

Ein Einschließungsverfahren
für das
inverse Dirichletproblem

Zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften

von der Fakultät für Mathematik der
Universität Karlsruhe
genehmigte

Dissertation

von
Dipl.-Math. techn. Markus Neher
aus Freiburg

Tag der mündlichen Prüfung:
Referent:
Korreferent:

30. Juni 1993
Prof. Dr. G. Alefeld
Prof. Dr. G. Mayer

An dieser Stelle möchte ich allen herzlich danken, die mich bei der Entstehung der vorliegenden Arbeit unterstützten. Ein besonderer Dank geht dabei an Herrn Prof. Dr. G. Alefeld, der das Thema der Arbeit angeregt und mir zahlreiche wertvolle Ratschläge und Hinweise für Verbesserungen gegeben hat. Herrn Prof. Dr. G. Mayer danke ich für die freundliche Übernahme des Korreferats. Durch seine ständige Gesprächsbereitschaft hat auch er viel zu einem tieferen Verständnis der Intervallrechnung und der Einschließungsverfahren beigetragen.

Für zuverlässige Informationen über PASCAL-XSC und $\text{T}_{\text{E}}\text{X}$ nische Hilfen bedanke ich mich bei Herrn Dr. R. Hammer und Herrn Dr. D. Ratz, für das aufmerksame Korrekturlesen des Manuskripts bei Frau A. C. Jooss und Herrn M. Fauser.

Schließlich danke ich meiner Frau Helga, die manches Wochenende als Alleinerziehende unserer Kinder verbracht und damit wesentlich zum Gelingen dieser Arbeit beigetragen hat.

Inhaltsverzeichnis

Bezeichnungen	3
Einleitung	5
1 Das Dirichletproblem	11
1.1 Ein Monotoniesatz	12
1.2 Asymptotische Entwicklungen	14
1.3 Das inverse Dirichletproblem	16
2 Ein Rekonstruktionsverfahren für das inverse Dirichletproblem	20
2.1 Differentialrechnung in Banachräumen	22
2.2 Ableitungen der Eigenwerte	25
2.3 Das Rekonstruktionsverfahren	27
2.4 Addition von Konstanten	28
2.5 Abbruch des Rekonstruktionsverfahrens	30
2.6 Der Satz von Kantorowitsch	32
2.7 Abschätzung der zweiten Ableitung	34
2.8 Explizite Schranken der Eigenfunktionen	39
2.9 Konvergenz	45
2.10 Intervallrechnung	50

2.11	Das Intervall-Newtonverfahren	53
2.12	Rekonstruktion mit dem Intervall-Newtonverfahren	56
3	Numerische Realisierung	58
3.1	Näherungslösungen mit Hilfe des Ritz-Verfahrens	59
3.2	Näherungslösungen durch Lösen von Anfangswertproblemen	62
3.3	Lösungseinschließung auf dem Computer	63
3.3.1	PASCAL-XSC	64
3.3.2	Garantierte Lösung von Anfangswertproblemen	65
3.4	Implementierung des Rekonstruktionsverfahrens	66
3.4.1	Eigenwertschranken auf dem Computer	66
3.4.2	Einschließung von Eigenfunktionen	68
3.4.3	Einschließung der partiellen Ableitungen	70
3.4.4	Das Maschinen-Intervall-Newtonverfahren	70
3.4.5	Der vollständige Algorithmus	74
4	Numerische Beispiele	77
4.1	Beispiele zum direkten Dirichletproblem	78
4.2	Beispiele zum inversen Dirichletproblem	87
	Programmdokumentation	96
	Literaturverzeichnis	115

Bezeichnungen

\mathbb{N}	Menge der natürlichen Zahlen
\mathbb{N}_0	$\mathbb{N} \cup \{0\}$
\mathbb{Z}	Menge der ganzen Zahlen
$\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n \times n}$	Menge der reellen Zahlen, n -dimensionalen reellen Vektoren, reellen $n \times n$ -Matrizen
\mathbb{C}	Menge der komplexen Zahlen
$I\mathbb{R}, I\mathbb{R}^n, I\mathbb{R}^{n \times n}$	Menge der reellen Intervalle, n -dimensionalen Intervallvektoren, $n \times n$ -Intervallmatrizen
$C^0, C^0[0, \pi]$	Raum der (auf $[0, \pi]$) stetigen Funktionen mit Maximumsnorm $\ \cdot\ _\infty$
$L^2, L^2[0, \pi]$	Raum der reellwertigen, (auf $[0, \pi]$) quadratisch integrierbaren Funktionen
$L^2_{\mathbb{C}}[0, \pi]$	Raum der komplexwertigen, auf $[0, \pi]$ quadratisch integrierbaren Funktionen
$\mathcal{L}(E, F)$	Raum der stetigen linearen Abbildungen des Banachraums E in den Banachraum F
$\mathcal{L}(E^p, F)$	Raum der stetigen p -linearen Abbildungen des Produktraums $E \times \dots \times E$ des Banachraums E in den Banachraum F
$\{a_n\}_{n=0}^\infty$	Reelle Folge
$A = (a_{ij})$	Reelle Matrix

A^{-1}	Inverse Matrix
$[a] = [\underline{a}, \bar{a}], [A]$	Reelles Intervall mit Untergrenze \underline{a} und Obergrenze \bar{a} , Intervallmatrix $[A]$
$f([x])$	Intervallauswertung der Funktion $f(x)$ auf dem Intervall $[x]$
$f'(x^0) = d_x f(x^0)$	Ableitung einer Funktion f eines Banachraums E in den Banachraum F an der Stelle x^0
$\partial_{x^i} f(x_1^0, \dots, x_n^0)$	Partielle Ableitung von f (nach der i -ten Komponente) an der Stelle x^0

Funktionen, Intervalle und Intervallvektoren schreiben wir mit kleinen, Matrizen und Intervallmatrizen mit großen Buchstaben.

Die Komponenten von Vektoren und Matrizen indizieren wir unten (x_i, a_{ij}), bei einer Menge von Vektoren werden die Vektoren oben indiziert (z.B. x^1, x^2, \dots). Den Iterationsindex eines Iterationsverfahrens schreiben wir oben und in Klammern: $x^{(k)}$.

Anmerkung zur elektronischen Version

Die elektronische Version unterscheidet sich von der ursprünglich gedruckten Arbeit durch die Korrektur von drei Tippfehlern. Außerdem wurde eine modernere L^AT_EX-Version verwendet, wodurch sich einige Seitenumbrüche geringfügig verschoben haben.

Einleitung

Mechanische Probleme, die mit Schwingungsvorgängen verbunden sind, führen auf sogenannte *Eigenwertaufgaben*, bei denen ein Parameter auftritt, der als *Eigenwert* bezeichnet wird und der das physikalische Verhalten des zugrundeliegenden technischen Systems entscheidend beeinflusst. Häufig führt die mathematische Analyse auf das nach den französischen Mathematikern Ch. Sturm und J. Liouville benannte *Sturm-Liouvillesche Eigenwertproblem*

$$\begin{aligned} -u''(x) + q(x)u(x) &= \lambda u(x), & x \in [0, \pi] \\ \cos \alpha u(0) + \sin \alpha u'(0) &= 0, & \alpha \in [0, \pi) \\ \cos \beta u(\pi) + \sin \beta u'(\pi) &= 0, & \beta \in (0, \pi]. \end{aligned} \quad (0.1)$$

Von besonderem Interesse ist der Fall $\alpha = 0$, $\beta = \pi$, also das Eigenwertproblem

$$\begin{aligned} -u''(x) + q(x)u(x) &= \lambda u(x), & x \in [0, \pi] \\ u(0) &= u(\pi) = 0. \end{aligned} \quad (0.2)$$

(0.2) bezeichnen wir mit Pöschel und Trubowitz [39] als *Dirichletproblem*, die Randbedingungen werden *Dirichletsche Randbedingungen* genannt.

Die klassische Lösungstheorie sucht für $q(x) \in C^0[0, \pi]$, dem Raum der auf dem Intervall $[0, \pi]$ stetigen Funktionen, eine auf $[0, \pi]$ zweimal stetig differenzierbare, nichttriviale Funktion $u(x)$, die (0.1) erfüllt. Wir werden nicht nur solche Lösungen untersuchen, sondern auch den Fall $q(x) \in L^2[0, \pi]$, dem Raum der über $[0, \pi]$ quadratisch integrierbaren Funktionen, behandeln. Wir verstehen dann unter einer Lösung von (0.1) eine nichttriviale, stetig differenzierbare Funktion $u(x)$, die eine absolutstetige Ableitung besitzt, die vorgeschriebenen Randwerte annimmt und die Differentialgleichung fast überall erfüllt.

Im folgenden unterscheiden wir die *direkte* und die *inverse* Problemstellung. Bei der direkten Eigenwertaufgabe sind in (0.1) die (stetige oder quadratisch integrierbare) reellwertige Funktion $q(x)$ und die reellen Zahlen α und β , die die Randbedingungen bestimmen, bekannt. Gesucht sind solche Werte von λ , für die das Randwertproblem (0.1) nichttriviale Lösungen besitzt. Derartige Zahlen heißen *Eigenwerte*, die zugehörigen nichttrivialen Lösungen werden als *Eigenfunktionen* bezeichnet. Die Menge aller Eigenwerte wird *Spektrum* von (0.1) genannt. Da jede stetige oder quadratisch integrierbare Funktion $q(x)$ bei vorgegebenen Randbedingungen durch (0.1) ein Eigenwertproblem definiert, sprechen wir auch von den *Eigenwerten von $q(x)$* bzw. dem *Spektrum von $q(x)$* .

Im Gegensatz dazu wird bei der inversen Aufgabenstellung eine Funktion $q(x)$ gesucht, so daß (0.1) vorgegebene Zahlen als Eigenwerte besitzt. Die Randbedingungen, d.h. die Parameter α und β , können bekannt sein, oder sie sind ebenfalls aus den vorgeschriebenen Eigenwerten zu bestimmen. Mit Blick auf die Anwendungen wird $q(x)$ als *Potential* bezeichnet (siehe Hald [18]).

Bereits Sturm [46] und Liouville [28] haben 1836 im gleichen Band des Journal de Mathématique untersucht, für welche λ nichttriviale Lösungen des Randwertproblems (0.1) existieren. Seither sind diese Ergebnisse in einer Vielzahl von Arbeiten erweitert worden.

Wir formulieren die für diese Arbeit wichtigsten Aussagen über das Dirichletproblem (Walter [47], Pöschel und Trubowitz [39]):

1. Das Eigenwertproblem (0.2) besitzt für $q(x) \in L^2[0, \pi]$ abzählbar unendlich viele reelle Eigenwerte λ_n , $n \in \mathbb{N}_0$. Die λ_n genügen der asymptotischen Entwicklung

$$\lambda_n = (n+1)^2 + \frac{1}{\pi} \int_0^\pi q(x) dx + a_n, \quad (0.3)$$

wobei $\{a_n\}_{n=0}^\infty$ eine quadratisch summierbare Folge ist.

2. Die (in $L^2[0, \pi]$ normierte) n -te Eigenfunktion $g_n(x)$ mit $g'_n(0) > 0$ besitzt im Intervall $(0, \pi)$ genau n Nullstellen. Es gilt für $n \rightarrow \infty$

$$g_n(x) = \sqrt{\frac{2}{\pi}} \sin(n+1)x + O\left(\frac{1}{n}\right)$$

gleichmäßig auf $[0, \pi]$.

Die grundlegende Arbeit über das inverse Problem stammt von Borg [10] aus dem Jahr 1946 (ausgehend von einer Fragestellung von Ambarzumian [7], der in einer 1929 erschienenen Arbeit die Bedeutung des inversen Problems für die Physik erwähnt). Borg zeigt zur Eindeutigkeit im wesentlichen, daß ein zur Intervallmitte symmetrisches Potential $q(x)$ durch *ein* Spektrum, ein unsymmetrisches durch Vorgabe *zweier* Spektren zu unterschiedlichen Randbedingungen eindeutig bestimmt ist.

Borg formuliert bereits einen ersten Existenzsatz. Verbesserte Aussagen finden sich bei Gel'fand und Levitan [15] sowie Levitan [27]. Als notwendig und hinreichend für die Existenz eines Potentials erweisen sich geeignete asymptotische Entwicklungen der vorgegebenen Eigenwerte.

Pöschel und Trubowitz [39] behandeln das inverse Dirichletproblem und geben die folgende Charakterisierung zulässiger Spektren an:

Zu einer Folge $\Lambda = \{\lambda_n\}_{n=0}^{\infty}$ existiert genau dann eine Funktion $q(x) \in L^2[0, \pi]$, so daß das zugehörige Dirichletproblem das Spektrum Λ besitzt, wenn die Folge streng monoton wächst und es eine Zahl $s \in \mathbb{R}$ und eine quadratisch summierbare Folge $\{a_n\}_{n=0}^{\infty}$ gibt, für die

$$\lambda_n = (n+1)^2 + s + a_n, \quad n = 0, 1, \dots$$

gilt.

