

# **„Weniger ist mehr.“**

Eigenwertfilterung am Beispiel hermitescher  
Eigenwertprobleme

**Thorsten Matthias Lucke**

Betreuer: Prof. Jrg Liesen, Prof. Christian Mehl

Arbeit zur Erlangung des akademischen Grades  
Bachelor of Science

Department Name  
Technische Universitt Berlin

16. Juni 2017

# Dedicatio

For anyone.

# Declaratio

Hiermit erkläre ich, dass ich alles von Wikipedia abgeschrieben habe, so wie sich das gehört für eine großartige Arbeit.

# Recognitio

Danke an

- Thesis Selbsthilfegruppe
- Prof. Liesen, Prof. Mehl
- Chopin, dessen Musik mich durch die Frustphasen der Arbeit geführt hat

# Inhaltsverzeichnis

<b>1</b>	<b>Präludium</b>	<b>6</b>
1.1	Grundlagen . . . . .	7
<b>2</b>	<b>Filtertechniken</b>	<b>8</b>
2.1	Rayleigh-Ritz-Verfahren / Galerkin . . . . .	8
2.2	Konturintegration . . . . .	13
2.3	Beispiel . . . . .	15
<b>3</b>	<b>Beschleunigung durch rationale Funktionen</b>	<b>18</b>
3.1	Konturintegration . . . . .	19
3.2	Der FEAST-Algorithmus . . . . .	19
<b>4</b>	<b>Implementation und numerische Experimente</b>	<b>20</b>
4.1	Rayleigh-Ritz . . . . .	20
4.2	Beschleunigtes RR-Verfahren . . . . .	20
4.3	Feast . . . . .	20
4.4	Mathematische Idee . . . . .	20
4.5	Beispiel . . . . .	23
4.6	Realisierung des FEAST-Algorithmus' . . . . .	25
4.6.1	Naive Implementation . . . . .	25
4.6.2	Rational Krylov Toolbox . . . . .	28
<b>5</b>		<b>31</b>
<b>A</b>	<b>yolo</b>	<b>33</b>

# Kapitel 1

## Präludium

Das Lösen von Eigenwertproblemen ist eine Standarddisziplin in der numerischen linearen Algebra. Gleichungen der Gestalt

$$Ax = \lambda Bx \tag{1.1}$$

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 einer Website von Bedeutung. Entsprechend strotzt der Kanon von angebotenen numerischen Lösungsmethoden von Vielfalt und Virtuosität.

Nun mag der Fall eintreten, da es notwendig wird, lediglich eine Teilmenge aus der Menge aller Eigenpaare zu untersuchen.

**Geeignetes Beispiel finden.**

Wie soll nun aber der eifrige Numeriker die gewünschte Menge an Eigenpaaren finden?

Man könnte zunächst versuchen sämtliche Eigenpaare zu berechnen, die das Problem hergibt und händisch die gewünschte Teilmenge auszuwählen. Bei einem Problem dieser Größenordnung dürfte der Rechenknecht allerdings eine ordentliche Weile beschäftigt sein und das Leben ist fröhlich zu knapp, um auf das Terminieren von Algorithmen zu warten.

---

<sup>1</sup>Hier wird auf den bekannten *Satz vom Fußball* angespielt. Dieser besagt, dass auf einem Fußball zwei Punkte existieren, die zu Spielbeginn und zur Halbzeit an der gleichen Stelle liegen – informell formuliert.

## 1.1 Grundlagen

gauss-legendre quadratur, spektral projektoren (später?), rationale funktionen, spektralzerlegung für allg eigenwertproblem

Um das Lesen dieser Arbeit mehr zu einer Freude denn zu einer Schikane zu machen, soll dieser Abschnitt einige Grundlagen der linearen Algebra und der Funktionentheorie bereitstellen. Obschon sich der Autor bemüht hat, in der Literatur gängige Notation zu benutzen, bittet er den verständnisvollen Leser bei Unklarheiten im Anhang „Notationen“ nachzuschlagen.

Den Anfang machen Definitionen und Resultaten aus der Matrizen­theorie. Eine Matrix  $A \in \mathbb{C}^{n,n}$  wird als *hermitesch* bezeichnet, falls sie die Identität  $A = A^H$  erfüllt. Sie 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  $A \in \mathbb{C}^{n,n}$  eine solche HPD-Matrix und sind  $x, y \in \mathbb{C}^n$ , so werden wir anstelle von  $x^H A y$  die Notation  $\langle x, y \rangle_A$  verwenden. Im Falle  $A = I_n$  schreiben wir kurz  $\langle x, y \rangle$ .

ne anders:  $\langle x, y \rangle = y^H x$  oder doch nicht anders? einfach einträge tauschen oder so  
Im Rahmen dieser Arbeit werden überwiegend Hermitesche Eigenwertprobleme betrachtet.

**Proposition 1.1.** Ist  $p$  eine Projektion auf den Unterraum  $U$ , dann gilt  $p(u) = u$  für alle  $u \in U$ .

*Beweis.* Sei zunächst  $u \in \text{Bild } p$ . Dann existiert  $v \in V$  mit  $p(v) = u$ . Nach Definition gilt aber auch  $p^2(v) = u = p(u) = u$ . Also folgt  $p(u) = u$ . gilt  $\text{Bild}(p) = U$ ?  $\square$

Bilinearität von  $\langle \cdot, \cdot \rangle_A$  Kontur (komplex), Jordankurven, Invarianter Unterraum

# Kapitel 2

## Filtertechniken

Dieses Kapitel widmet sich der Frage, ob und wie man aus der Menge aller Eigenpaare des Problems

$$Ax = \lambda Bx$$

eine gewünschte Teilmenge auswählen kann. Dazu werden im Rahmen dieser Arbeit aus dem Katalog der Verfahren die zwei Folgenden ausgewählt und vorgestellt: das Rayleigh-Ritz-Verfahren / Galerkin-Methoden<sup>1</sup> und Konturintegration.

Ob ein Zusammenhang zwischen beiden Konzepten besteht, wird im nächsten Kapitel geklärt.

### 2.1 Rayleigh-Ritz-Verfahren / Galerkin

Eine **Die?** Idee dieser Klasse von Verfahren ist das Approximieren des von den gesuchten Eigenvektoren aufgespannten Unterraums. Bevor wir das allgemeine Eigenwertproblem untersuchen, wenden wir unsere Aufmerksamkeit dem gewöhnlichen Eigenwertproblem

$$Ax = \lambda x$$

zu und betrachten für eine Zahl  $m \in \mathbb{N}$  mit  $m \leq n$  einen  $m$ -dimensionalen Unterraum  $\mathcal{U}_m \subseteq \mathbb{C}^n$ . Dieser zunächst nicht näher bestimmte *Suchraum* wird als Grundlage für das Verfahren gewählt. Gesucht sind nun Paare  $(\tilde{\lambda}, \tilde{x}) \in \mathbb{C} \times \mathcal{U}_m \setminus \{0\}$  – die wir als approximierte Lösungen des Eigenproblems verstehen wollen – welche die Eigenschaft

$$\langle u, A\tilde{x} - \tilde{\lambda}\tilde{x} \rangle = 0 \tag{2.1}$$

für alle  $u \in \mathcal{U}_m$  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*

---

<sup>1</sup>Galerkin und Rayleigh ritz literatur blabla



bezüglich des Suchraums  $\mathcal{U}_m$  genannt.

