6.
- [5] Alan R. Levin, Deyin Zhang und Eric Polizzi. “FEAST fundamental framework for electronic structure calculations: Reformulation and solution of the muffin-tin problem”. In: *Computer Physics Communications* 183.11 (2012), S. 2370 –2375. ISSN: 0010-4655. DOI: <http://dx.doi.org/10.1016/j.cpc.2012.06.004>. URL: <http://www.sciencedirect.com/science/article/pii/S0010465512002056>.
- [6] J. Kestyn und E. Polizzi und P. Tang. “FEAST eigensolver for non-hermitian problems”. In: (2015).
- [7] Lawrence Page u. a. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, 1999. URL: <http://ilpubs.stanford.edu:8090/422/>.
- [8] Beresford N. Parlett. *The Symmetric Eigenvalue Problem.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998. ISBN: 0-89871-402-8.
- [9] Robert Plato. *Numerische Mathematik kompakt Grundlagenwissen für Studium und Praxis.* Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage, Wiesbaden, 2010. ISBN: 978-3-8348-9644-5.
- [10] Eric Polizzi. “Density-matrix-based algorithm for solving eigenvalue problems”. In: *Phys. Rev. B* 79 (11 2009), S. 115112. DOI: 10.1103/PhysRevB.79.115112. URL: <https://link.aps.org/doi/10.1103/PhysRevB.79.115112>.
- [11] Yousef Saad. *Numerical methods for large eigenvalue problems.* SIAM, 2011. DOI: 10.1137/1.9781611970739.



- [12] *Statistiken zu Netflix*. <https://de.statista.com/themen/1840/netflix/>. Letzter Zugriff: 03.09.2017.
- [13] Mario Berljafa und Stefan Gttel. *Rational Krylov Toolbox for MATLAB*. <http://guettel.com/rktoolbox/index.html>. Letzter Zugriff: 05.09.2017.
- [14] Gilbert W. Stewart. *Matrix Algorithms Volume 2: Eigensystems*. Other Titles in Applied Mathematics. Society for Industrial und Applied Mathematics, 2001. ISBN: 9780898715033. URL: <https://books.google.de/books?id=Y3k5TNTnEmEC>.
- [15] Ping Tak Peter Tang und Eric Polizzi. “FEAST As A Subspace Iteration Eigensolver Accelerated By Approximate Spectral Projection”. In: *SIAM Journal on Matrix Analysis and Applications* 35.2 (2014), S. 354–390. DOI: 10.1137/13090866X. eprint: <https://doi.org/10.1137/13090866X>. URL: <https://doi.org/10.1137/13090866X>.
- [16] *The FEAST algorithm for Hermitian eigenproblems*. [http://guettel.com/rktoolbox/examples/html/example\\_feast.html](http://guettel.com/rktoolbox/examples/html/example_feast.html). Letzter Zugriff: 05.09.2017.
- [17] Marc Van Barel. “Designing rational filter functions for solving eigenvalue problems by contour integration”. In: *Linear Algebra and Its Applications* 502 (Aug. 2016), S. 346–365. DOI: 10.1016/j.laa.2015.05.029. URL: <https://lirias.kuleuven.be/handle/123456789/498035>.
- [18] Jörg Liesen und Zdenek Strakos. *Krylov Subspace Methods: Principles and Analysis*. Numerical Mathematics and Scientific Computation. OUP Oxford, 2013. URL: <https://books.google.de/books?id=hDZoAgAAQBAJ>.