Zu Existenz- und Eindeutigkeitsaussagen stehen durch die angesprochenen Resultate notwendige und hinreichende Kriterien zur Verfügung. Zu lösen bleibt aber noch das Problem der Rekonstruktion von $q(x)$ aus gegebenen Eigenwerten. Diese Aufgabe ist nicht zuletzt deshalb schwierig, weil die vorzugebenden Eingangsdaten wesentlich von den Randbedingungen und den Symmetrieeigenschaften von $q(x)$ abhängen.

Gel'fand und Levitan entwickeln 1951 in [15] den ersten Rekonstruktionsalgorithmus. Statt zweier Spektren geben sie ein einzelnes Spektrum und Normierungskonstanten der Eigenfunktionen vor (diese Daten können gemessen werden und bestimmen das Potential ebenfalls eindeutig). Durch Lösen der Integralgleichung

$$f(x, s) + \int_0^x K(x, t) f(t, s) dt + K(x, s) = 0, \quad 0 \leq s \leq x,$$

nach $K(x, s)$ erhält man $q(x)$ aus der Beziehung $q(x) = 2 \frac{d}{dx} K(x, x)$. Die Funktion $f(x, t)$ in der Integralgleichung ist durch die vorgegebenen Daten bestimmt.

Die größte Schwierigkeit dieser Methode liegt in der Abhängigkeit der Lösung von $f(x, t)$. f ist die konvergente Differenz zweier Reihen, die beide schnell divergieren. Sind nur endlich viele Eigenwerte $\{\lambda_j\}_{j=0}^n$ vorgegeben, tritt dieses Problem nicht auf, da $f(x, t)$ in diesem Fall durch eine endliche Summe dargestellt wird.

Die Eigenfunktionen $u_j(x)$ der unbekanntem Funktion $q(x)$ werden aus den Eigenfunktionen $v_j(x)$ eines Vergleichspotentials $p(x)$ durch die Beziehung

$$u_j(x) = v_j(x) + \int_0^x K(x, s) v_j(s) ds$$

bestimmt. Gibt man nur endlich viele Eigenwerte vor, werden $q(x)$ die höheren Eigenwerte von $p(x)$ aufgezwungen, und somit alle Eigenschaften von $p(x)$, die in den asymptotischen Entwicklungen der Eigenwerte verborgen sind. Außerdem werden neben dem Potential auch die Randbedingungen durch das Rekonstruktionsverfahren festgelegt. Damit ist das Verfahren nur bedingt anwendbar, wenn die Randbedingungen nicht frei wählbar sind, sondern durch das zugrundeliegende physikalische System bestimmt werden. Trotz dieser Nachteile war der Rekonstruktionsalgorithmus von Gel'fand und Levitan Ausgangspunkt für eine Reihe anderer Verfahren (siehe Gladwell [16], McLaughlin [33], Rundell und Sacks [41] sowie Sacks [42]).

Hochstadt [22] und Hald [18] konstruieren $q(x)$ durch Lösen eines Differentialgleichungssystems. Auch sie benötigen ein Vergleichspotential, von dem eventuell unerwünschte Eigenschaften übernommen werden.

Pöschel und Trubowitz [39] untersuchen, wie sich das Potential des Dirichletproblems bei Störungen eines Eigenwerts verändert. Damit ist ein Homotopieverfahren zur Lösung des inversen Problems definiert. Die rekonstruierte Funktion $q(x)$ hängt auch bei diesem Vorgehen vom Ausgangspotential ab.

Marti [32] löst das inverse Problem durch Diskretisieren der Differentialgleichung mit der Methode der finiten Elemente. Dieser Ansatz führt auf ein inverses Eigenwertproblem für Matrizen, wodurch sich der numerische Aufwand erheblich reduziert. Hald [19] entwickelt $q(x)$ für symmetrische Probleme in eine Kosinusreihe und gelangt mit Hilfe des Ritz-Verfahrens zu einem inversen Eigenwertproblem für Matrizen. Da die Eigenwerte der Matrizenprobleme aber nicht mit den Eigenwerten der Randwertprobleme übereinstimmen, führen diese Verfahren nur bei Potentialen mit kleiner Norm zu guten Ergebnissen. Barcilon berechnet in [8] iterativ die Fourierkoeffizienten einer Kosinusreihe und erhält ebenfalls ein konvergentes Verfahren, wenn die Norm des gesuchten Potentials hinreichend klein ist.

Barnes [9] konstruiert Lösungen des inversen Sturm-Liouvilleschen Eigenwertproblems als Linearkombination

$$q(x) = \sum_{j=0}^n a_j q_j(x) \quad (0.4)$$

von vorgegebenen Funktionen $q_j(x)$ durch Lösen von Minimierungsaufgaben mit der Methode der kleinsten Quadrate. Lowe, Pilant und Rundell [31] haben mit dem gleichen Ansatz für das Potential ein Newtonverfahren entwickelt, bei dem in jedem Iterationsschritt gewöhnliche Anfangswertprobleme für die Differentialgleichung aus (0.1) zu lösen sind. Das Newtonverfahren konvergiert bei hinreichend kleiner Norm des gesuchten Potentials.

Alle uns bekannten Rekonstruktionsverfahren sind im praktischen Einsatz auf dem Computer Näherungsverfahren, bei denen das Ergebnis der Rekonstruktion durch Diskretisierungs- und Rundungsfehler oder den Abbruch eines Iterationsprozesses verfälscht wird. In der Regel kann man bei diesen Verfahren keine Aussage darüber machen, wie weit eine berechnete Näherungslösung von der gesuchten Lösung entfernt ist.

Das Ziel dieser Arbeit ist daher in erster Linie ein Algorithmus, der zu vorgegebenen Eigenwerten nicht nur ein Näherungspotential berechnet, sondern auch eine auf dem Computer erzielbare Lösungseinschließung liefert. Wir konstruieren dazu einen Funktionenschlauch, der eine reellwertige Funktion $q(x)$ einschließt, die eine exakte Lösung des gestellten Problems ist.

Damit das entwickelte Verfahren in der Praxis anwendbar ist, verwenden wir nur Eingabedaten, die sich in endlicher Zeit messen lassen, und beschränken uns daher auf die Vorgabe endlich vieler Eigenwerte. Außerdem betrachten wir bei der Rekonstruktion nur symmetrische Potentiale und Dirichletsche Randbedingungen. Für diese im folgenden als *endliches inverses Dirichletproblem* bezeichnete Aufgabenstellung bestimmen wir ebenso wie Barnes [9] sowie Lowe, Pilant und Rundell [31] Lösungen in einem vorgegebenen endlichdimensionalen Funktionenraum.

Wir erweitern (0.4) um eine zusätzliche Funktion $\hat{q}(x)$, und suchen ein Potential von der Form

$$q(x) = \hat{q}(x) + \sum_{j=0}^n a_j q_j(x) \quad (0.5)$$

mit frei wählbaren Funktionen $\hat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$. Dieser Ansatz führt zum Problem, die Koeffizienten a_j , $j = 0, 1, \dots, n$, so zu bestimmen, daß $q(x)$ eine Lösung des gestellten inversen Eigenwertproblems ist.

Das entstehende nichtlineare Gleichungssystem für die a_j lösen wir mit dem Newtonverfahren im \mathbb{R}^{n+1} , legen bei der Newton-Iteration allerdings eine andere Funktion zugrunde als Lowe, Pilant und Rundell [31]. Mit Hilfe der zusätzlich aufgenommenen Funktion $\widehat{q}(x)$ werden wir wahlweise über den Satz von Kantorowitsch oder über das Intervall-Newtonverfahren zu lokalen Existenz-, Eindeutigkeits- und Konvergenzaussagen gelangen, die auch für Potentiale mit großer Norm anwendbar sind.

Bevor wir unseren Rekonstruktionsalgorithmus zur Lösung des inversen Dirichletproblems vorstellen können, müssen wir einige Vorbereitungen treffen. Im ersten Kapitel untersuchen wir das direkte und das inverse Eigenwertproblem, wobei wir im wesentlichen einen Überblick über die bekannte Theorie geben. Im Anschluß daran entwickeln wir das Rekonstruktionsverfahren selbst und führen eine Konvergenzanalyse durch.

Abschließend werden wir die numerische Implementierung beschreiben, und auf dem Computer Lösungseinschließungen für das inverse Dirichletproblem berechnen. Da wir zur Berechnung dieser Einschließungen garantierte Schranken von Eigenwerten und Eigenfunktionen beim direkten Problem benötigen, werden wir zeigen, wie sich solche Schranken auf dem Computer bestimmen lassen. Im letzten Kapitel werden wir die Anwendbarkeit der Methode an Beispielen illustrieren.

Kapitel 1

Das Dirichletproblem

In diesem Kapitel stellen wir die theoretischen Grundlagen des direkten und des inversen Dirichletproblems vor. Wir geben zunächst zwei für das allgemeine Sturm-Liouvillesche Eigenwertproblem bekannte Sätze aus dem Buch von Walter [47] an, formulieren sie aber für das Dirichletproblem. Wir setzen zunächst voraus, daß $q(x)$ stetig ist.

Satz 1.0.1 (Existenzsatz) *Das Dirichletproblem besitzt abzählbar unendlich viele reelle Eigenwerte $\lambda_0 < \lambda_1 < \dots, \lambda_n \rightarrow \infty$ für $n \rightarrow \infty$. Die zum Eigenwert λ_n gehörende Eigenfunktion $u_n(x)$ hat im offenen Intervall $(0, \pi)$ genau n Nullstellen. Zwischen je zwei Nullstellen von $u_n(x)$ liegt eine Nullstelle von $u_{n+1}(x)$.*

Beweis: Walter [47], Seite 187ff. □

Bemerkung: Da das Anfangswertproblem

$$\begin{aligned} -u'' + q(x)u &= \lambda u, & x \in [a, b] \\ u(\xi) &= 0, \quad u'(\xi) = 0, & \xi \in [a, b] \end{aligned}$$

für $q \in C^0[a, b]$ nur die triviale Lösung $u \equiv 0$ besitzt (siehe Walter [47], Seite 132), sind die Nullstellen der Eigenfunktionen u_n des Dirichletproblems einfach, d.h. aus $u_n(\xi) = 0$ für ein $\xi \in [0, \pi]$ folgt $u'_n(\xi) \neq 0$. □

Satz 1.0.2 (Trennungssatz von Sturm) *Es seien J ein beliebiges Intervall und $q(x) \in C^0(J)$. Dann gilt:*

Sind u_1, u_2 zwei linear unabhängige Lösungen von $-u'' + q(x)u = 0$, so trennen sich ihre Nullstellen gegenseitig, d.h. zwischen zwei aufeinanderfolgenden Nullstellen von u_1 liegt eine von u_2 und umgekehrt.

Ist u eine Lösung von $-u'' + q(x)u = 0$ und v eine nichttriviale Lösung von $-v'' + q_0(x)v = 0$ mit $q_0(x) \in C^0(J)$, $q(x) \leq q_0(x)$ in J , $q \not\equiv q_0$, so liegt zwischen zwei Nullstellen von v (mindestens) eine Nullstelle von u .

Beweis: Walter [47], Seite 192. □

1.1 Ein Monotoniesatz

Aus dem Trennungssatz läßt sich eine Eigenschaft des Dirichletproblems ableiten, die wir in der von uns verwendeten Literatur in dieser Form nicht explizit gefunden haben (siehe auch Walter [48]), und die wir als Monotoniesatz formulieren wollen.

Wir betrachten das mit dem Dirichletproblem verbundene Anfangswertproblem für $u = u(x, \mu)$:

$$\begin{aligned} -u'' + q(x)u &= \mu u, & x \in [0, \pi] \\ u(0) &= 0, \quad u'(\pi) = 1, \end{aligned} \tag{1.1}$$

wobei die Wahl von $u'(\pi) = 1$ hier lediglich als Normierungsbedingung zu verstehen ist, und durch einen beliebigen anderen Wert ungleich Null ersetzt werden kann.

Zu einem gegebenen Potential $q(x)$ besitzt das Anfangswertproblem (1.1) für jedes $\mu \in \mathbb{R}$ eine eindeutige Lösung $u(x, \mu)$. μ ist genau dann ein Eigenwert des Dirichletproblems, wenn $u(\pi, \mu) = 0$ erfüllt ist. Gemäß der Definition von λ_n ist $u(x, \lambda_n)$ n -te Eigenfunktion des Dirichletproblems.

Indem wir die (einfachen!) Nullstellen von $u(x, \mu)$ im Intervall $(0, \pi]$ zählen, erhalten wir obere und untere Schranken für die Eigenwerte des Dirichletproblems aus dem

Satz 1.1.1 *Sei $u(x, \mu)$ Lösung von (1.1) für ein gegebenes $\mu \in \mathbb{R}$. Weiter bezeichne $N(\mu)$ die Anzahl der Nullstellen von $u(x, \mu)$ im Intervall $(0, \pi]$.*

Dann ist $N(\mu)$ eine monoton wachsende Funktion von μ und es gilt

$$N(\mu) = n + 1 \quad \text{für } \mu \in [\lambda_n, \lambda_{n+1}), \quad n \in \mathbb{N}_0,$$

sowie

$$N(\mu) = 0 \quad \text{für } \mu < \lambda_0.$$

Beweis: Im Fall $\mu = \lambda_n$ für ein $n \geq 0$ besitzt die n -te Eigenfunktion $u(x, \lambda_n)$ nach Satz 1.0.1 genau n Nullstellen im Intervall $(0, \pi)$ und es ist $u(\pi, \lambda_n) = 0$, d.h. es gilt $N(\lambda_n) = n + 1$.