Wir wollen nun annehmen, dass mit der Menge von Vektoren  $\{u_i\}_{i=1:m} \subseteq \mathbb{C}^n$  eine Orthonormalbasis des Unterraums  $\mathcal{U}_m$  gegeben ist. Definieren wir dann die Matrix  $U_m := [u_i]_{i=1:m} \in \mathbb{C}^{n,m}$ , so muss wegen  $\tilde{x} \in \mathcal{U}_m$  ein Vektor  $y \in \mathbb{C}^m$  existieren, mit  $U_m y = \tilde{x}$ . Die Forderung (2.1) ist dann zu der Gleichung

$$U_m^H (AU_m y - \tilde{\lambda} U_m y) = 0$$

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

$$(U_m^H A U_m) y = \tilde{\lambda} y \tag{2.2}$$

ein neues Eigenwertproblem. Jedes Eigenpaar  $(\tilde{\lambda}, y)$  von (2.2) liefert dann nach Konstruktion mit  $(\tilde{\lambda}, U_m y)$  ein Ritz-Paar des gewöhnlichen Eigenwertproblems bezüglich des Suchraums  $\mathcal{U}_m$ . In Abhängigkeit von der Wahl des Suchraumes variiert natürlich Güte der Approximation. Ist etwa  $\mathcal{U}_m$  durch Eigenvektoren aufgespannt, so ist jedes Ritz-Paar schon ein Eigenpaar. Wir werden später aus den Untersuchungen des allgemeinen Eigenwertproblems folgern, dass bereits im Falle der Invarianz von  $\mathcal{U}_m$  unter  $A$  jedes Ritz-Paar von  $A$  bereits ein Eigenpaar ist. An dieser Stelle begnügen wir uns vorerst mit dieser erfreulichen Botschaft.

Zusammengefasst ergibt sich mit den obigen Überlegungen für die Berechnung von Ritz-Paaren folgender Algorithmus (XXX):

- berechne ONB  $\{u_i\}_{i=1:m}$  von  $\mathcal{U}_m$ , setze  $U_m := [u_i]_{i=1:m}$
- setze  $\tilde{A} := U_m^H A U_m$  und löse  $\tilde{A} y_i = \lambda_i y_i$ ,  $y_i \in \mathbb{C}^m$ ,  $i = 1 : m$
- $x_i := U_m y_i$
- gib Ritz-Paare  $(\lambda_i, x_i)$  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 (2.2) Ritz-Paare zu extrahieren, aber wäre nicht auch denkbar, sämtliche Eigenpaare von  $A$  zu approximieren und die zum Unterraum  $\mathcal{U}_m$  korrespondierende Teilmenge direkt auszuwählen?

Dies mag in Einzelfällen in der Tat sinnvoller sein. Sprechen wir allerdings von Matrixdimensionen jenseits der Vorstellungskraft, ist eine vollständige Berechnung aller Eigenpaare mitunter ein sehr zeitintensives Vergnügen. Bei genauerer Betrachtung der Gleichung (2.2) fällt auf, dass die Matrix  $(U_m^H A U_m) \in \mathbb{C}^{m,m}$  im Falle  $n \gg m$  ein mitunter 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 (XXX) geringer ist, als beim Approximieren sämtlicher Eigenpaare von  $A$ . Dies wird im vierten Kapitel anhand ausgewählter Beispiele illustriert.

Wir gehen nun einen Schritt weiter und erweitern die obige Theorie auf das verallgemeinerte Eigenwertproblem

$$Ax = \lambda Bx. \quad (2.3)$$

Dabei gehen wir ganz analog zum gewöhnlichen Eigenwertproblem vor. Es sei daher wieder  $\mathcal{U}_m \subseteq \mathbb{C}^n$  ein  $m$ -dimensionaler Suchraum. Gefunden werden sollen nun Paare  $(\tilde{\lambda}, \tilde{x}) \in \mathbb{C} \times \mathcal{U}_m \setminus \{0\}$  die der Orthogonalitätsbedingung

$$A\tilde{x} - \tilde{\lambda}B\tilde{x} \perp \mathcal{U}_m \quad (2.4)$$

genügen. Wieder substituieren wir  $U_m y = \tilde{x}$  und erhalten die zu (2.4) äquivalente Forderung

$$U_m^H (AU_m y - \tilde{\lambda}BU_m y) = 0.$$

Schließlich lösen wir das transformierte Eigenwertproblem

$$(U_m^H A U_m)y = \tilde{\lambda}(U_m^H B U_m)y. \quad (2.5)$$

Wie bereits beim gewöhnlichen Eigenwertproblem kann aus jeder Lösung  $(\tilde{\lambda}, y)$  von (2.5) mit  $(\tilde{\lambda}, U_m y)$  ein Ritz-Paar für das verallgemeinerte Eigenwertproblem gewonnen werden.

Mit dieser Konstruktion lässt sich der Algorithmus (XXX) wie folgt abwandeln:

- berechne ONB  $\{u_i\}_{i=1:m}$  von  $\mathcal{U}_m$ , setze  $U_m := [u_i]_{i=1:m}$
- setze  $\tilde{A} := U_m^H A U_m$ ,  $\tilde{B} := U_m^H B U_m$  und löse  $\tilde{A}y_i = \lambda_i \tilde{B}y_i$ ,  $y_i \in \mathbb{C}^n, i = 1 : m$
- $x_i := U_m y_i$
- gib Ritz-Paare  $(\lambda_i, x_i)$  aus

Nun, da die Idee der Verfahren vorgestellt wurde, wollen wir einige Beobachtungen festhalten. Eingangs wurde behauptet, dass die Invarianz des Suchraumes  $\mathcal{U}_m$  unter

$A$  dazu führt, dass jedes Ritz-Paar bezüglich  $\mathcal{U}_m$  ein Eigenpaar von  $A$  ist. Doch warum ist das so? Dies zu beantworten verpflichtet sich der folgende Satz.

**Satz 2.1.** Neben zwei Matrizen  $A, B \in \mathbb{C}^{n,n}$  – wobei  $B$  eine HPD-Matrix ist – sei für  $m \in \mathbb{N}$  mit  $m \leq n$  ein  $m$ -dimensionaler Unterraum  $\mathcal{U}_m \subseteq \mathbb{C}^n$  gegeben. Ist dieser invariant unter  $B$ , so ist jedes  $B$ -Ritzpaar von  $A$  bezüglich  $\mathcal{U}_m$  auch ein verallgemeinertes Eigenpaar von  $A$ .

*Beweis.* Ausstehend. Die erforderlichen Bedingungen und die Beweisführung ist noch nicht klar.  $\square$

Da das gewöhnliche Eigenwertproblem ein Spezialfall des allgemeinen Eigenwertproblems ist, können wir aus dem eben bewiesenen Resultat unmittelbar das folgende Korollar ableiten.

**Korollar 2.2.** Ist  $A \in \mathbb{C}^{n,n}$  und  $\mathcal{U}_m \subseteq \mathbb{C}^n$  ein  $m$ -dimensionaler Unterraum, so ist jedes Ritz-Paar von  $A$  bezüglich  $\mathcal{U}_m$  ein Eigenpaar von  $A$ .

*Beweis.* Betrachte für  $B := I_n$  das verallgemeinerte Eigenwertproblem

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

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

Diese Erkenntnis ist vor allem aus algorithmischer Sicht höchst nützlich. Implementieren man nämlich das Rayleigh-Ritz-Verfahren iterativ (siehe Quelle blablabla), so kann der Iterationsprozess abgebrochen werden, sobald der Unterraum  $\text{Bild}(U_m^i)$  invariant unter  $(?)$  ist.

Das Rayleigh-Ritz-Verfahren steht in einem engen Zusammenhang zu Projektionsverfahren. Mit einer geeigneten Projektionsmatrix  $P \in \mathbb{C}^{n,n}$  ist ein Paar  $(\lambda, x) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$  (oder  $\in \mathbb{C} \times \mathcal{U}_m \setminus \{0\}$ ?) genau dann eine Lösung der Gleichung (2.3), wenn es eine Lösung von

$$P^H A P x = \lambda P^H B P x$$

ist. Richtig definiert, besitzt  $P$  noch weitere nützliche Eigenschaften, die der folgende Satz vorstellt.