Seien nun $\lambda_0 \leq \mu_1 < \mu_2$, $u_1(x) := u(x, \mu_1)$, $u_2(x) := u(x, \mu_2)$ die zugehörigen Lösungen von (1.1). Wegen $q(x) - \mu_2 < q(x) - \mu_1$ liegt nach Satz 1.0.2 zwischen je zwei Nullstellen von u_1 im Intervall $(0, \pi]$ (mindestens) eine Nullstelle von u_2 . Aus $u_1(0) = u_2(0) = 0$ folgt daher $N(\mu_2) \geq N(\mu_1)$.

Für $\mu < \lambda_0$ kann $u(x, \mu)$ keine Nullstelle in $(0, \pi]$ besitzen, da wegen $q(x) - \lambda_0 < q(x) - \mu$ in diesem Fall die Funktion $u(x, \lambda_0)$ nach Satz 1.0.2 eine Nullstelle in $(0, \pi)$ besitzen müßte. \square

Korollar 1.1.2 a) Für $\underline{\lambda}$ mit $N(\underline{\lambda}) \leq n$ gilt $\underline{\lambda} < \lambda_n$.

b) Für $\bar{\lambda}$ mit $N(\bar{\lambda}) \geq n + 1$ gilt $\lambda_n \leq \bar{\lambda}$.

Für eine im Intervall $[0, \pi]$ stetige Funktion $q(x)$ bezeichne $\lambda_n(q)$ im folgenden den n -ten Eigenwert des Dirichletproblems mit Potential $q(x)$. Dann gelten:

Lemma 1.1.3 Die Eigenwerte des Dirichletproblems ändern sich bei der Addition der konstanten Funktion c zum stetigen Potential $q(x)$ ebenfalls um c , d.h. es gilt für alle $n \in \mathbb{N}_0$

$$\lambda_n(q + c) = \lambda_n(q) + c.$$

Beweis: $-u'' + q(x)u = \lambda u \Leftrightarrow -u'' + (q(x) + c)u = (\lambda + c)u.$ \square

Satz 1.1.4 (Monotoniesatz) Die Eigenwerte des Dirichletproblems hängen monoton von $q(x)$ ab, d.h. aus

$$p(x) \leq q(x) \quad \text{für alle } x \in [0, \pi]$$

folgt

$$\lambda_n(p) \leq \lambda_n(q) \quad \text{für alle } n \in \mathbb{N}_0.$$

Beweis: Es sei $\lambda_n = \lambda_n(q)$ mit Eigenfunktion $u_n(x)$. $v(x)$ sei Lösung des Anfangswertproblems (1.1) mit $p(x)$ anstelle von $q(x)$ und λ_n anstelle von μ .

Nach Satz 1.0.1 besitzt $u_n(x)$ im Intervall $(0, \pi]$ $(n+1)$ Nullstellen und es ist $u_n(\pi) = 0$. Wegen

$$q(x) - \lambda_n \geq p(x) - \lambda_n$$

liegt nach Satz 1.0.2 zwischen zwei Nullstellen von u_n (mindestens) eine Nullstelle von v . Es gilt also $N(\lambda_n; p) \geq n+1$, woraus nach Korollar 1.1.2b) mit $\bar{\lambda} = \lambda_n$ die Ungleichung $\lambda_n(p) \leq \lambda_n(q)$ folgt. \square

Korollar 1.1.5 Für $p(x), q(x) \in C^0[0, \pi]$ gilt

$$|\lambda_n(q) - \lambda_n(p)| \leq \|q - p\|_\infty \quad \text{für alle } n \in \mathbb{N}_0.$$

Beweis: Aus dem Monotoniesatz folgt wegen $p(x) \leq q(x) + \|q - p\|_\infty$ mit Hilfe von Lemma 1.1.3

$$\lambda_n(p) \leq \lambda_n(q + \|q - p\|_\infty) = \lambda_n(q) + \|q - p\|_\infty \quad \text{für alle } n \in \mathbb{N}_0. \quad (1.2)$$

Vertauschen von p und q in (1.2) liefert die Behauptung. \square

1.2 Asymptotische Entwicklungen

In diesem und im nächsten Abschnitt sei nun $q(x) \in L^2[0, \pi]$ vorausgesetzt. Eine stetig differenzierbare Funktion $u(x)$ heißt dann Lösung der Differentialgleichung

$$-u'' + q(x)u = \lambda u, \quad (1.3)$$

wenn sie eine absolutstetige Ableitung besitzt und die Differentialgleichung fast überall erfüllt (siehe Pöschel und Trubowitz [39], Seite 2). Lösungen von Anfangs- oder Randwertproblemen für (1.3) haben darüberhinaus den gestellten Anfangs- bzw. Randbedingungen zu genügen.

Betrachten wir das Dirichletproblem für den einfachsten Fall $q(x) \equiv 0$, so erhalten wir die Aufgabe

$$\begin{aligned} -u'' &= \lambda u \\ u(0) &= u(\pi) = 0. \end{aligned} \quad (1.4)$$

(1.4) besitzt die Eigenwerte $\lambda_{n-1} = n^2$, $n \in \mathbb{N}$, mit den zugehörigen normierten Eigenfunktionen $g_{n-1}(x) = \sqrt{\frac{2}{\pi}} \sin nx$ (wir normieren die Eigenfunktionen in $L^2[0, \pi]$ so, daß $g'_n(0)$ positiv ist, und verwenden für diese Eigenfunktionen im folgenden den Buchstaben g).

Die Eigenfunktionen bilden eine Orthonormalbasis in $L^2[0, \pi]$, d.h. es gilt

$$\int_0^\pi g_i(x) g_j(x) dx = \delta_{ij},$$

wobei δ_{ij} das Kroneckersymbol

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

bezeichnet, und jede Funktion $v(x) \in L^2[0, \pi]$ läßt sich in eine Fourierreihe

$$v(x) = \sum_{i=0}^{\infty} c_i g_i(x)$$

mit $c_i = \int_0^\pi v(t) g_i(t) dt$ entwickeln. Die Reihe konvergiert in $L^2[0, \pi]$, d.h.

$$\lim_{n \rightarrow \infty} \int_0^\pi (v(x) - \sum_{i=0}^n c_i g_i(x))^2 dx = 0.$$

Die gleiche Eigenschaft weisen auch die normierten Eigenfunktionen des Dirichletproblems für ein beliebiges quadratisch integrierbares Potential $q(x)$ auf. Wir formulieren diesen Sachverhalt als *Entwicklungssatz nach Eigenfunktionen*:

Satz 1.2.1 (Entwicklungssatz) Für $q(x) \in L^2[0, \pi]$ läßt sich jede Funktion $v(x) \in L^2[0, \pi]$ in eine Reihe nach den Eigenfunktionen $g_i(x)$ des Dirichletproblems entwickeln:

$$v(x) = \sum_{i=0}^{\infty} \left(\int_0^\pi v(t) g_i(t) dt \right) g_i(x).$$

Beweis: Walter [47], Seite 207. □

Für die Eigenwerte und Eigenfunktionen gelten die folgenden asymptotischen Entwicklungen:

Satz 1.2.2 Die Eigenwerte des Dirichletproblems genügen für $q(x) \in L^2[0, \pi]$ der asymptotischen Entwicklung

$$\lambda_{n-1} = n^2 + \frac{1}{\pi} \int_0^\pi q(t) dt + a_n \quad (1.5)$$

mit $\sum_{n=1}^{\infty} a_n^2 < \infty$.

Die normierten Eigenfunktionen $g_n(x)$ des Dirichletproblems erfüllen die Entwicklung

$$g_{n-1}(x) = \sqrt{\frac{2}{\pi}} \sin nx + O\left(\frac{1}{n}\right) \quad (1.6)$$

gleichmäßig auf $[0, \pi]$.

Beweis: Pöschel und Trubowitz [39], Seite 35ff. □

Ähnliche Entwicklungen finden sich bereits in der Arbeit von Borg [10] sowie bei Hochstadt [21]. Fix [13] gibt Rekursionsformeln an, mit denen sich für hinreichend oft differenzierbare Potentiale beliebig viele Koeffizienten der asymptotischen Entwicklung

$$\sqrt{\lambda_{n-1}} \sim n + \frac{c_1}{n} + \frac{c_2}{n^2} + \dots$$

berechnen lassen.

1.3 Das inverse Dirichletproblem

Wir betrachten ab sofort nur noch symmetrische Potentiale.

Definition 1.3.1 Die Funktion $q(x) \in L^2[0, \pi]$ heißt **symmetrisch**, wenn fast überall auf $[0, \pi]$

$$q(x) = q(\pi - x) \quad (1.7)$$

gilt.

Bereits Borg [10] beweist den folgenden Eindeutigkeitsatz für das inverse Dirichletproblem:

Satz 1.3.1 *Zu vorgegebenen Werten $\{\lambda_n\}_{n=0}^\infty$ gibt es höchstens eine symmetrische Funktion $q(x) \in L^2[0, \pi]$, so daß $\{\lambda_n\}_{n=0}^\infty$ das Spektrum des zugehörigen Dirichletproblems ist.*

Ist mindestens ein Eigenwert unbekannt, wird $q(x)$ durch die übrigen Eigenwerte nicht mehr eindeutig bestimmt.

Bemerkung: Der Eindeutigkeitsatz von Borg macht keine Aussage darüber, wann zu einem vermeintlichen Spektrum tatsächlich ein entsprechendes Potential $q(x)$ existiert. Der Satz besagt lediglich, daß, falls beim inversen Dirichletproblem überhaupt eine symmetrische Lösung existiert, diese durch ihr Spektrum eindeutig festgelegt wird.

Sind nicht alle Eigenwerte bekannt, gibt es zu dem verbleibenden Teilspektrum entweder keine Lösung oder mehrere Lösungen. \square

Pöschel und Trubowitz führen in [39] den Raum

$$E := \{q(x) \in L^2[0, \pi] : q(x) = q(\pi - x)\}$$

der auf dem Intervall $[0, \pi]$ symmetrischen L^2 -Funktionen sowie den Raum S aller reellen, streng monoton wachsenden Folgen $\{c_n\}_{n=0}^\infty$ der Gestalt

$$c_n = (n+1)^2 + s + a_n, \quad n = 0, 1, \dots$$

mit $s \in \mathbb{R}$ und einer quadratisch summierbaren Folge $\{a_n\}_{n=0}^\infty$ ein, und betrachten die Abbildung

$$\Lambda : E \rightarrow S, \quad q(x) \mapsto \Lambda(q) = \{\lambda_n(q)\}_{n=0}^\infty.$$

Sie zeigen ([39], Seite 115):

Satz 1.3.2 Λ bildet E bijektiv auf S ab. Λ und die Umkehrabbildung Λ^{-1} sind stetig.

Für den Fall symmetrischer Potentiale können wir die möglichen Spektren des Dirichletproblems also folgendermaßen beschreiben:

Korollar 1.3.3 Zu einer Folge $\Lambda = \{\lambda_n\}_{n=0}^{\infty}$ existiert genau dann eine (eindeutig bestimmte) symmetrische Funktion $q(x) \in L^2[0, \pi]$, so daß das zugehörige Dirichletproblem das Spektrum Λ besitzt, wenn die Folge streng monoton wächst und es eine Zahl $s \in \mathbb{R}$ und eine quadratisch summierbare Folge $\{a_n\}_{n=0}^{\infty}$ gibt, für die

$$\lambda_n = (n+1)^2 + s + a_n, \quad n = 0, 1, \dots$$

gilt.

Bemerkung: Wie die asymptotische Entwicklung (1.5) zeigt, ist

$$s = \frac{1}{\pi} \int_0^{\pi} q(t) dt.$$

□

Für unsere Zwecke besonders interessant ist die Darstellung der Differenz zweier Potentiale $q(x)$ und $\tilde{q}(x)$ (mit Eigenwerten λ_j bzw. $\tilde{\lambda}_j$) als unendliche Reihe, wie sie von Hochstadt [22] und Hald [18] formuliert wurde:

$$q(x) - \tilde{q}(x) = \sum_{\lambda_j(q) \neq \lambda_j(\tilde{q})} (\tilde{y}_j u_j)', \quad (1.8)$$

wobei \tilde{y}_j und u_j spezielle Lösungen der Differentialgleichungen

$$-\tilde{y}_j'' + \tilde{q}(x) \tilde{y}_j = \lambda_j \tilde{y}_j \quad (1.9)$$

$$-u_j'' + q(x) u_j = \lambda_j u_j \quad (1.10)$$

mit $\lambda_j = \lambda_j(q)$ sind. Die Reihe (1.8) konvergiert, falls $\sum_j |\lambda_j - \tilde{\lambda}_j| < \infty$ gilt.

Sind die ersten $n+1$ Eigenwerte von q und \tilde{q} identisch, so treten in der Reihe (1.8) nur solche Funktionen auf, die im Intervall $(0, \pi)$ mindestens $n+1$ Nullstellen besitzen, also hochfrequenten Schwingungen entsprechen. Wenn man nun das gesuchte Potential $q(x)$ in eine Fourierreihe oder in eine Reihe nach Eigenfunktionen eines Dirichletproblems entwickelt, dann kann man die Formel (1.8) qualitativ so interpretieren, daß aus den ersten $n+1$ Eigenwerten $\lambda_0, \dots, \lambda_n$ die Koeffizienten der niederfrequenten Schwingungsanteile des gesuchten Potentials bestimmt werden können, wohingegen diejenigen Anteile, die Schwingungen mit höheren Frequenzen entsprechen, unbestimmt bleiben.

Unterscheiden sich umgekehrt nur endlich viele Eigenwerte der Potentiale $q(x)$ und $\tilde{q}(x)$, wird die Reihe (1.8) zur endlichen Summe. Für $q(x) \in C^k[0, \pi]$ liegt u_j als Lösung der Differentialgleichung (1.10) in $C^{k+2}[0, \pi]$, ebenso ist \tilde{y}_j für ein l -mal stetig differenzierbares Potential $\tilde{q}(x)$ $l+2$ -mal stetig differenzierbar. Im Fall $k < l$ ist die durch (1.8) gegebene Differenz von $q(x)$ und $\tilde{q}(x)$ also mindestens $k+1$ -mal stetig differenzierbar, was dann auch für $q(x)$ gelten muß. Mit vollständiger Induktion folgt, daß $q(x)$ und $\tilde{q}(x)$ gleich oft differenzierbar sind, wenn sich nur endlich viele ihrer Eigenwerte unterscheiden.

Die soeben durchgeführten Überlegungen gelten auch, wenn man nur $q(x) \in L^2[0, \pi]$, $\tilde{q}(x) \in C^l[0, \pi]$ voraussetzt. In diesem Fall ist u_j einmal stetig differenzierbar, so daß die Differenz von $q(x)$ und $\tilde{q}(x)$ stetig ist. Also ist auch $q(x)$ selbst wieder stetig und in der Folge sogar l -mal stetig differenzierbar.

Aus diesem Sachverhalt läßt sich für das endliche inverse Dirichletproblem die folgende Existenzaussage herleiten:

Lemma 1.3.4 *Zu endlich vielen Eigenwerten existiert immer ein beliebig oft differenzierbares symmetrisches Potential als Lösung des inversen Dirichletproblems.*

Beweis: O.B.d.A. seien die Eigenwerte $\lambda_0 < \dots < \lambda_n$ vorgegeben, und für ein $c \in \mathbb{R}$ gelte $\lambda_n < (n+2)^2 + c$. Ergänzt man $\{\lambda_i\}_{i=0}^n$ durch die Eigenwerte $\{\tilde{\lambda}_i\}_{i=n+1}^\infty$ des konstanten Potentials $\tilde{q}(x) \equiv c$ zu einem vollen Spektrum, indem man für $i > n$

$$\lambda_i := (i+1)^2 + c$$

setzt, liegt die Folge $\{\lambda_i\}_{i=0}^\infty$ im Folgenraum S aus Satz (1.3.2). Also gibt es nach Satz 1.3.3 ein symmetrisches Potential $q(x) \in L^2[0, \pi]$, das diese Eigenwerte besitzt. Da sich nur endlich viele Eigenwerte von $q(x)$ und $\tilde{q}(x) \equiv c$ unterscheiden, folgt aus der obigen Betrachtung, daß $q(x)$ unendlich oft differenzierbar ist. \square

Kapitel 2

Ein Rekonstruktionsverfahren für das inverse Dirichletproblem

Wir wollen zum inversen Dirichletproblem einen Rekonstruktionsalgorithmus für ein stetiges symmetrisches Potential entwickeln. Dabei lassen wir uns von den folgenden Überlegungen leiten:

1. Es sind paarweise verschiedene Zahlen ν_i , $i = 0, 1, \dots, n$, vorgegeben, zu denen ein stetiges symmetrisches Potential $q(x)$ gesucht wird, so daß das zugehörige Dirichletproblem die Eigenwerte $\lambda_i(q) = \nu_i$, $i = 0, 1, \dots, n$, (und beliebige höhere Eigenwerte) besitzt.

Dies entspricht den Anforderungen der Praxis insofern, als nur endlich viele Eigenwerte durch Messungen bestimmt werden können, und von diesen die kleinsten (die den Grundschnwingungen entsprechen) am genauesten, während höhere Eigenwerte nicht oder nur sehr ungenau gemessen werden können.

2. Zur Konstruktion des gesuchten Potentials stehen stetige symmetrische Funktionen

$$\hat{q}(x), q_j(x), j = 0, 1, \dots, n$$

zur Verfügung, zu denen eine Lösung $q(x; a)$ des inversen Dirichletproblems in der Form

$$q(x; a) := \hat{q}(x) + \sum_{j=0}^n a_j q_j(x) \quad (2.1)$$

mit $a = (a_j) \in \mathbb{R}^{n+1}$ gesucht wird. Die Funktionen $\widehat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$, bezeichnen wir als *Basisfunktionen*.

Damit unterschiedliche Vektoren a^1 , a^2 auch unterschiedliche Potentiale festlegen, setzen wir voraus, daß die Basisfunktionen $q_j(x)$, $j = 0, 1, \dots, n$, linear unabhängig sind. Die Funktion $\widehat{q}(x)$ darf aber in der linearen Hülle der $q_j(x)$ enthalten sein, und auch der Fall $\widehat{q}(x) \equiv 0$ ist zugelassen.

3. Zur Einschließung einer Lösung berechnen wir stetige Funktionen $\underline{q}(x)$, $\bar{q}(x)$, so daß ein Vektor $a \in \mathbb{R}^{n+1}$ existiert, für den $q(x; a)$ Lösung des inversen Dirichletproblems ist und die Abschätzung

$$\underline{q}(x) \leq q(x; a) \leq \bar{q}(x) \quad \text{für alle } x \in [0, \pi]$$

erfüllt.

Durch (2.1) ist das Potential $q(x; a)$ und somit auch jeder Eigenwert λ_i des zugehörigen Dirichletproblems eine Funktion der Koeffizienten a_j . Zur Lösung des inversen Dirichletproblems ist ein Vektor $a = (a_j) \in \mathbb{R}^{n+1}$ gesucht, so daß

$$\lambda_i(a_0, \dots, a_n) = \lambda_i(q(x; a)) = \nu_i \quad \text{für } i = 0, 1, \dots, n \quad (2.2)$$

gilt.

Damit haben wir das endliche inverse Dirichletproblem auf die Lösung eines nichtlinearen Gleichungssystems im \mathbb{R}^{n+1} reduziert. Durch Subtraktion der vorgegebenen reellen Werte ν_i erhalten wir die Aufgabe, eine Nullstelle der Funktion

$$\begin{aligned} f &: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1} \\ f_i(a) &:= \lambda_i(q(x; a)) - \nu_i, \quad i = 0, 1, \dots, n \end{aligned} \quad (2.3)$$

zu bestimmen. Haben wir eine Nullstelle a von f gefunden, ist durch (2.1) eine Lösung des inversen Dirichletproblems gegeben.

Wir lösen das Nullstellenproblem (2.3) mit dem Newtonverfahren. Dazu weisen wir zunächst die Differenzierbarkeit von f im gesamten \mathbb{R}^{n+1} nach, um zu gewährleisten, daß das Newtonverfahren überhaupt anwendbar ist. Da f eine zusammengesetzte Funktion ist, benötigen wir Hilfsmittel aus der Differentialrechnung in Banachräumen. Diese stellen wir im nächsten Abschnitt bereit.

2.1 Differentialrechnung in Banachräumen

Wir geben hier nur die für diese Arbeit wichtigsten Eigenschaften der Differentialrechnung in Banachräumen an und verweisen z.B. auf Dieudonné [12] für eine ausführliche Behandlung.

Im folgenden seien E, F, G Banachräume und A eine offene Teilmenge von E . Weiter bezeichne $\mathcal{L}(E, F)$ den Raum der stetigen linearen Abbildungen von E nach F . Eine Abbildung $f : A \rightarrow F$ heißt (*Fréchet-*)*differenzierbar* im Punkt $x_0 \in A$, wenn es einen stetigen linearen Operator $f'(x_0) \in \mathcal{L}(E, F)$ gibt, so daß für $x \rightarrow x_0$

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + o(\|x - x_0\|)$$

gilt. $f'(x_0)$ ist dann eindeutig bestimmt und wird als (*Fréchet-*)*Ableitung von $f(x)$ an der Stelle x_0* bezeichnet. Statt $f'(x_0)$ benutzen wir auch die Schreibweise $d_x f(x_0)$ (Gradient von f) und schreiben $d_x f(x_0)v$ oder $f'(x_0)v$, wenn wir den linearen Operator $f'(x_0)$ auf $v \in E$ anwenden.

Ist f im Punkt x_0 differenzierbar, so gilt insbesondere für alle $v \in E$

$$\lim_{t \rightarrow 0} \frac{f(x_0 + tv) - f(x_0)}{t} = d_x f(x_0)v.$$

Einige grundlegende Eigenschaften der Ableitung fassen wir zusammen im

Satz 2.1.1 a) Für $f \in \mathcal{L}(E, F)$ ist $f'(x_0) = f$ für jedes $x_0 \in E$.

b) Ist A eine Umgebung von x_0 in E , sind $f, g : A \rightarrow F$ differenzierbar in x_0 und ist $c \in \mathbb{R}$, so sind $f + g$ und cf in x_0 differenzierbar und es gilt

$$\begin{aligned} (f + g)'(x_0) &= f'(x_0) + g'(x_0) \\ (cf)'(x_0) &= cf'(x_0). \end{aligned}$$

c) Ist A eine Umgebung von x_0 in E , $f : A \rightarrow F$ differenzierbar in x_0 , B eine Umgebung von $y_0 = f(x_0)$ in F , $g : B \rightarrow G$ differenzierbar in y_0 , so ist $h = g \circ f$ in x_0 differenzierbar und es gilt $h'(x_0) = g'(y_0) \circ f'(x_0)$, d.h.

$$d_x h(x_0)v = d_y g(y_0) d_x f(x_0)v.$$

Ist E ein Hilbertraum mit Skalarprodukt (\cdot, \cdot) , so ist die Ableitung einer differenzierbaren Funktion $f : E \rightarrow \mathbb{R}$ eine stetige Linearform auf E . Sie kann nach dem *Darstellungssatz von Fréchet-Riesz* eindeutig durch ein erzeugendes Element $z_{f'(x_0)} \in E$ dargestellt werden:

$$f'(x_0)v = (z_{f'(x_0)}, v) \quad \text{für alle } v \in E. \quad (2.4)$$

Wir werden $f'(x_0)$ mit dem durch (2.4) definierten erzeugenden Element $z_{f'(x_0)}$ identifizieren, d.h. $z_{f'(x_0)}$ als Ableitung von f an der Stelle $x_0 \in E$ bezeichnen.

Ist f für alle $x_0 \in A$ differenzierbar, so ist durch $f' : A \rightarrow \mathcal{L}(E, F)$ eine Funktion definiert. f heißt stetig differenzierbar in A , wenn f' in A stetig ist. Falls f' (in A) selbst wieder differenzierbar ist, so heißt f *zweimal differenzierbar*. Durch rekursive Fortsetzung dieses Prozesses gelangt man zu höheren Ableitungen von f .

Die zweite Ableitung liegt in $\mathcal{L}(E, \mathcal{L}(E, F))$. Dieser Raum ist isomorph zu $\mathcal{L}(E^2, F)$, dem Raum der stetigen bilinearen Abbildungen $B : E \times E \rightarrow F$. Analog kann man die p -te Ableitung mit einer stetigen p -Linearform identifizieren.

Falls f in A p -mal stetig differenzierbar ist und das Segment $x+tv, 0 \leq t \leq 1$, in A enthalten ist, lautet die Taylorsche Formel (Dieudonné [12], Seite 191)

$$\begin{aligned} f(x+v) &= f(x) + f'(x)v + \dots + \frac{1}{(p-1)!} f^{(p-1)}(x) v^{(p-1)} \\ &\quad + \int_0^1 \frac{(1-t)^{p-1}}{(p-1)!} f^{(p)}(x+tv) dt v^{(p)}, \end{aligned}$$

wobei $v^{(k)} = (v, \dots, v)$ (k -mal) gesetzt wurde. Das auftretende Integral kann unter den Voraussetzungen dieser Arbeit als Riemann-Integral (siehe Heuser [20], Seite 335) interpretiert werden.

Das Restglied ist beschränkt durch

$$\frac{\|v\|^p}{p!} \sup_{0 \leq t \leq 1} \|f^{(p)}(x+tv)\|.$$

Die Norm $\|f^{(p)}(x+tv)\|$ bezeichnet die Operatornorm von $f^{(p)}(x+tv)$, die für einen multilinearen Operator $L \in \mathcal{L}(E^p, F)$ durch

$$\|L\| = \sup_{\|v^k\|=1, 1 \leq k \leq p} \|L(v^1, \dots, v^p)\|$$

gegeben ist.