**Satz 2.3.** Es sei  $B \in \mathbb{C}^{n,n}$  eine HPD-Matrix und für  $m \in \mathbb{N}$  mit  $m \leq n$  sei ein  $m$ -dimensionaler Unterraum  $\mathcal{U}_m \subseteq \mathbb{C}^n$  gegeben. Sei weiter  $\{u_i\}_{i=1:m} \subseteq \mathcal{U}_m$  eine Basis  $B$ -orthogonaler Vektoren, das heißt, die Matrix  $U_m := [u_i]_{i=1:m} \in \mathbb{C}^{n,m}$  erfülle die Gleichung  $U_m^H B U_m = I_m$ . Dann ist die von der Matrix  $P := U_m U_m^H B \in \mathbb{C}^{n,n}$  induzierte lineare Abbildung

$$p: \mathbb{C}^n \rightarrow \mathbb{C}^n, x \mapsto U_m U_m^H B x$$

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

$$\|x - p(x)\|_B = \min_{y \in \mathcal{U}} \|x - y\|_B, x \in \mathbb{C}^n.$$

*Beweis.* Aus der  $B$ -Orthogonalität von  $U_m$  folgt für alle  $x \in \mathbb{C}^n$

$$p^2(x) = U_m U_m^H B U_m U_m^H B x = U_m U_m^H B x = p(x)$$

und damit gilt die Mengengleichheit  $p(\mathbb{C}^n) = \mathcal{U}$  nach Konstruktion. Ist nun  $y \in \mathcal{U}$  und  $x \in \mathbb{C}^n$ , dann folgt wegen  $p(y) = y$  und  $B = B^H$  auch

$$\begin{aligned} \langle y, x - p(x) \rangle_B &= y^H (Bx - B U_m U_m^H B x) \\ &= y^H (Bx - B^H U_m U_m^H B x) \\ &= y^H B x - p(y)^H B x = 0. \end{aligned}$$

Es gilt demnach  $x - p(x) \perp_B \mathcal{U}$ . Die Optimierungsaufgabe wird 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, denn aufgrund der positiven Definitheit von Normen, gilt Gleichheit genau dann, wenn  $p(x) = x$  erfüllt ist.  $\square$

Mit Hilfe der im Satz eingeführten Matrix  $P$ , lässt sich die Gleichung (2.5) elegant umformulieren. Durch Linksmultiplikation mit  $B^H U_m$  und dem Einschub der Identität  $U_m^H B U_m$  gilt nämlich

$$B^H U_m (U_m^H A U_m) (U_m^H B U_m) y = \lambda B^H U_m (U_m^H B U_m) (U_m^H B U_m) y$$

und folglich

$$P^H A P U_m y = \lambda P^H B P U_m y.$$

Wenn wir uns an dieser Stelle erinnern, dass  $x$  durch  $U_m y$  ersetzt wurde, dann erhalten wir tatsächlich

$$P^H A P x = \lambda P^H B P x.$$

Im Fall der gewöhnlichen Eigenwertgleichung erhalten wir speziell

$$P A P x = \lambda x.$$

Es ist leider nicht ohne Weiteres möglich einen bestimmten Eigenraum gezielt zu

approximieren, wenn keine weiteren Informationen bekannt sind. Gängige Wahl für Suchräume sind Krylow-Räume. Deren Struktur führt dazu, dass Unterräume approximiert werden, die von den Eigenvektoren mit den betragsmäßig größten Eigenwerten aufgespannt werden. Es obliegt daher dem persönlichen Geschick einen passenden Suchraum aufzustellen.

Im kommenden Abschnitt wird daher eine Methode vorgestellt, mit dessen Hilfe gezielt Eigenpaare gefunden werden können.

## 2.2 Konturintegration

Wir wollen nun annehmen, dass zwei Zahlen  $\lambda_1, \lambda_2 \in \mathbb{R}$  vorgegeben sind, die das abgeschlossene Intervall  $I := [\lambda_1, \lambda_2] \subseteq \mathbb{R}$  definieren. Für zwei hermitesche Matrizen  $A, B \in \mathbb{C}^{n,n}$  mit der zusätzlichen Forderung, dass  $B$  positiv definit ist, sollen Paare der Gestalt  $(\lambda, x) \in I \times \mathbb{C}^n$  ermittelt werden, welche der verallgemeinerten Eigenwertgleichung

$$Ax = \lambda Bx \tag{2.6}$$

genügen. Um diese Eigenpaare<sup>2</sup> bestimmen zu können, führen wir die durch

$$\begin{aligned} G: \Omega &\rightarrow \mathbb{C}^{n,n} \\ \omega &\mapsto (\omega B - A)^{-1} \end{aligned}$$

definierte GREEN-Funktion, wobei  $\Omega \subseteq \mathbb{C}$  eine passende Teilmenge der komplexen Zahlen ist. Insbesondere müssen  $\Omega$  und das Spektrum von  $B^{-1}A$  disjunkt sein, da  $G$  andernfalls nicht wohldefiniert ist.

Die Funktion  $G$  wird nun über eine Jordan-Kurve  $\gamma$  die um das vorgegebene Intervall  $I$  herumläuft, in der Gestalt

$$-\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega$$

integriert.<sup>3</sup>

Wir wollen nun annehmen, dass es genau  $k \in \mathbb{N}$  Eigenwerte gibt, die im Inneren des Intervalls  $I$  liegen.<sup>4</sup> Für zu diesen Eigenwerten passende Eigenvektoren

---

<sup>2</sup>Um der besseren Lesbarkeit Willen werden im Folgenden die verallgemeinerten Eigenvektoren und deren verallgemeinerte Eigenwerte kurz als Eigenvektor und Eigenwert bezeichnet.

<sup>3</sup>Mit  $\iota$  ist im Folgenden stets die imaginäre Einheit bezeichnet, also  $\iota = \sqrt{-1}$ .

<sup>4</sup>Diese Forderung ist wohldefiniert, da der Eigenschaften von  $A$  und  $B$  wegen alle Eigenwerte

$\{x_i\}_{i=1:k} \subseteq \mathbb{C}^n$  sei außerdem die Matrix  $X_k := [x_i]_{i=1:k}$  gegeben. Dann lässt die Konturintegration Rückschlüsse auf die Eigenvektoren zu, denn mit Mitteln der komplexen Analysis kann man zeigen, dass die Identität

$$X_k X_k^T = \sum_{i=1}^k x_i x_i^T = -\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega \quad (2.7)$$

gilt. **genauer? Wie sieht X genau aus?**

Wir wählen nun  $k$  linear unabhängige Vektoren  $\{y_i\}_{i=1:k} \subseteq \mathbb{C}^n$  aus, die wir in der Matrix  $Y_k := [y_i]_{i=1:k}$  zusammenfassen. Unsere gewünschte Matrix  $Q$  definieren wir nun durch Matrixmultiplikation vermöge

$$Q := X_k X_k^T Y_k. \quad (2.8)$$

Die Spalten von  $Q$  sind also gerade Linearkombinationen aus den Eigenvektoren. Sind dann  $\{\phi_i\}_{i=1:k} \subseteq \mathbb{C}^n$  Vektoren die das transformierte Eigenwertproblem (4.2) lösen und ist  $\Phi_k := [\phi_i]_{i=1:k}$  gegeben, so gilt  $X_k = Q\Phi_k$ .