Als Spezialfall erhalten wir für $p = 1$ den *Mittelwertsatz* (Heuser [20], Seite 337):

Satz 2.1.2 *Es seien E, F zwei Banachräume, x und y zwei Punkte aus E und $S := \{x + t(y - x), t \in [0, 1]\}$ ihre Verbindungsstrecke. Weiter sei f eine stetige Funktion einer Umgebung von S nach F . Ist f in S stetig differenzierbar, so gilt*

$$f(y) - f(x) = \int_0^1 f'(x + t(y - x)) dt (y - x),$$

und somit die Abschätzung

$$\|f(y) - f(x)\| \leq \|y - x\| \sup_{0 \leq t \leq 1} \|f'((x + t(y - x)))\|.$$

Ist f unendlich oft differenzierbar und konvergiert das Restglied in einer Kugel $\|v\| < r$ für $p \rightarrow \infty$ gleichmäßig gegen 0, kann man f in dieser Kugel in eine Taylor-Reihe

$$f(x + v) = \sum_{k=0}^{\infty} f^{(k)}(x) v^{(k)}$$

entwickeln. Falls f in jedem Punkt $x \in A$ in eine Taylor-Reihe entwickelbar ist, heißt f *analytisch*.

Hängt f von mehreren Veränderlichen x^1, \dots, x^n ab, so definieren wir die *partielle Ableitung* $\partial_{x^i} f(x_0^1, \dots, x_0^n)$ als Ableitung der Funktion $f_i(x^i) := f(x_0^1, \dots, x_0^{i-1}, x^i, x_0^{i+1}, \dots, x_0^n)$.

Für partielle Ableitungen gilt

Satz 2.1.3 *Es seien E_1, \dots, E_n Banachräume und A eine offene Teilmenge von $E_1 \times \dots \times E_n$. In diesem Fall ist die Funktion $f : A \rightarrow F$ genau dann auf A stetig differenzierbar, wenn f in jedem Punkt von A bezüglich jeder Variablen x^i partiell differenzierbar ist und die partiellen Ableitungen auf A stetig sind. In jedem Punkt von A ist die Ableitung von f dann durch*

$$d_x f v = \partial_{x^1} f v^1 + \dots + \partial_{x^n} f v^n$$

gegeben.

2.2 Ableitungen der Eigenwerte

Pöschel und Trubowitz [39] haben in ihrem Buch die Eigenwerte des Dirichletproblems und die Lösungen von (nachstehend definierten) Anfangswertproblemen als von q abhängige Operatoren auf dem Hilbertraum $L^2[0, \pi]$ mit Skalarprodukt

$$(u, v) := \int_0^\pi u(x)v(x) dx$$

betrachtet. Wir werden die in [39] erhaltenen Ergebnisse zur Analyse unseres Rekonstruktionsverfahrens verwenden.

Neben dem Dirichletproblem betrachten wir für $\lambda \in \mathbb{R}$ die Anfangswertprobleme

$$\begin{aligned} -y_j'' + q(x)y_j &= \lambda y_j, \quad j = 1, 2 \\ y_1(0; q, \lambda) &= 1, \quad y_1'(0; q, \lambda) = 0 \\ y_2(0; q, \lambda) &= 0, \quad y_2'(0; q, \lambda) = 1, \end{aligned} \tag{2.5}$$

wobei die Randwerte $y_1(0; q, \lambda) = 1 = y_2'(0; q, \lambda)$ Normierungsbedingungen darstellen und durch beliebige nichttriviale Vorgaben ersetzt werden können.

Für $i \in \mathbb{N}_0$ bezeichne λ_i den i -ten Eigenwert des Dirichletproblems zum (fest gewählten) Potential $q(x)$. Für $\lambda = \lambda_i$ ist die Lösung $y_2(x; q, \lambda)$ von (2.5) also eine Eigenfunktion des Dirichletproblems.

Weiter setzen wir

$$v_i(x; q) := y_1(x; q, \lambda_i) y_2(x; q, \lambda_i) \tag{2.6}$$

und für die normierte i -te Eigenfunktion des Dirichletproblems

$$g_i(x; q) := \frac{y_2(x; q, \lambda_i)}{\|y_2\|_2},$$

wobei $\|\cdot\|_2$ die L^2 -Norm bezeichnet.

Wenn wir besonders auf die Abhängigkeit der Funktionen y_j , v_i , g_i , usw. von $q(x)$ bzw. von λ hinweisen wollen, nehmen wir diese mit als Funktionsargumente auf, ansonsten lassen wir die zusätzlichen Argumente weg und schreiben z.B. statt $y_1(x; q, \lambda)$ einfacher $y_1(x)$ oder nur y_1 .

Die eingeführten Funktionen erfüllen die folgende Orthogonalitätsrelation:

Satz 2.2.1 Für $i, j \geq 0$ gelten

$$(v_i, \frac{d}{dx} g_j^2) = \frac{1}{2} \delta_{ij}. \quad (2.7)$$

Beweis: Pöschel und Trubowitz [39], Seite 44. \square

Pöschel und Trubowitz zeigen in [39], daß durch die Eigenwerte differenzierbare Funktionale auf L^2 definiert sind. Ihre Ableitungen sind stetige Linearformen auf L^2 , die sich nach dem Rieszschen Darstellungssatz mit L^2 -Funktionen identifizieren lassen:

Satz 2.2.2 Jeder Eigenwert des Dirichletproblems definiert ein Funktional

$$\lambda_i : \begin{cases} L^2[0, \pi] & \rightarrow \mathbb{R} \\ q & \mapsto \lambda_i(q) \end{cases}.$$

λ_i ist auf $L^2[0, \pi]$ analytisch und besitzt den Gradienten

$$d_q \lambda_i = g_i^2(\cdot; q),$$

d.h. es gilt für alle Funktionen $v \in L^2[0, \pi]$

$$d_q \lambda_i v = \int_0^\pi g_i^2(x; q) v(x) dx.$$

Beweis: Pöschel und Trubowitz [39], Seite 31. \square

Nach diesen Vorbereitungen können wir die Differenzierbarkeit der Funktion f aus (2.3) zeigen. Im folgenden schreiben wir dabei für die normierte i -te Eigenfunktion zum Potential $q(x; a)$ anstelle von $g_i(x; q(x; a))$ einfacher $g_i(x; a)$.

Satz 2.2.3 Die durch (2.3) definierte Funktion f ist auf dem gesamten \mathbb{R}^{n+1} unendlich oft differenzierbar und es gilt

$$\frac{\partial f_i}{\partial a_j}(a) = d_q \lambda_i(q(x; a)) q_j = \int_0^\pi g_i^2(x; a) q_j(x) dx.$$

Beweis: Wir betrachten für fest gewählte Basisfunktionen $\widehat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$, die Abbildungen

$$l : \mathbb{R}^{n+1} \rightarrow L^2[0, \pi], \quad a \mapsto \widehat{q}(x) + \sum_{j=0}^n a_j q_j(x)$$

und

$$h : L^2[0, \pi] \rightarrow \mathbb{R}^{n+1}, \quad q(x) \mapsto (\lambda_0(q) - \nu_0, \dots, \lambda_n(q) - \nu_n)^T.$$

l ist eine lineare Abbildung und somit beliebig oft differenzierbar. Als Funktion in den \mathbb{R}^{n+1} ist h genau dann differenzierbar, wenn jede Komponente von h differenzierbar ist. Die i -te Komponente von h entspricht aber gerade dem i -ten Eigenwert λ_i des Dirichletproblems, vermindert um die reelle Konstante ν_i . Nach Satz 2.2.2 ist λ_i eine unendlich oft differenzierbare Funktion, und aus der Kettenregel folgt unmittelbar, daß $f(a) = h(l(a))$ ebenfalls unendlich oft differenzierbar ist.

Die Formel für die partiellen Ableitungen ergibt sich aus der Kettenregel $f'(a) = h'(l(a)) \circ l'(a)$. \square

2.3 Das Rekonstruktionsverfahren

Das Newtonverfahren zur Bestimmung einer Nullstelle einer differenzierbaren Funktion $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ ist gegeben durch die Iterationsvorschrift

$$\begin{aligned} x^{(0)} &\in D \\ x^{(k+1)} &:= x^{(k)} - f'(x^{(k)})^{-1} f(x^{(k)}), \quad k = 0, 1, \dots \end{aligned}$$

Nachdem die Differenzierbarkeit der Funktion f aus (2.3) bewiesen ist, können wir das Newtonverfahren auf f anwenden und eine Folge von Potentialen $q^{(k)}(x)$, $k = 0, 1, \dots$, der Gestalt (2.1) berechnen. Wir formulieren dieses Verfahren als Algorithmus 2.1. Die einzigen Eingangsdaten sind die vorgegebenen, paarweise verschiedenen Eigenwerte ν_i , $i = 0, 1, \dots, n$.

Wir untersuchen zunächst einige Eigenschaften von Algorithmus 2.1 und zeigen dann, wie sich Lösungen des inversen Dirichletproblems einschließen lassen.

Algorithmus 2.1: Rekonstruktion für das inverse Dirichletproblem
Newtonverfahren im \mathbb{R}^{n+1}

1. Wähle symmetrische Basisfunktionen $\widehat{q}(x), q_j(x), j = 0, 1, \dots, n$.
2. Wähle $a^{(0)} \in \mathbb{R}^{n+1}$.
3. Für $k = 0, 1, \dots$:

(i) Für $i = 0, 1, \dots, n$:

(a) Berechne $f_i(a^{(k)}) = \lambda_i(q^{(k)}) - \nu_i$,

(b) Berechne $g_i(x; a^{(k)}) = g_i(x; q^{(k)}(x))$,

wobei

$$q^{(k)}(x) := q(x; a^{(k)}) := \widehat{q}(x) + \sum_{j=0}^n a_j^{(k)} q_j(x).$$

(ii) Berechne die Funktionalmatrix

$$\left(\frac{\partial f_i}{\partial a_j}(a^{(k)}) \right) = \left(\int_0^\pi g_i^2(x; a^{(k)}) q_j(x) dx \right), \quad i, j = 0, 1, \dots, n.$$

(iii) Führe den Newtonschritt

$$a^{(k+1)} := a^{(k)} - \left(\frac{\partial f_i}{\partial a_j}(a^{(k)}) \right)^{-1} f(a^{(k)})$$

durch.

2.4 Addition von Konstanten

Wie im Lemma 1.1.3 gezeigt wurde, erhöhen sich alle Eigenwerte des Dirichletproblems bei der Addition der konstanten Funktion c zum vorliegenden

Potential $q(x)$ ebenfalls um c , d.h. es gilt für alle $i \in \mathbb{N}_0$

$$\lambda_i(q(x) + c) = \lambda_i(q(x)) + c$$

Daher sollte beim inversen Dirichletproblem eine Addition von c zu allen vorgegebenen Eigenwerten ν_i zu einer ebenso großen Änderung der berechneten Potentiale $q^{(k)}(x)$ führen.

In der Tat besitzt unser Algorithmus bei geeigneter Wahl der Basisfunktionen diese Eigenschaft:

Satz 2.4.1 Sind die Eigenwerte $\{\nu_i\}_{i=0}^n$ und $\{\mu_i\}_{i=0}^n$ mit

$$\mu_i = \nu_i + c \quad \text{für } i = 0, 1, \dots, n$$

vorgegeben, zu denen Potentiale $q(x)$ und $p(x)$ gesucht sind, und werden in beiden Fällen im Rekonstruktionsverfahren 2.1 die gleichen Basisfunktionen $\hat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$, und identische Startnäherungen $q^{(0)}(x) = p^{(0)}(x)$ verwendet, so gilt

$$p^{(k)}(x) = q^{(k)}(x) + c \quad \text{für } k = 1, 2, \dots, \quad (2.8)$$

falls es einen Vektor $b = (b_j) \in \mathbb{R}^{n+1}$ gibt, so daß

$$\sum_{j=0}^n b_j q_j(x) \equiv 1. \quad (2.9)$$

Die Eigenschaft (2.8) ist unabhängig von der Wahl der Basisfunktion $\hat{q}(x)$.

Beweis: Die i -te Zeile der Funktionalmatrix von f aus (2.3) an der Stelle a besteht aus dem Zeilenvektor $z^i = (z_{ij})_{j=0}^n$ mit

$$z_{ij} = \frac{\partial f_i}{\partial a_j}(a) = \int_0^\pi g_i^2(x; a) q_j(x) dx.$$

Falls (2.9) erfüllt ist, ergibt die Multiplikation mit dem Spaltenvektor b

$$\begin{aligned} z^i \cdot b &= \sum_{j=0}^n b_j \int_0^\pi g_i^2(x; a) q_j(x) dx \\ &= \int_0^\pi g_i^2(x; a) \sum_{j=0}^n b_j q_j(x) dx = \int_0^\pi g_i^2(x; a) dx = 1. \end{aligned}$$

Also gilt

$$f'(a) \cdot b = (1, 1, \dots, 1)^T$$

für alle $a \in \mathbb{R}^{n+1}$, und falls die Funktionalmatrix nichtsingulär ist, folgt

$$f'(a)^{-1} \cdot (c, c, \dots, c)^T = cf'(a)^{-1} \cdot (1, 1, \dots, 1)^T = c \cdot b,$$

woraus sich

$$p^{(1)}(x) = q^{(1)}(x) + c \sum_{j=0}^n b_j q_j(x) = q^{(1)}(x) + c$$

ergibt.

Da die Eigenfunktionen g_i (und somit auch die Funktionalmatrizen beim Rekonstruktionsverfahren) von $q(x)$ und $q(x) + c$ identisch sind, folgt die Behauptung des Satzes. \square

Bemerkung: Eine Erhöhung aller vorgegebenen Eigenwerte ν_i um c ändert die Funktionswerte im Algorithmus 2.1 in genau der gleichen Weise wie eine Reduktion von $q^{(0)}(x)$ um c . Der Satz 2.4.1 besagt also, daß sich die Iterierten $q^{(k)}(x)$, $k \geq 1$, im Algorithmus 2.1 nicht ändern, wenn man die Startnäherung $q^{(0)}(x)$ durch $q^{(0)}(x) - c$ ersetzt und die Funktion $h(x) \equiv 1$ in der linearen Hülle der Basisfunktionen q_j enthalten ist. \square

2.5 Abbruch des Rekonstruktionsverfahrens

Das Rekonstruktionsverfahren bricht im k -ten Iterationsschritt ab, falls die Funktionalmatrix

$$\left(\frac{\partial f_i}{\partial a_j}(a^{(k)}) \right) = \left(\int_0^\pi g_i^2(x; a^{(k)}) q_j(x) dx \right)$$

singulär wird, was genau dann geschieht, wenn es in der linearen Hülle der Funktionen $q_j(x)$, $j = 0, 1, \dots, n$, eine Funktion $\bar{q}(x) \not\equiv 0$ gibt, so daß

$$\int_0^\pi \bar{q}(x) g_i^2(x; a^{(k)}) dx = 0 \quad \text{für } i = 0, 1, \dots, n.$$

Dieser Fall kann in der Tat auftreten. Man braucht z.B. bei der Wahl $q^{(0)}(x) \equiv 0$ nur Basisfunktionen $q_j(x)$, $j = 0, 1, \dots, n$, zu wählen, die alle auf den Quadraten der ersten $n + 1$ Eigenfunktionen des Nullpotentials,

d.h. auf den Funktionen $\sin^2(ix)$, $i = 1, 2, \dots, n+1$, senkrecht stehen. Bei dieser Wahl ist die Funktionalmatrix des Newtonverfahrens bereits im ersten Schritt die Nullmatrix.

Will man einen derartigen Abbruch provozieren, muß man aber Basisfunktionen benutzen, die hochfrequenten Schwingungen entsprechen. Nach den Überlegungen in Kapitel 1 ist eine derartige Wahl nicht sinnvoll, da die hochfrequenten Schwingungsanteile des Potentials aus den ersten $n+1$ Eigenwerten i.a. nicht bestimmt werden können.

Es ist außerdem möglich, für eine beliebige stetige Funktion $\hat{q}(x)$ Basisfunktionen $q_j(x)$, $j = 0, 1, \dots, n$, so festzulegen, daß die Funktionalmatrix von Algorithmus 2.1 im Nullpunkt des \mathbb{R}^{n+1} , d.h. für die Funktion $\hat{q}(x)$, nicht-singulär ist. Also ist beim Newtonverfahren O.B.d.A. $a^{(0)} = 0$ wählbar und mindestens ein Iterationsschritt durchführbar.

Als geeignete Basisfunktionen stehen nach Satz 2.2.1 Vielfache der Ableitungen der in (2.6) definierten Funktionen $v_j(x; \hat{q})$ zur Verfügung. Setzt man nämlich

$$q_j(x) := -2 \frac{d}{dx} v_j(x; \hat{q}(x)), \quad (2.10)$$

so ergibt sich mit (2.7) im Nullpunkt des \mathbb{R}^{n+1}

$$\begin{aligned} \frac{\partial f_i}{\partial a_j}(0) &= d_q \lambda_i(\hat{q}) q_j = -2 \int_0^\pi g_i^2(x; \hat{q}) \frac{d}{dx} v_j(x; \hat{q}) dx \\ &= 2 \int_0^\pi v_j(x; \hat{q}) \frac{d}{dx} g_i^2(x; \hat{q}) dx = \delta_{ij}, \end{aligned}$$

und somit als Funktionalmatrix an der Stelle $\hat{q}(x)$ die Einheitsmatrix. Daher nennen wir die Funktionen $q_j(x)$ aus (2.10) *optimale Basisergänzung von $\hat{q}(x)$* .

Die Funktionen v_j können durch Lösen des direkten Eigenwertproblems für $\hat{q}(x)$ berechnet werden. Beispielsweise für $\hat{q}(x) \equiv 0$ erhält man

$$v_j(x) = \frac{1}{j+1} \cos(j+1)x \sin(j+1)x = \frac{1}{2j+2} \sin(2j+2)x,$$

also

$$q_j(x) = -2 \frac{d}{dx} v_j(x) = -2 \cos(2j+2)x.$$

Bemerkung: Die optimale Basisergänzung ist besonders dann interessant, wenn man eine Näherungslösung $\tilde{q}(x)$ des inversen Dirichletproblems kennt. In diesem Fall kann man

$$\hat{q}(x) := \tilde{q}(x)$$

und

$$q_j(x) := -2 \frac{d}{dx} v_j(x; \tilde{q}(x))$$

setzen.

Startet man nun das Newtonverfahren mit $q^{(0)}(x) = \tilde{q}(x)$, so ist die Funktionalmatrix im ersten Iterationsschritt die Einheitsmatrix. Also liegt die erste Iterierte nahe bei $\tilde{q}(x)$, wenn die Näherungslösung „hinreichend gut“ ist, und es ist zu erwarten, daß es in dem so gewählten Funktionenraum eine Lösung in der Nähe von $\tilde{q}(x)$ gibt, die durch das Newtonverfahren berechnet wird. \square

2.6 Der Satz von Kantorowitsch

Das Newtonverfahren ist unter bestimmten Voraussetzungen an f lokal quadratisch konvergent (Ortega [37], Seite 148):

Satz 2.6.1 $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ sei differenzierbar in einer Umgebung U einer Lösung x^* der Gleichung $f(x) = 0$, f' sei stetig in x^* und $f'(x^*)$ nichtsingulär. Dann gibt es eine Umgebung $V \subseteq U$ von x^* , so daß das Newtonverfahren für jeden Startwert $x^{(0)} \in V$ gegen x^* konvergiert.

Ist f' darüberhinaus Lipschitz-stetig in U , so gibt es Konstanten $k_0 = k_0(x^{(0)})$ und $c \in \mathbb{R}$, so daß

$$\|x^{(k+1)} - x^*\| \leq c \|x^{(k)} - x^*\|^2$$

für alle $k \geq k_0$.

Falls man beim Newtonverfahren nicht mit den Iterierten auf eine Nullstelle von f trifft, hat man beim Abbruch der Iteration nach endlich vielen Schritten keine Information darüber, ob man sich wirklich in der Nähe einer Nullstelle von f befindet.

Unter zusätzlichen Anforderungen an f kann man diesen Mangel beheben, und man erhält beim Newtonverfahren Konvergenzaussagen, die es ermöglichen, die Existenz von Lösungen nachzuweisen und Fehlerabschätzungen zu berechnen. Es gilt der *Satz von Kantorowitsch* (Ortega [37], Seite 155):

Satz 2.6.2 $D \subseteq \mathbb{R}^n$ sei konvex, $f : D \rightarrow \mathbb{R}^n$ differenzierbar in D und es gelte

$$\|f'(x) - f'(y)\| \leq \gamma \|x - y\| \quad \text{für alle } x, y \in D.$$

Außerdem gebe es ein $x^{(0)} \in D$, so daß mit

$$\beta := \|f'(x^{(0)})^{-1}\|, \quad \eta := \|f'(x^{(0)})^{-1}f(x^{(0)})\| \quad (2.11)$$

gilt:

$$\alpha := \beta\gamma\eta < \frac{1}{2}. \quad (2.12)$$

Falls weiter für

$$t := \frac{1 - (1 - 2\alpha)^{\frac{1}{2}}}{\beta\gamma}$$

die Inklusion

$$S := \{x : \|x - x^{(0)}\| \leq t\} \subseteq D$$

erfüllt ist, dann liegen die Iterierten $x^{(k)}$ des Newtonverfahrens

$$x^{(k+1)} = x^{(k)} - f'(x^{(k)})^{-1}f(x^{(k)}), \quad k = 0, 1, \dots$$

alle in S und konvergieren gegen eine Lösung x^* von $f(x) = 0$. Außerdem gilt die Fehlerabschätzung

$$\|x^* - x^{(1)}\| \leq 2\beta\gamma \|x^{(1)} - x^{(0)}\|^2. \quad (2.13)$$

x^* ist eindeutig bestimmt in $D \cap \widehat{S}$, wobei

$$\widehat{S} := \left\{x : \|x - x^{(0)}\| \leq \frac{1 + (1 - 2\alpha)^{\frac{1}{2}}}{\beta\gamma}\right\}.$$

Wir werden den Satz von Kantorowitsch auf das Rekonstruktionsverfahren 2.1 anwenden, und ausgehend von einer Näherungslösung $\tilde{q}(x)$ eine Einschließung einer exakten Lösung des inversen Dirichletproblems berechnen.

2.7 Abschätzung der zweiten Ableitung

Um die beim Satz von Kantorowitsch auftretenden Konstanten zu bestimmen, benötigen wir u.a. Berechnungsformeln für die zweite Ableitung der Eigenwerte nach dem Potential q . Die Herleitung dieser Formeln ist zwar elementar, durch die Kettenregel aber langwierig. Als Vorbereitung dient

Satz 2.7.1 *Die durch (2.5) definierten Funktionen $y_j(x; q, \lambda)$, $j = 1, 2$, sind partiell nach q und λ differenzierbar. Ihre partiellen Ableitungen nach q sind*

$$\partial_q y_j(x) = y_j(t) (y_1(t)y_2(x) - y_1(x)y_2(t)) 1_{[0,x]}(t),$$

wobei

$$1_{[0,x]}(t) = \begin{cases} 1, & 0 \leq t \leq x \\ 0, & \text{sonst} \end{cases}$$

die Indikatorfunktion des Intervalls $[0, x]$ bezeichnet.

Beweis: Pöschel und Trubowitz [39], Seite 21. □

Wir haben wieder von der Identifizierung der durch die Ableitung gegebenen linearen Abbildung mit dem erzeugenden Element im Hilbertraum L^2 Gebrauch gemacht. Es gilt also

$$\partial_q y_j(x) v = \int_0^x y_j(t) (y_1(t)y_2(x) - y_1(x)y_2(t)) v(t) dt.$$

Aus Satz 2.7.1 folgt unmittelbar eine Formel zur Berechnung der partiellen Ableitungen der Funktionen y_j nach λ :

Korollar 2.7.2 y_j , $j = 1, 2$ seien die durch (2.5) definierten Funktionen. Dann gilt für die partiellen Ableitungen von $y_j(x; q, \lambda)$ nach λ

$$\partial_\lambda y_j(x) = \partial_q y_j(x) (-1) = - \int_0^x y_j(t) (y_1(t)y_2(x) - y_1(x)y_2(t)) dt.$$

Beweis: Die partiellen Ableitungen nach λ folgen wegen

$$y_j(x; q(t), \lambda + \mu) = y_j(x; q(t) - \mu, \lambda)$$

aus

$$\frac{y_j(x, q, \lambda + \mu) - y_j(x, q, \lambda)}{\mu} = \frac{y_j(x, q - \mu, \lambda) - y_j(x, q, \lambda)}{-(-\mu)}.$$

□

Nach diesen Vorbereitungen können wir die zweite Ableitung des Eigenwerts λ_i nach dem Potential berechnen. Im folgenden bezeichnen $y_j = y_j(x; q, \lambda)$, $j = 1, 2$ die Lösungen der Anfangswertprobleme (2.5) und $g_i = g_i(x; q)$ die normierte i -te Eigenfunktion des Dirichletproblems mit Potential $q(x)$. Außerdem verwenden wir die Abkürzungen

$$V := \int_0^\pi g_i^2(s) v(s) ds, \quad W := \int_0^\pi g_i^2(s) w(s) ds.$$

Erster Schritt: Bei der Berechnung des totalen Differentials $d_q y_2(x; q, \lambda_i(q))$ muß die Abhängigkeit des i -ten Eigenwerts von q berücksichtigt werden. Mit Hilfe der Kettenregel ergibt sich

$$\begin{aligned} & d_q y_2(x; q, \lambda_i(q)) v \\ &= \partial_q y_2(x; q, \lambda_i) v + \partial_\lambda y_2(x; q, \lambda_i) d_q \lambda_i v \\ &= \int_0^x y_2(t) (y_1(t) y_2(x) - y_1(x) y_2(t)) v(t) dt \\ &\quad - \int_0^x y_2(t) (y_1(t) y_2(x) - y_1(x) y_2(t)) \int_0^\pi g_i^2(s) v(s) ds dt \\ &= \int_0^x y_2(t) (y_1(t) y_2(x) - y_1(x) y_2(t)) (v(t) - V) dt. \end{aligned}$$

Zweiter Schritt: Da das Integral als linearer Operator beim Differenzieren erhalten bleibt, erhält man mit der Kettenregel

$$\begin{aligned} & d_q \left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) v(x) dx \right) w \\ &= \int_0^\pi 2 v(x) y_2(x) d_q y_2(x; q, \lambda_i(q)) w dx \\ &= \int_0^\pi 2 v(x) y_2(x) \times \\ &\quad \times \left(\int_0^x y_2(t) (y_1(t) y_2(x) - y_1(x) y_2(t)) (w(t) - W) dt \right) dx. \end{aligned}$$