In der Praxis sind natürlich weder die Anzahl der Eigenwerte noch die dazugehörigen Eigenvektoren bekannt. Während die Anzahl mit Hilfe einer passenden Filterfunktion bestimmt werden kann, verhilft die wegen (4.3) und (4.4) gültige Gleichheit

$$Q = -\frac{1}{2\pi\iota} \left( \int_{\gamma} G(\omega) \, d\omega \right) \cdot Y \quad (2.9)$$

zur Berechnung der Eigenvektoren.

---

von  $B^{-1}A$  reell sind.

## 2.3 Beispiel

In diesem Abschnitt soll mit Hilfe eines einfachen Beispiels die eben erarbeitete Theorie verifiziert werden. Dafür betrachten wir die beiden Matrizen  $A, B \in \mathbb{C}^{n,n}$ , welche durch

$$A := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{und} \quad B := \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

gegeben sind. Durch einfaches Nachrechnen überprüft man, dass das verallgemeinerte Eigenwertproblem (4.1) durch Vektoren  $x_1 \in \text{Span}_{\mathbb{C}}\{e_1\}$  mit zugehörigem Eigenwert  $\lambda_1 = 0$ , Vektoren  $x_2 \in \text{Span}_{\mathbb{C}}\{e_2\}$  mit zugehörigem Eigenwert  $\lambda_2 = 2$ , sowie Vektoren  $x_3 \in \text{Span}_{\mathbb{C}}\{e_3\}$  mit zugehörigem Eigenwert  $\lambda_3 = -4$  gelöst wird.<sup>5</sup>

Wir wollen nun auf dem reellen Intervall  $[-3, 3]$  die Eigenpaare bestimmen. Als Integrationskontur wählen wir daher den Kreis mit einem Radius von drei Längeneinheiten und dem Mittelpunkt im Ursprung. Das heißt, wir werden die GREEN-Funktion  $G$  über die Kurve

$$\gamma: [0, 2\pi] \rightarrow \mathbb{C}^n, \quad \varphi \mapsto 3e^{i\varphi} \quad (2.10)$$

integrieren. Da nach Konstruktion keiner der Eigenwerte auf dem Graphen von  $\gamma$  liegt, ist  $G$  auf der gesamten Kontur wohldefiniert.

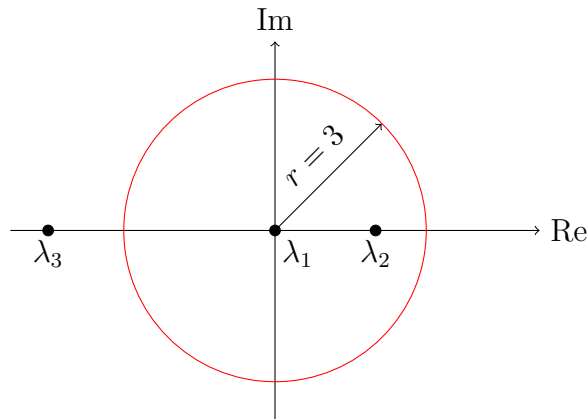


Abbildung 2.1: Skizze der Kurve  $\gamma$  in der komplexen Ebene.

Zunächst überprüfen wir die Identität (4.3). Mit  $x_1 := \begin{bmatrix} \sqrt{2}t & 0 & 0 \end{bmatrix}^T$  und  $x_2 := \begin{bmatrix} 0 & \sqrt{2}t & 0 \end{bmatrix}^T$  sind zwei passende Eigenvektoren für unser Eigenwertproblem gegeben.

<sup>5</sup>Hier bezeichnen  $e_1, e_2, e_3 \in \mathbb{C}^n$  die kanonischen Einheitsvektoren.

ben, und in der Tat gilt

$$\begin{aligned}
-\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega &= -\frac{1}{2\pi\iota} \int_0^{2\pi} G(\gamma(\omega)) \cdot \gamma'(\omega) \, d\omega \\
&= -\frac{1}{2\pi\iota} \int_0^{2\pi} \begin{bmatrix} \frac{2}{3e^{\iota\omega}} & 0 & 0 \\ 0 & \frac{2}{3e^{\iota\omega}-2} & 0 \\ 0 & 0 & \frac{4}{3e^{\iota\omega}+4} \end{bmatrix} \cdot 3\iota e^{\iota\omega} \, d\omega \\
&= \text{diag}(-2, -2, 0) \\
&= x_1 x_1^T + x_2 x_2^T.
\end{aligned}$$

Als Nächstes überprüfen wir, ob die Matrix  $X_2 := \begin{bmatrix} x_1 & x_2 \end{bmatrix}$  mit Hilfe der in Abschnitt 4.4 beschriebenen Methode rekonstruierbar ist.

Dort wird von uns zunächst die Konstruktion der Matrix  $Q$  aus (4.4) verlangt. Da wir bereits wissen, dass wir zwei Eigenvektoren finden wollen, wählen wir also beliebig eine Matrix  $Y_2 \in \mathbb{C}^{3,2}$  vollen Ranges – die wir in diesem Beispiel mit  $Y_2 := \begin{bmatrix} e_1 & e_1 + e_2 \end{bmatrix}$  festlegen wollen – und multiplizieren diese von rechts an das Integral der GREEN-Funktion über die durch (4.6) definierte Kontur.<sup>6</sup> Wir erhalten somit

$$Q = -\frac{1}{2\pi\iota} \left( \int_{\gamma} G(\omega) \, d\omega \right) \cdot Y_2 = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 0 & -2 \\ 0 & 0 \end{bmatrix}.$$

Nun lösen wir das in (4.2) beschriebene Eigenwertproblem. Wir suchen also Paare  $(\mu, \phi) \in \mathbb{C} \times \mathbb{C}^n$  die der Gleichung

$$A_Q \phi = \mu B_Q \phi$$

genügen. Durch Nachrechnen überzeugt man sich davon, dass dieses von allen Vektoren  $\phi_1 \in \text{Span}_{\mathbb{C}}\{e_1\}$  mit dem Eigenwert  $\mu_1 = 0$  sowie allen Vektoren  $\phi_2 \in \text{Span}_{\mathbb{C}}\{-e_1 + e_2\}$  mit dem Eigenwert  $\mu_2 = 2$  gelöst wird.

Die soeben ermittelten Eigenwerte stimmen also schon mit den Eigenwerten unseres betrachteten Problems überein. Wählen wir schließlich die Matrix

$$\Phi_2 := \begin{bmatrix} \phi_1 & \phi_2 \end{bmatrix} := \begin{bmatrix} -\iota/\sqrt{2} & \iota/\sqrt{2} \\ 0 & -\iota/\sqrt{2} \end{bmatrix},$$

---

<sup>6</sup>Es mag den Leser verwundern, wie bei Unkenntnis über die Anzahl  $k \in \mathbb{N}$  der zu findenden Eigenvektoren die Spaltenzahl von  $Y_k$  korrekt zu ermitteln ist. Eine gängige Methode ist zum Beispiel das Konstruieren einer passenden Filterfunktion, welche die nicht erwünschten Eigenwerte aussortiert (Vgl. Abschnitt 4.6.2).



mit den Eigenvektoren  $\phi_1$  und  $\phi_2$ , so erhalten wir wie gewünscht

$$Q\Phi_2 = \begin{bmatrix} -2 & -2 \\ 0 & -2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\iota/\sqrt{2} & \iota/\sqrt{2} \\ 0 & -\iota/\sqrt{2} \end{bmatrix} = \begin{bmatrix} \sqrt{2}\iota & 0 \\ 0 & \sqrt{2}\iota \\ 0 & 0 \end{bmatrix} = X_2.$$

**Bemerkung.** An dieser Stelle sei darauf hingewiesen, dass die Matrix  $\Phi_2$  aus kosmetischen Gründen gerade so gewählt wurde, dass die Kalkulation am Ende  $X_2$  ergibt und somit alles hübsch anzusehen ist. Der Wahl einer anderen Matrix  $\Phi'_2$  mit skalaren Vielfachen der hier gewählten Eigenvektoren steht selbstverständlich nichts im Wege und führt freilich zu einer anderen Matrix  $X'_2$ , deren Spalten ebenfalls zulässige Eigenvektoren enthält.