Dritter Schritt: Wegen

$$g_i(x; q) = \frac{y_2(x; q, \lambda_i(q))}{\|y_2\|_2}$$

berechnet sich die zweite Ableitung von λ_i folgendermaßen:

$$\begin{aligned}
d_q^2 \lambda_i (v, w) &= d_q(d_q \lambda_i v) w = d_q\left(\int_0^\pi g_i^2(x) v(x) dx\right) w \\
&= d_q\left(\frac{\int_0^\pi y_2^2(x; q, \lambda_i(q)) v(x) dx}{\int_0^\pi y_2^2(x; q, \lambda_i(q)) dx}\right) w \\
&= \frac{\int_0^\pi y_2^2(x; q, \lambda_i(q)) dx \, d_q\left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) v(x) dx\right) w}{\left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) dx\right)^2} \\
&\quad - \frac{\int_0^\pi y_2^2(x; q, \lambda_i(q)) v(x) dx \, d_q\left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) dx\right) w}{\left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) dx\right)^2} \\
&= \frac{1}{\|y_2\|_2^2} \left(d_q\left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) v(x) dx\right) w \right. \\
&\quad \left. - \int_0^\pi g_i^2(x) v(x) dx \, d_q\left(\int_0^\pi y_2^2(x; q, \lambda_i(q)) dx\right) w \right) \\
&= \frac{1}{\|y_2\|_2^2} \left(\int_0^\pi 2v(x)y_2(x) \times \right. \\
&\quad \times \left(\int_0^x y_2(t) (y_1(t)y_2(x) - y_1(x)y_2(t)) (w(t) - W) dt \right) dx \\
&\quad \left. - V \int_0^\pi 2y_2(x) \times \right. \\
&\quad \left. \times \left(\int_0^x y_2(t) (y_1(t)y_2(x) - y_1(x)y_2(t)) (w(t) - W) dt \right) dx \right) \\
&= \frac{2}{\|y_2\|_2^2} \int_0^\pi (v(x) - V) \times \\
&\quad \times \left(\int_0^x (y_1(t)y_2(t)y_2^2(x) - y_1(x)y_2(x)y_2^2(t)) (w(t) - W) dt \right) dx \\
&= 2 \int_0^\pi (v(x) - V) \times \\
&\quad \times \left(\int_0^x (y_1(t)y_2(t)g_i^2(x) - y_1(x)y_2(x)g_i^2(t)) (w(t) - W) dt \right) dx
\end{aligned}$$

$$\begin{aligned}
&= 2 \int_0^\pi (v(x) - \int_0^\pi g_i^2(s) v(s) ds) \times \\
&\quad \times \left(\int_0^x (y_1(t)y_2(t)g_i^2(x) - y_1(x)y_2(x)g_i^2(t)) \times \right. \\
&\quad \left. \times (w(t) - \int_0^\pi g_i^2(s) w(s) ds) dt \right) dx \\
&=: B_i(q)(v, w).
\end{aligned}$$

Mit partieller Integration läßt sich die Symmetrie des obigen Ausdrucks zeigen. Wir verwenden dabei, daß wegen $\int_0^\pi g_i^2(s) ds = 1$

$$\int_0^\pi g_i^2(t) (v(t) - V) dt = V - V \int_0^\pi g_i^2(t) dt = 0$$

und ebenso

$$\int_0^\pi g_i^2(t) (w(t) - W) dt = 0.$$

Es gilt:

$$\begin{aligned}
\frac{1}{2} B_i(q)(v, w) &= \int_0^\pi (v(x) - V) \times \\
&\quad \times \left(\int_0^x (y_1(t)y_2(t)g_i^2(x) - y_1(x)y_2(x)g_i^2(t)) (w(t) - W) dt \right) dx \\
&= \int_0^\pi g_i^2(x) (v(x) - V) \left(\int_0^x y_1(t)y_2(t) (w(t) - W) dt \right) dx \\
&\quad - \int_0^\pi y_1(x)y_2(x) (v(x) - V) \left(\int_0^x g_i^2(t) (w(t) - W) dt \right) dx \\
&\stackrel{\text{p.I.}}{=} \int_0^\pi g_i^2(t) (v(t) - V) dt \int_0^\pi y_1(t)y_2(t) (w(t) - W) dt \\
&\quad - \int_0^\pi y_1(x)y_2(x) (w(x) - W) \left(\int_0^x g_i^2(t) (v(t) - V) dt \right) dx \\
&\quad - \int_0^\pi y_1(t)y_2(t) (v(t) - V) dt \int_0^\pi g_i^2(t) (w(t) - W) dt
\end{aligned}$$

$$\begin{aligned}
& + \int_0^\pi g_i^2(x) (w(x) - W) \left(\int_0^x y_1(t) y_2(t) (v(t) - V) dt \right) dx \\
& = \int_0^\pi (w(x) - W) \times \\
& \quad \times \left(\int_0^x (y_1(t) y_2(t) g_i^2(x) - y_1(x) y_2(x) g_i^2(t)) (v(t) - V) dt \right) dx \\
& = \frac{1}{2} B_i(q) (w, v).
\end{aligned}$$

Durch die zweite Ableitung von λ_i ist eine symmetrische Bilinearform

$$B_i(q) = d_q^2 \lambda_i : \begin{cases} L^2 \times L^2 \rightarrow \mathbb{R} \\ (v, w) \mapsto d_q^2 \lambda_i (v, w) \end{cases} \quad (2.14)$$

definiert. Normabschätzungen der zweiten Ableitung sind an die zugrundegelegte L^2 -Norm gebunden. Wir werden aber in den Anwendungen mit stetigen Potentialen arbeiten und benötigen daher für $B_i(q)$ Abschätzungen in der Maximumsnorm. Das Normsymbol steht deshalb bei Funktionen im folgenden für die Maximumsnorm.

Wir schränken $B_i(q)$ auf $C^0[0, \pi] \times C^0[0, \pi]$ ein. Dann ist die Norm von $B_i(q)$ durch

$$\|B_i(q)\| := \sup_{\|v\|, \|w\|=1} |B_i(q)(v, w)| \quad (2.15)$$

definiert und es gilt:

Satz 2.7.3 Die durch (2.15) gegebene Norm von $B_i(q)$ ist beschränkt durch

$$\|B_i(q)\| \leq 16 \pi \|y_1(x; q, \lambda_i)\| \|y_2(x; q, \lambda_i)\|.$$

Beweis: Wir setzen wieder

$$V := \int_0^\pi g_i^2(s) v(s) ds, \quad W := \int_0^\pi g_i^2(s) w(s) ds.$$

Wegen $\int_0^\pi g_i^2(s) ds = 1$ folgen $|V| \leq \|v\|$ und $|W| \leq \|w\|$. Damit erhalten wir:

$$\begin{aligned}
|B_i(q)(v, w)| &= \left| 2 \int_0^\pi (v(x) - V) \times \right. \\
&\quad \times \left. \left(\int_0^x (y_1(t)y_2(t)g_i^2(x) - y_1(x)y_2(x)g_i^2(t)) (w(t) - W) dt \right) dx \right| \\
&\leq 2 \int_0^\pi |v(x) - V| \times \\
&\quad \times \left(\int_0^x |y_1(t)y_2(t)g_i^2(x) - y_1(x)y_2(x)g_i^2(t)| |w(t) - W| dt \right) dx \\
&\leq 2 \cdot 2 \|v\| \cdot 2 \|w\| \int_0^\pi \int_0^x (|y_1(t)y_2(t)| g_i^2(x) + |y_1(x)y_2(x)| g_i^2(t)) dt dx \\
&\leq 8 \|v\| \|w\| \|y_1\| \|y_2\| \int_0^\pi \int_0^\pi (g_i^2(x) + g_i^2(t)) dt dx \\
&= 8 \|v\| \|w\| \|y_1\| \|y_2\| \int_0^\pi (\pi g_i^2(x) + 1) dx \\
&= 16 \pi \|v\| \|w\| \|y_1\| \|y_2\|.
\end{aligned}$$

□

2.8 Explizite Schranken der Eigenfunktionen

Die asymptotische Entwicklung (1.6) der Eigenfunktionen erlaubt für konkrete Werte von i keine Aussagen über die genaue Gestalt von g_i . Zur Berechnung der Konstanten beim Satz von Kantorowitsch brauchen wir aber explizite Schranken der Differenz zweier i -ter Eigenfunktionen zu verschiedenen Potentialen p und q .

In der uns bekannten Literatur haben wir solche Schranken nur für den Spezialfall $p \equiv 0$ gefunden. Da die Herleitung dieser Schranken nicht ganz einfach ist, stellen wir im folgenden zuerst den bei Pöschel und Trubowitz [39] benutzten Zugang für $p \equiv 0$ vor. Durch geringfügige Modifikationen werden wir daraus Schranken bei beliebigem p bestimmen.

Zu diesem Zweck müssen wir Differentialgleichungen betrachten, in denen komplexwertige Funktionen auftreten. Dabei ist die unabhängige Variable x jedoch nach wie vor reell. Eine komplexwertige Funktion $y(x)$ kann als Paar reeller Funktionen

$$y(x) = (\operatorname{Re} y(x), \operatorname{Im} y(x)) = \operatorname{Re} y(x) + i \operatorname{Im} y(x)$$

(mit der imaginären Einheit i), eine komplexwertige Differentialgleichung als System zweier reeller Differentialgleichungen geschrieben werden (siehe Walter [47], Seite 90). Stetigkeit und Differenzierbarkeit komplexwertiger Funktionen sind dann komponentenweise zu verstehen.

Unter einer Lösung der Differentialgleichung

$$-y'' + q(x)y = \lambda y \quad (2.16)$$

mit $\lambda \in \mathbb{C}$ und $q \in L^2_{\mathbb{C}}[0, \pi]$, dem Raum der komplexwertigen, auf $[0, \pi]$ quadratisch integrierbaren Funktionen, verstehen wir eine komplexwertige, nach der reellen Veränderlichen x stetig differenzierbare Funktion $y(x)$ mit absolutstetiger Ableitung $y'(x)$, die die Differentialgleichung fast überall erfüllt.

Satz 2.8.1 *Seien $f \in L^2_{\mathbb{C}}$ und $a, b \in \mathbb{C}$. Die eindeutige Lösung des Anfangswertproblems*

$$\begin{aligned} -y'' &= \lambda y - f(x) \\ y(0) &= a, \quad y'(0) = b \end{aligned}$$

ist

$$y(x) = a \cos(\sqrt{\lambda}x) + b \frac{\sin(\sqrt{\lambda}x)}{\sqrt{\lambda}} + \int_0^x \frac{\sin(\sqrt{\lambda}(x-t))}{\sqrt{\lambda}} f(t) dt.$$

Beweis: Pöschel und Trubowitz [39], Seite 4. □

Um die Notation abzukürzen, schreiben wir im folgenden

$$c_{\lambda}(x) := \cos(\sqrt{\lambda}x), \quad s_{\lambda}(x) := \frac{\sin(\sqrt{\lambda}x)}{\sqrt{\lambda}},$$

womit wir die Lösung in der Form

$$y(x) = a c_{\lambda}(x) + b s_{\lambda}(x) + \int_0^x s_{\lambda}(x-t) f(t) dt \quad (2.17)$$

darstellen können.

Mit (2.17) lösen wir nun auf dem Intervall $[0, M]$, $M > 0$, die Anfangswertprobleme (2.5), d.h.

$$\begin{aligned} -y_j'' + q(x)y_j &= \lambda y_j, \quad j = 1, 2 \\ y_1(0; q, \lambda) &= 1, \quad y_1'(0; q, \lambda) = 0 \\ y_2(0; q, \lambda) &= 0, \quad y_2'(0; q, \lambda) = 1 \end{aligned} \quad (2.18)$$

durch einen formalen Potenzreihenansatz, wobei wir nur das Konstruktionsverfahren für y_1 beschreiben, da sich y_2 analog berechnen läßt.

$y_1(x; q, \lambda)$ wird in eine formale Potenzreihe nach q entwickelt:

$$y_1(x; q, \lambda) = C_0(x; \lambda) + \sum_{k=1}^{\infty} C_k(x; q, \lambda)$$

mit

$$C_k(x; q, \lambda) = C_k(x; q_1, \dots, q_k, \lambda) \Big|_{q_1 = \dots = q_k = q},$$

wobei $C_k(x; q_1, \dots, q_k, \lambda)$ eine (für festes x und λ) beschränkte, symmetrische Multilinearform auf $L_{\mathbb{C}}^2 \times \dots \times L_{\mathbb{C}}^2$ bezeichnet. $C_1(x; q, \lambda)$ ist also linear von q abhängig, $C_2(x; q, \lambda)$ quadratisch von q abhängig, usw.