# Kapitel 3

## Beschleunigung durch rationale Funktionen

Dieses Kapitel wird aufzeigen, dass sich Projektive Verfahren und Konturintegration verheiraten lassen :D

Nachdem vorgestellt wurde, wie mit Polynomen Eigenwerte extrahiert werden können, verpflichtet sich dieses Kapitel der Vorstellung einer anderen Klasse von Methoden, die das Filtern von Eigenwerten ermöglichen. Die Protagonisten dieser Methoden sind die sogenannten *rationale Funktionen*. Ausgehend von zwei Polynomen  $p, q \in \mathbb{C}[t]$  mit

$$p := \sum_{k=0}^n p_k t^k \quad \text{und} \quad q := \sum_{k=0}^n q_k t^k$$

definieren wir eine rationale Funktion  $r: \mathbb{C} \setminus N_q \rightarrow \mathbb{C}$  vermöge

$$r(t) := \frac{p(t)}{q(t)}$$

und identifizieren wie üblich die Argumente von  $p$  und  $q$  mit der Unbestimmten  $t$ . Eine rationale Funktion wird als *echt gebrochen* bezeichnet, falls die Bedingung  $\text{Grad}(p) < \text{Grad}(q)$  erfüllt ist. **Nullstellen von  $q$  müssen aus dem Definitionsbereich verschwinden.**

Wir erinnern uns an das Rayleigh-Ritz-Verfahren aus Kapitel 2. Die Konvergenz (?) hängt davon ab, mit welchem Startvektor das Verfahren beginnt...

Ein optimaler Filter ist der Spektralprojektor  $\rho(B^{-1}A) = X_k X_k^H B \dots$  warum, siehe PTEP S. 356.

### **3.1 Konturintegration**

### **3.2 Der FEAST-Algorithmus**

# Kapitel 4

## Implementation und numerische Experimente

Da für die im Folgenden diskutierten Verfahren eine Methode benötigt wird, um die in Dimension reduzierten Eigenwertprobleme zu lösen, betrachten wir kurz das folgende Verfahren

### QR-ITERATION

#### 4.1 Rayleigh-Ritz

#### 4.2 Beschleunigtes RR-Verfahren

#### 4.3 Feast

#### 4.4 Mathematische Idee

**JORDAN KURVE?** Zunächst wollen wir uns mit den mathematischen Ideen, welche die Grundlage für den FEAST-Algorithmus bilden, vertraut machen. Dafür werden wir die von E. Polizzi in seinem Paper [Pol09] verwendeten Notationen in großen Teilen – aber nicht vollständig – übernehmen.

Es seien nun zwei Zahlen  $\lambda_1, \lambda_2 \in \mathbb{R}$  vorgegeben, die das reelle Intervall  $I := [\lambda_1, \lambda_2] \subseteq \mathbb{R}$  definieren. Für zwei hermitesche Matrizen  $A, B \in \mathbb{C}^{n,n}$  mit der zusätzlichen Forderung, dass  $B$  positiv definit ist, sollen Paare der Gestalt  $(\lambda, x) \in I \times \mathbb{C}^n$  ermittelt werden, welche der verallgemeinerten Eigenwertgleichung

$$Ax = \lambda Bx \tag{4.1}$$

genügen. Um diese Eigenpaare<sup>1</sup> bestimmen zu können, wird das Problem mit Hilfe einer gewissen Matrix  $Q \in \mathbb{C}^{n,n}$  in das äquivalente Problem

$$A_Q \phi = \mu B_Q \phi \quad (4.2)$$

überführt. Hierbei sind  $A_Q = Q^T A Q$ ,  $B_Q = Q^T B Q$  und  $(\mu, \phi) \in I \times \mathbb{C}^n$ . Wird die Matrix  $Q$  richtig gewählt, so ist dann jeder zulässige Eigenwert von (4.2) auch ein zulässiger Eigenwert von (4.1) und umgekehrt.<sup>2</sup> Für die Ermittlung der gesuchten Eigenvektoren bedarf es hingegen zusätzlicher Arbeit.

Zur Bestimmung der Transformationsmatrix  $Q$  wird eine Konturintegration bemüht. Ausgangspunkt dieser Integration ist die durch

$$G: \Omega \rightarrow \mathbb{C}^{n,n} \\ \omega \mapsto (\omega B - A)^{-1}$$

definierte GREEN-Funktion, wobei  $\Omega \subseteq \mathbb{C}$  eine passende Teilmenge der komplexen Zahlen ist. Insbesondere müssen  $\Omega$  und das Spektrum von  $B^{-1}A$  disjunkt sein, da  $G$  andernfalls nicht wohldefiniert ist.

Die Funktion  $G$  wird nun über eine geschlossene komplexe Kurve  $\gamma$ , die um das vorgegebene Intervall  $I$  herumläuft, in der Gestalt

$$-\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega$$

integriert.<sup>3</sup>

Wir wollen nun annehmen, dass es genau  $k \in \mathbb{N}$  Eigenwerte gibt, die im Inneren des Intervalls  $I$  liegen.<sup>4</sup> Für zu diesen Eigenwerten passende Eigenvektoren  $\{x_i\}_{i=1:k} \subseteq \mathbb{C}^n$  sei außerdem die Matrix  $X_k := [x_i]_{i=1:k}$  gegeben. Dann lässt die Konturintegration Rückschlüsse auf die Eigenvektoren zu, denn man kann zeigen, dass die Identität

$$X_k X_k^T = \sum_{i=1}^k x_i x_i^T = -\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega \quad (4.3)$$

<sup>1</sup>Um der besseren Lesbarkeit Willen werden im Folgenden die verallgemeinerten Eigenvektoren und deren verallgemeinerte Eigenwerte kurz als Eigenvektor und Eigenwert bezeichnet.

<sup>2</sup>Daher ist die Angabe  $\mu \in I$  gerechtfertigt.

<sup>3</sup>Mit  $\iota$  ist im Folgenden stets die imaginäre Einheit bezeichnet, also  $\iota = \sqrt{-1}$ .

<sup>4</sup>Diese Forderung ist wohldefiniert, da der Eigenschaften von  $A$  und  $B$  wegen alle Eigenwerte von  $B^{-1}A$  reell sind.

gilt.

Wir wählen nun  $k$  linear unabhängige Vektoren  $\{y_i\}_{i=1:k} \subseteq \mathbb{C}^n$  aus, die wir in der Matrix  $Y_k := [y_i]_{i=1:k}$  zusammenfassen. Unsere gewünschte Matrix  $Q$  definieren wir nun durch Matrixmultiplikation vermöge

$$Q := X_k X_k^T Y_k. \quad (4.4)$$

Die Spalten von  $Q$  sind also gerade Linearkombinationen aus den Eigenvektoren. Sind dann  $\{\phi_i\}_{i=1:k} \subseteq \mathbb{C}^n$  Vektoren die das transformierte Eigenwertproblem (4.2) lösen und ist  $\Phi_k := [\phi_i]_{i=1:k}$  gegeben, so gilt  $X_k = Q\Phi_k$ .

In der Praxis sind natürlich weder die Anzahl der Eigenwerte noch die dazugehörigen Eigenvektoren bekannt. Während die Anzahl mit Hilfe einer passenden Filterfunktion bestimmt werden kann, verhilft die wegen (4.3) und (4.4) gültige Gleichheit

$$Q = -\frac{1}{2\pi\iota} \left( \int_{\gamma} G(\omega) \, d\omega \right) \cdot Y \quad (4.5)$$

zur Berechnung der Eigenvektoren.

## 4.5 Beispiel

In diesem Abschnitt soll mit Hilfe eines einfachen Beispiels die eben erarbeitete Theorie verifiziert werden. Dafür betrachten wir die beiden Matrizen  $A, B \in \mathbb{C}^{n,n}$ , welche durch

$$A := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \text{und} \quad B := \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

gegeben sind. Durch einfaches Nachrechnen überprüft man, dass das verallgemeinerte Eigenwertproblem (4.1) durch Vektoren  $x_1 \in \text{Span}_{\mathbb{C}}\{e_1\}$  mit zugehörigem Eigenwert  $\lambda_1 = 0$ , Vektoren  $x_2 \in \text{Span}_{\mathbb{C}}\{e_2\}$  mit zugehörigem Eigenwert  $\lambda_2 = 2$ , sowie Vektoren  $x_3 \in \text{Span}_{\mathbb{C}}\{e_3\}$  mit zugehörigem Eigenwert  $\lambda_3 = -4$  gelöst wird.<sup>5</sup>

Wir wollen nun auf dem reellen Intervall  $[-3, 3]$  die Eigenpaare bestimmen. Als Integrationskontur wählen wir daher den Kreis mit einem Radius von drei Längeneinheiten und dem Mittelpunkt im Ursprung. Das heißt, wir werden die GREEN-Funktion  $G$  über die Kurve

$$\gamma: [0, 2\pi] \rightarrow \mathbb{C}^n, \quad \varphi \mapsto 3e^{i\varphi} \quad (4.6)$$

integrieren. Da nach Konstruktion keiner der Eigenwerte auf dem Graphen von  $\gamma$  liegt, ist  $G$  auf der gesamten Kontur wohldefiniert.

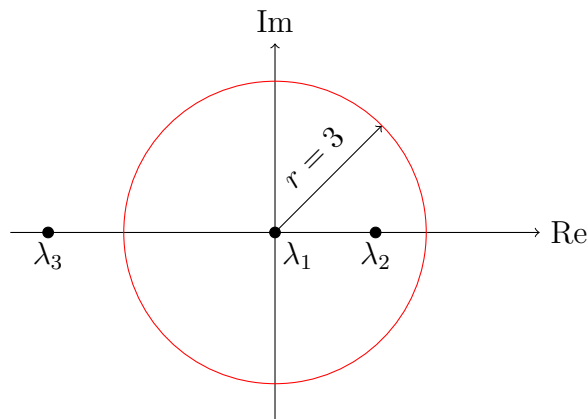


Abbildung 4.1: Skizze der Kurve  $\gamma$  in der komplexen Ebene.

Zunächst überprüfen wir die Identität (4.3). Mit  $x_1 := \begin{bmatrix} \sqrt{2}t & 0 & 0 \end{bmatrix}^T$  und  $x_2 := \begin{bmatrix} 0 & \sqrt{2}t & 0 \end{bmatrix}^T$  sind zwei passende Eigenvektoren für unser Eigenwertproblem gegeben.

<sup>5</sup>Hier bezeichnen  $e_1, e_2, e_3 \in \mathbb{C}^n$  die kanonischen Einheitsvektoren.

ben, und in der Tat gilt

$$\begin{aligned}
 -\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega &= -\frac{1}{2\pi\iota} \int_0^{2\pi} G(\gamma(\omega)) \cdot \gamma'(\omega) \, d\omega \\
 &= -\frac{1}{2\pi\iota} \int_0^{2\pi} \begin{bmatrix} \frac{2}{3e^{\iota\omega}} & 0 & 0 \\ 0 & \frac{2}{3e^{\iota\omega}-2} & 0 \\ 0 & 0 & \frac{4}{3e^{\iota\omega}+4} \end{bmatrix} \cdot 3\iota e^{\iota\omega} \, d\omega \\
 &= \text{diag}(-2, -2, 0) \\
 &= x_1 x_1^T + x_2 x_2^T.
 \end{aligned}$$

Als Nächstes überprüfen wir, ob die Matrix  $X_2 := \begin{bmatrix} x_1 & x_2 \end{bmatrix}$  mit Hilfe der in Abschnitt 4.4 beschriebenen Methode rekonstruierbar ist.

Dort wird von uns zunächst die Konstruktion der Matrix  $Q$  aus (4.4) verlangt. Da wir bereits wissen, dass wir zwei Eigenvektoren finden wollen, wählen wir also beliebig eine Matrix  $Y_2 \in \mathbb{C}^{3,2}$  vollen Ranges – die wir in diesem Beispiel mit  $Y_2 := \begin{bmatrix} e_1 & e_1 + e_2 \end{bmatrix}$  festlegen wollen – und multiplizieren diese von rechts an das Integral der GREEN-Funktion über die durch (4.6) definierte Kontur.<sup>6</sup> Wir erhalten somit

$$Q = -\frac{1}{2\pi\iota} \left( \int_{\gamma} G(\omega) \, d\omega \right) \cdot Y_2 = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 0 & -2 \\ 0 & 0 \end{bmatrix}.$$

Nun lösen wir das in (4.2) beschriebene Eigenwertproblem. Wir suchen also Paare  $(\mu, \phi) \in \mathbb{C} \times \mathbb{C}^n$  die der Gleichung

$$A_Q \phi = \mu B_Q \phi$$

genügen. Durch Nachrechnen überzeugt man sich davon, dass dieses von allen Vektoren  $\phi_1 \in \text{Span}_{\mathbb{C}}\{e_1\}$  mit dem Eigenwert  $\mu_1 = 0$  sowie allen Vektoren  $\phi_2 \in \text{Span}_{\mathbb{C}}\{-e_1 + e_2\}$  mit dem Eigenwert  $\mu_2 = 2$  gelöst wird.

Die soeben ermittelten Eigenwerte stimmen also schon mit den Eigenwerten unseres betrachteten Problems überein. Wählen wir schließlich die Matrix

$$\Phi_2 := \begin{bmatrix} \phi_1 & \phi_2 \end{bmatrix} := \begin{bmatrix} -\iota/\sqrt{2} & \iota/\sqrt{2} \\ 0 & -\iota/\sqrt{2} \end{bmatrix},$$

<sup>6</sup>Es mag den Leser verwundern, wie bei Unkenntnis über die Anzahl  $k \in \mathbb{N}$  der zu findenden Eigenvektoren die Spaltenzahl von  $Y_k$  korrekt zu ermitteln ist. Eine gängige Methode ist zum Beispiel das Konstruieren einer passenden Filterfunktion, welche die nicht erwünschten Eigenwerte aussortiert (Vgl. Abschnitt 4.6.2).



mit den Eigenvektoren  $\phi_1$  und  $\phi_2$ , so erhalten wir wie gewünscht

$$Q\Phi_2 = \begin{bmatrix} -2 & -2 \\ 0 & -2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\iota/\sqrt{2} & \iota/\sqrt{2} \\ 0 & -\iota/\sqrt{2} \end{bmatrix} = \begin{bmatrix} \sqrt{2}\iota & 0 \\ 0 & \sqrt{2}\iota \\ 0 & 0 \end{bmatrix} = X_2.$$

**Bemerkung.** An dieser Stelle sei darauf hingewiesen, dass die Matrix  $\Phi_2$  aus kosmetischen Gründen gerade so gewählt wurde, dass die Kalkulation am Ende  $X_2$  ergibt und somit alles hübsch anzusehen ist. Der Wahl einer anderen Matrix  $\Phi'_2$  mit skalaren Vielfachen der hier gewählten Eigenvektoren steht selbstverständlich nichts im Wege und führt freilich zu einer anderen Matrix  $X'_2$ , deren Spalten ebenfalls zulässige Eigenvektoren enthält.

## 4.6 Realisierung des FEAST-Algorithmus'

Nun, da die mathematischen Ideen diskutiert und durch ein Beispiel verifiziert worden sind, widmen wir uns der Implementation des FEAST-Algorithmus' in MATLAB. Wir orientieren uns dabei an der Vorgehensweise aus dem Beispiel in Abschnitt 4.5. Auch die Notation wird beibehalten.

### 4.6.1 Naive Implementation

Zunächst wählen wir also für  $k \in \mathbb{N}$  eine vollrangige Zufallsmatrix  $Y_k$ , um die Transformationsmatrix  $Q$  zu ermitteln. Dazu führen wir die GREEN-Funktion

$$\text{funGreen} = @(w) \text{inv}(w.*B-A)$$

ein und wählen eine passende Integrationskontur  $C$ . Um es einfach zu halten, wird diese in der Implementation die Menge

$$C = \left\{ z \in \mathbb{C} : \left\| z - \frac{\lambda_{\max} - \lambda_{\min}}{2} \right\|_1 = 1 \right\}$$

sein, wobei  $\lambda_{\min}$  und  $\lambda_{\max}$  die vorgegebenen Intervallgrenzen bezeichnen. Die entsprechenden Quadraturpunkte werden im Quellcode durch die Variablen

$$n0, n1, n2, n3$$

repräsentiert.

Ist  $Q$  bestimmt, benutzen wir die MATLAB interne Eigenwertfunktion

$$\text{eig}(A_Q, B_Q)$$

um die RITZ-Paare des transformierten Problems zu berechnen. Schließlich berechnen wir die Eigenwerte und Eigenvektoren des ursprünglichen Problems. Die Matrix  $\Phi_k$  wird im Quelltext mit  $v_Q$  bezeichnet.

Wir wollen annehmen, dass der Input des Users die Voraussetzungen des Problems (4.1) erfüllt und die Anzahl  $k \in \mathbb{N}$  der Eigenwerte im Intervall  $[\lambda_{\min}, \lambda_{\max}]$  bekannt ist. Eine naive Implementation könnte daher wie folgt aussehen:

```

1 %% Eingabe: Matrizen A, B des Eigenwertproblems,
2 %%           Intervallgrenzen lmin, lmax,
3 %%           Anzahl k der erwarteten Eigenwerte
4 %% Ausgabe: Matrix V mit Eigenvektoren,
5 %%           Matrix D mit Eigenwerten auf der Diagonalen
6
7 %% Funktionsaufruf: [V, D] = feast[A, B, lmin, lmax, k]
8
9 function[V, D] = feast(A, B, lmin, lmax, k);
10
11 n = length(A);
12
13 % erzeuge Zufallsmatrix
14 Y = randn(n, k);
15
16 % generiere Green-Funktion
17 funGreen = @(w) inv( w.*B - A );
18
19 % berechne die Integrationsknotenpunkte
20 n0 = lmax;
21 n1 = (lmin + lmax)/2 + (lmax - (lmin + lmax)/2)*i;
22 n2 = lmin;
23 n3 = (lmin + lmax)/2 - (lmax - (lmin + lmax)/2)*i;
24
25 % Integrationskontur
26 C = [n1 n2 n3];
27
28 % berechne Transformationsmatrix Q
29 Q = (-1/(2*pi*i))*integral(funGreen, n0, n0,...
30     'Waypoints', C, 'ArrayValued', true) * Y;
31
32 % berechne Ritz-Paare des transformierten Problems

```

```

33 Aq = Q'*A*Q; Bq = Q'*B*Q;
34 [Vq, D] = eig(Aq, Bq);
35
36 % berechne die gesuchten Eigenvektoren
37 V = Q*Vq;

```

Algorithmus 4.1: Naive FEAST Implementation.

Überprüfen wir die Funktionalität mit den Matrizen  $A$  und  $B$  aus dem Abschnitt 4.5, erhalten wir folgende Ausgabe:

```

>> [V, D] = feast(A, B, -3, 3, 2)

V =

    -1.4142 - 0.0000i    0.0000 + 0.0000i
     0.0000 + 0.0000i    1.4142 - 0.0000i
    -0.0000 + 0.0000i   -0.0000 + 0.0000i

D =

    0.0000    0
         0    2.0000

>>

```

Die vorgestellte Umsetzung des FEAST-Algorithmus' lässt sich ohne Weiteres auf dynamische Matrixdimensionen übertragen. Da sich mit zunehmender Dimension unter Umständen auch die Kardinalität des Spektrums und die Anzahl der Eigenräume ändert, sollten gegebenenfalls Optimierungsmaßnahmen getroffen werden, wie etwa die Wahl der Knotenpunkte für die Konturintegration.

Man könnte sich nun die Frage stellen, warum man anstatt (4.1) nicht einfach das Problem

$$B^{-1}Ax = \lambda x$$

zu lösen versucht. Der positiven Definitheit von  $B$  wegen, wäre diese Umformulierung möglich und mit bekömmlicheren Verfahren lösbar.

Nun ist aber das Invertieren von Matrizen bekanntermaßen numerisch eher ungünstig, da man unter Umständen mit hohem Rechenaufwand und dem Verlust von Stabilität bezahlen muss. Ein erneuter Blick auf (4.3) oder (4.4) stimmt demnach

benklich, da sowohl mit der Integration als auch mit dem Berechnen der GREEN-Funktion gewisse Risiken eingegangen werden.

Man muss sich also die Frage stellen, ob eine Implementation des FEAST-Algorithmus ohne Inversion und Integration möglich ist.

### 4.6.2 Rational Krylov Toolbox

Wie in der Einleitung versprochen, soll abschließend ein numerischer Werkzeugkoffer präsentiert werden, der es ermöglicht den FEAST-Algorithmus elegant und ohne die eben beschriebenen Bedenklichkeiten umzusetzen.

Mit der „*Rational Krylov Toolbox for MATLAB*“ [Ber16] stellen deren Entwickler MARIO BERLJAFÄ und STEFAN GÜTTEL eine Bibliothek zur Verfügung, die es ermöglicht, gängige KRYLOV-Unterraum-Verfahren unkompliziert zu implementieren. Bevor wir dieses Tool genauer betrachten, wollen wir uns klarmachen, warum die Anwendung eines solchen Verfahrens möglich ist. Dazu begnügen wir uns an dieser Stelle mit einer Plausibilitätsbetrachtung.

Studieren wir erneut die in (4.3) formulierte Identität

$$X_k X_k^T := \sum_{i=1}^k x_i x_i^T = -\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega,$$

fällt sodann auf, dass die Abbildung

$$P: \mathbb{C}^n \rightarrow \mathbb{C}^n, v \mapsto Pv := \left( -\frac{1}{2\pi\iota} \int_{\gamma} G(\omega) \, d\omega \right) v$$

Vektoren in den von den Eigenvektoren  $(x_i)_{1:k}$  aufgespannten Unterraum abbildet. Man kann sogar zeigen, dass die Abbildung eine  $B$ -orthogonale Projektion ist. Die Idee, den FEAST-Algorithmus per Projektionsverfahren umzusetzen, liegt demnach gar nicht so fern.

Eine mögliche Implementation stellen die Autoren in ihren Beispielen auf der Seite

[http://www.guettel.com/rktoolbox/examples/html/example\\_feast.html](http://www.guettel.com/rktoolbox/examples/html/example_feast.html)

VOR:

```
1 % Search space basis V of dimension m.
2 m = 10;
```

```

3  V = randn(N, m);
4
5  for iter = 1:8
6      % Apply rational filter to V.
7      V = r(A, V);
8      % Compute and sort Ritz pairs.
9      Am = V'*A*V; Bm = V'*B*V;
10     [W, D] = eig(Am, Bm);
11     [D, ind] = sort(diag(D)); W = W(:, ind);
12     % B-normalize W.
13     nrm = sqrt(diag(W'*Bm*W)); W = W/diag(nrm);
14     % Form approximate eigenvectors.
15     V = V*W;
16     % Check residuals and number of eigenpairs inside
17     % search interval.
18     Di = diag(D(lmin < D & D < lmax));
19     Vi = V(:, lmin < D & D < lmax);
20     resid(iter) = norm(A*Vi - B*Vi*Di, 'fro');
21     nrvec(iter) = size(Vi, 2);
22 end
    
```

Algorithmus 4.2: FEAST-Implementation mit Unterraumprojektion.

Diese beginnt mit der Initialisierung und dem Festlegen eines Suchraums, der durch die Spalten der Matrix  $V$  aufgespannt wird. Hierbei ist  $N$  die Dimension der Matrizen  $A$  und  $B$  und  $m$  die Anzahl der erwarteten Eigenpaare.<sup>7</sup>

In der folgenden Iterationsschleife wird der Eigenraum approximiert. Besonderes Augenmerk muss auf die Filterfunktion  $r$  gelegt werden. Diese ist gerade so konzipiert, dass Argumente innerhalb des Suchintervalls in einen Bereich  $(1 - \varepsilon, 1 + \varepsilon)$  abgebildet werden. Der Wert  $\varepsilon > 0$  ist hierbei vom Input abhängig und kann stark variieren. Argumente außerhalb des Intervalls werden in ähnlicher Manier auf einen Bereich  $(-\delta, \delta)$  abgebildet – für ein entsprechendes  $\delta > 0$ .

Der Toolbox-interne Aufruf  $r(A, V)$  ermöglicht nun das Berechnen der Wirkung des Filters auf den Suchraum  $V$ .

Schlussendlich löst der Algorithmus auch hier das transformierte Eigenwertproblem und ermittelt daraus approximativ die gesuchten Eigenpaare.

<sup>7</sup>Auch in diesem Beispiel ist diese Anzahl der Eigenpaare im Voraus bekannt. Trotzdem wird ein Filter eingesetzt.

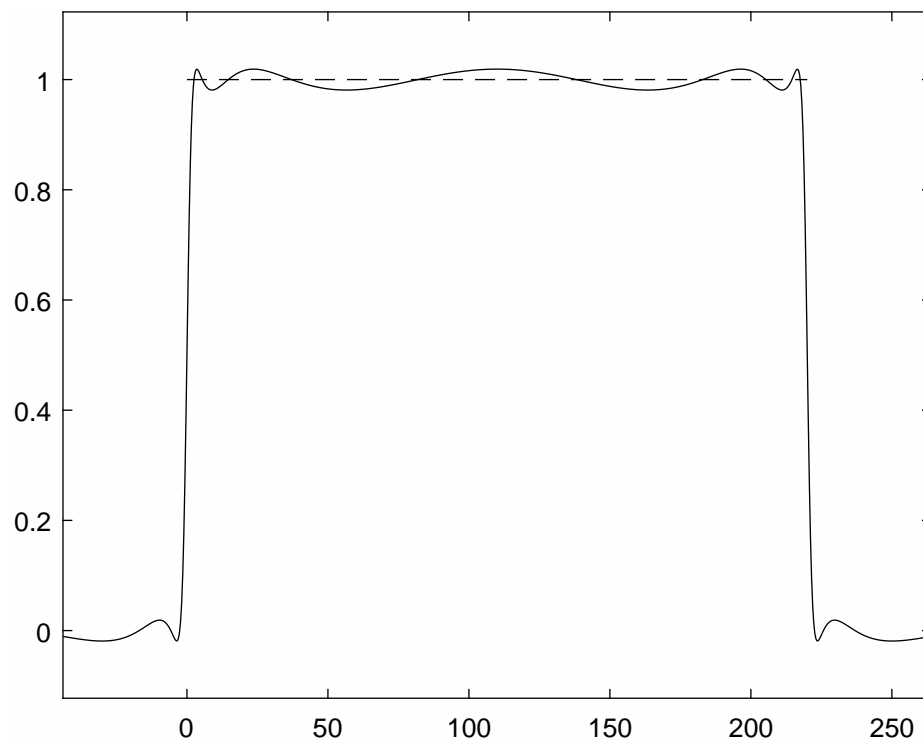


Abbildung 4.2: Filterfunktion  $r$  auf dem Intervall  $[0, 220]$  wirkend.

# Kapitel 5

Die folgende Proposition weist nach, dass der eben vorgestellte Algorithmus tatsächlich ein Projektionsverfahren ist.

**Proposition 5.1.** Die durch die Matrix  $V_m V_m^H \in \mathbb{C}^{n,n}$  induzierte lineare Abbildung

$$P: \mathbb{C}^n \rightarrow \mathbb{C}^n, x \mapsto V_m V_m^H x$$

ist eine orthogonale Projektion auf den Unterraum  $V$  und löst das Minimierungsproblem

$$\min_{y \in V} \|x - y\|, x \in \mathbb{C}^n (\|x - P(x)\| = \dots)$$

*Beweis.* Sei  $x \in \mathbb{C}^n$  vorgegeben. Dann gilt  $P(x) = V_m V_m^H x \in \text{Bild}(V_m) = V$  nach Konstruktion. Da aus der Orthogonalität der Spalten von  $V_m$

$$P^2 = V_m V_m^H V_m V_m^H = V_m V_m^H = P$$

folgt, ist  $P$  somit eine Projektion auf  $V$ . Darüber hinaus gilt für alle Vektoren  $y \in V$

$$\langle y, x - P(x) \rangle = y^H (x - V_m V_m^H x) = y^H x - P(y)^H x = 0$$

Also ist  $x - P(x) \in V^\perp$  und  $P$  tatsächlich eine orthogonale Projektion. Wegen  $P(x) - y \in V$  folgt die Minimalität aus der Abschätzung

$$\begin{aligned} \|x - y\|^2 &= \|x - P(x) + P(x) - y\|^2 \\ &= \langle x - P(x) + P(x) - y, x - P(x) + P(x) - y \rangle \\ &= \|x - P(x)\|^2 + \langle x - P(x), P(x) - y \rangle + \langle P(x) - y, x - P(x) \rangle + \|P(x) - y\|^2 \\ &= \|x - P(x)\|^2 + \|P(x) - y\|^2 \\ &\geq \|x - P(x)\|^2. \end{aligned}$$

Damit ist die Behauptung bewiesen, da Gleichheit nur mit  $y = P(x)$  folgen kann.  $\square$

Wir wollen nun die eben erarbeitete Theorie auf das verallgemeinerte Eigenwertpro-

blem

$$Ax = \lambda Bx$$

übertragen.



# Anhang A

## yolo

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

# Literatur

- [Bar15] Marc van Barel. “Designing rational filter functions for solving eigenvalue problems by contour integration.” In: (2015).
- [Ber16] Mario Berljafa. “Rational Krylov Toolbox for MATLAB.” In: (2016).
- [Lie13] Jrg Liesen. *Krylov subspace methods. Principles and analysis*. 2013.
- [Pol09] Eric Polizzi. “Density-matrix-based algorithm for solving eigenvalue problems.” In: (2009).
- [Ste13] Gilbert W. Stewart. *Matrix algorithms*. Bd. 2: Eigensystems. 2013.
- [TP16] Ping Tak Peter Tang und Eric Polizzi. “Feast as a subspace iteration eigensolver accelerated by approximate spectral projection.” In: (2016).