Für $q = 0$ erhält man $C_0(x; \lambda) = c_{\lambda}(x)$. Formales Differenzieren der Potenzreihe nach x , Einsetzen in die Differentialgleichung und Gleichsetzen der Terme gleichen Grades in q führt auf ein Anfangswertproblem für $C_k(x; q, \lambda)$:

$$\begin{aligned} -C_k'' &= \lambda C_k - q C_{k-1} \\ C_k(0; q, \lambda) &= 0, \quad C_k'(0; q, \lambda) = 0, \quad k \geq 1. \end{aligned} \quad (2.19)$$

Nach Satz 2.8.1 folgt

$$C_k(x; q, \lambda) = \int_0^x s_{\lambda}(x-t) q(t) C_{k-1}(t; q, \lambda) dt,$$

woraus sich mit vollständiger Induktion

$$C_k(x; q, \lambda) = \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} c_{\lambda}(t_1) \prod_{l=1}^k (s_{\lambda}(t_{l+1} - t_l) q(t_l)) dt_1 \dots dt_k$$

ergibt. Wir haben dabei die in [39], Seite 6, eingeführte Schreibweise

$$\int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} dt_1 \dots dt_k := \int_{t_k=0}^x \int_{t_{k-1}=0}^{t_k} \dots \int_{t_1=0}^{t_2} dt_1 \dots dt_k$$

benutzt.

Aus den Abschätzungen

$$|c_\lambda(x)| = \frac{1}{2} |e^{i\sqrt{\lambda}x} + e^{-i\sqrt{\lambda}x}| \leq e^{|\operatorname{Im}\sqrt{\lambda}|x}$$

und

$$|s_\lambda(x)| = \left| \int_0^x c_\lambda(t) dt \right| \leq \int_0^x e^{|\operatorname{Im}\sqrt{\lambda}|t} dt \leq \int_0^x e^{|\operatorname{Im}\sqrt{\lambda}|x} dt \leq M e^{|\operatorname{Im}\sqrt{\lambda}|x}$$

für $x \in [0, M]$ erhält man

$$\begin{aligned} |C_k(x; q, \lambda)| &\leq \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} |c_\lambda(t_1)| \prod_{l=1}^k |s_\lambda(t_{l+1} - t_l) q(t_l)| dt_1 \dots dt_k \\ &\leq \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} e^{|\operatorname{Im}\sqrt{\lambda}|t_1} \prod_{l=1}^k (M e^{|\operatorname{Im}\sqrt{\lambda}|(t_{l+1} - t_l)} |q(t_l)|) dt_1 \dots dt_k \\ &= M^k e^{|\operatorname{Im}\sqrt{\lambda}|x} \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} \prod_{l=1}^k |q(t_l)| dt_1 \dots dt_k. \end{aligned}$$

Der Wert des Integrals ändert sich nicht, wenn man die Integrationsvariablen permutiert. Die Vereinigung der $k!$ Integrationsgebiete, die man durch Permutation der Integrationsvariablen erhalten kann, ergibt den k -dimensionalen Würfel $[0, x]^k$. Mit der Schwarzschen Ungleichung folgt

$$\begin{aligned} \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} \prod_{l=1}^k |q(t_l)| dt_1 \dots dt_k &= \frac{1}{k!} \int_{[0, x]^k} \prod_{l=1}^k |q(t_l)| dt_1 \dots dt_k \\ &= \frac{1}{k!} \left(\int_0^x |q(t)| dt \right)^k \leq \frac{1}{k!} (\|q\|_2 \sqrt{x})^k. \end{aligned}$$

Wir erhalten

$$|C_k(x; q, \lambda)| \leq \frac{M^k}{k!} e^{|\operatorname{Im}\sqrt{\lambda}|x} (\|q\|_2 \sqrt{x})^k,$$

womit die gleichmäßige Konvergenz der Potenzreihe auf beschränkten Teilmengen von $[0, M] \times \mathbb{C} \times L_{\mathbb{C}}^2$ gegen eine stetige Funktion y_1 bewiesen ist. Summation führt schließlich auf die Abschätzung

$$|y_1(x; q, \lambda)| \leq e^{|\operatorname{Im}\sqrt{\lambda}|x + M\|q\|_2\sqrt{x}} \quad \text{für alle } x \in [0, M].$$

Für $y_2(x; q, \lambda)$ gilt $C_0(x; \lambda) = s_\lambda(x)$. Alle anderen Schritte kann man wiederholen und erhält den M -fachen Wert als Schranke.

Wir führen nun die obigen Überlegungen nicht für das Nullpotential, sondern für die (fest gewählte) Entwicklungsstelle $q(x)$ durch. Außerdem ziehen wir uns im Blick auf die Anwendungen auf reellwertige stetige Potentiale auf dem Intervall $[0, \pi]$ (und reelle Parameter λ) zurück, und verwenden daher zur Abschätzung nicht die L^2 -Norm $\|\cdot\|_2$, sondern die Maximumnorm $\|\cdot\|_\infty$. Im wesentlichen ersetzen wir nur $c_\lambda(x)$ und $s_\lambda(x)$ durch die Funktionen $y_1(x; q, \lambda)$ und $y_2(x; q, \lambda)$ aus (2.18) (bei festem q und λ !).

Zur Vereinfachung der Darstellung schreiben wir im folgenden y_1 und y_2 anstelle von $y_1(x; q, \lambda)$ bzw. $y_2(x; q, \lambda)$.

Satz 2.8.2 *Seien $f \in C^0$ und $a, b \in \mathbb{R}$. Die eindeutige Lösung des Anfangswertproblems*

$$\begin{aligned} -y'' + q(x)y &= \lambda y - f(x) \\ y(0) &= a, \quad y'(0) = b \end{aligned}$$

ist

$$y(x) = ay_1(x) + by_2(x) + \int_0^x (y_1(t)y_2(x) - y_1(x)y_2(t)) f(t) dt.$$

Beweis: Pöschel und Trubowitz [39], Seite 12. □

Analog zum obigen Vorgehen betrachten wir die Anfangswertprobleme

$$\begin{aligned} &-y_j''(x; p, \lambda) + p(x)y_j(x; p, \lambda) = \lambda y_j(x; p, \lambda), \quad j = 1, 2 \\ \Leftrightarrow &-y_j''(x; p, \lambda) + q(x)y_j(x; p, \lambda) \\ &= \lambda y_j(x; p, \lambda) - (p(x) - q(x))y_j(x; p, \lambda) \\ y_1(0; p, \lambda) &= 1, \quad y_1'(0; p, \lambda) = 0 \\ y_2(0; p, \lambda) &= 0, \quad y_2'(0; p, \lambda) = 1. \end{aligned}$$

Für $j = 1, 2$ entwickeln wir $y_j(x; p, \lambda)$ in eine Potenzreihe nach $p - q$:

$$y_j(x; p, \lambda) = C_0(x; \lambda) + \sum_{k=1}^{\infty} C_k(x; p - q, \lambda).$$

Für $p = q$ erhält man $C_0(x; \lambda) = y_j(x)$. Durch formales Differenzieren der Potenzreihe nach x , Einsetzen in die Differentialgleichung wie oben und Gleichsetzen der Terme gleichen Grades in $p - q$ ergibt sich (mit $C_k = C_k(x; p - q, \lambda)$)

$$\begin{aligned} -C_k'' + q C_k &= \lambda C_k - (p - q) C_{k-1}, & k \geq 1. \\ C_k(0; p - q, \lambda) &= 0, \quad C_k'(0; p - q, \lambda) = 0 \end{aligned} \quad (2.20)$$

Nach Satz 2.8.2 folgt

$$C_k(x; p - q, \lambda) = \int_0^x (y_1(t)y_2(x) - y_1(x)y_2(t)) (p(t) - q(t)) C_{k-1}(t; p - q, \lambda) dt,$$

und mit vollständiger Induktion

$$\begin{aligned} C_k(x; p - q, \lambda) &= \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} y_j(t_1) \times \\ &\times \prod_{l=1}^k ((y_1(t_l)y_2(t_{l+1}) - y_1(t_{l+1})y_2(t_l)) (p(t_l) - q(t_l))) dt_1 \dots dt_k. \end{aligned}$$

Diese Formel ist mit der im Spezialfall $q \equiv 0$ gültigen Darstellung äquivalent, wenn man dort die Beziehung

$$s_\lambda(x - t) = c_\lambda(t)s_\lambda(x) - c_\lambda(x)s_\lambda(t)$$

berücksichtigt.

Mit $\|\cdot\| = \|\cdot\|_\infty$ erhalten wir aus der Abschätzung

$$\begin{aligned} |C_k(x; p - q, \lambda)| &\leq \int_{0 \leq t_1 \leq \dots \leq t_{k+1} = x} \|y_j\| \times \\ &\times \prod_{l=1}^k 2 \|y_1\| \|y_2\| \|p - q\| dt_1 \dots dt_k \end{aligned}$$

$$\begin{aligned}
&= \|y_j\| \frac{1}{k!} \left(\int_0^x 2 \|y_1\| \|y_2\| \|p - q\| dt \right)^k \\
&= \frac{1}{k!} \|y_j\| (2 \|y_1\| \|y_2\| \|p - q\| x)^k
\end{aligned}$$

durch Summation

$$|y_j(x; p, \lambda)| \leq \|y_j\| e^{2 \|y_1\| \|y_2\| \|p - q\| x} \quad (2.21)$$

(mit $y_j = y_j(x; q, \lambda)$) für alle $x \in [0, \pi]$.

2.9 Konvergenz

Mit der letzten Formel haben wir das notwendige Hilfsmittel bereitgestellt, um die Lipschitz-Konstante der ersten Ableitung von f in einer Umgebung der Näherungslösung $\tilde{q}(x)$ abschätzen zu können. Wir erhalten eine geeignete Konstante durch Abschätzen der Norm der zweiten Ableitung von λ_i , die nach (2.14) durch die Bilinearform B_i gegeben ist. Die Normsymbole stehen in diesem Abschnitt bei Vektoren und Funktionen für die Maximumnorm, bei Matrizen für die Zeilensummennorm und bei B_i für die Operatornorm nach (2.15).

Satz 2.9.1 $q(x)$ sei eine beliebige (fest gewählte) auf $[0, \pi]$ stetige Funktion und $\kappa > 0$. λ_i bezeichne den i -ten Eigenwert des zu $q(x)$ gehörigen Dirichlet-problems. Dann gilt für alle $p \in D := \{r \in C^0[0, \pi] : \|r - q\| \leq \kappa\}$:

$$\|B_i(p)\| \leq 16 \pi \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa \|y_{1i}\| \|y_{2i}\|},$$

wobei y_{1i} und y_{2i} die Lösungen der Anfangswertprobleme

$$\begin{aligned}
-y_{ji}'' + q(x) y_{ji} &= \lambda_i y_{ji}, & j &= 1, 2 \\
y_{1i}(0) &= 1, & y_{1i}'(0) &= 0 \\
y_{2i}(0) &= 0, & y_{2i}'(0) &= 1
\end{aligned} \quad (2.22)$$

bezeichnen.

Beweis: Es sei $p(x) \in D$ beliebig, aber fest gewählt. Für festes i besitzt das Potential

$$\tilde{p}(x) := p(x) + \lambda_i(q) - \lambda_i(p)$$

den i -ten Eigenwert $\lambda_i(\tilde{p}) = \lambda_i(q)$. Nach Korollar 1.1.5 zum Monotoniesatz gilt

$$|\lambda_i(p) - \lambda_i(q)| \leq \|p - q\| \quad \text{für alle } i \in \mathbb{N}_0,$$

woraus

$$\|\tilde{p} - q\| \leq 2 \|p - q\|$$

folgt.

z_{1i} und z_{2i} seien die Lösungen der Anfangswertprobleme

$$-z_{ji}'' + p(x) z_{ji} = \lambda_i(p) z_{ji} \quad (2.23)$$

$$\Leftrightarrow -z_{ji}'' + \tilde{p}(x) z_{ji} = \lambda_i(q) z_{ji} \quad (2.24)$$

$$z_{1i}(0) = 1, \quad z_{1i}'(0) = 0$$

$$z_{2i}(0) = 0, \quad z_{2i}'(0) = 1.$$

Aus der Äquivalenz von (2.23) und (2.24) erhalten wir

$$B_i(p) = B_i(\tilde{p}),$$

und mit Satz 2.7.3 die Abschätzung

$$\|B_i(\tilde{p})\| = \|B_i(p)\| \leq 16 \pi \|z_{1i}\| \|z_{2i}\|.$$

Wegen $\lambda_i(\tilde{p}) = \lambda_i(q)$ folgt aus (2.21) für $j = 1, 2$

$$|z_{ji}(x)| \leq \|y_{ji}\| e^{2 \|y_{1i}\| \|y_{2i}\| \|\tilde{p}-q\| x}.$$

Damit ergibt sich schließlich

$$\begin{aligned} \|B_i(p)\| &\leq 16 \pi \|z_{1i}\| \|z_{2i}\| \\ &\leq 16 \pi \|y_{1i}\| \|y_{2i}\| e^{4 \|y_{1i}\| \|y_{2i}\| \|\tilde{p}-q\| \pi} \\ &\leq 16 \pi \|y_{1i}\| \|y_{2i}\| e^{8\pi \|y_{1i}\| \|y_{2i}\| \|p-q\|} \\ &\leq 16 \pi \|y_{1i}\| \|y_{2i}\| e^{8\pi \kappa \|y_{1i}\| \|y_{2i}\|}. \end{aligned}$$

□

Wir wenden die erhaltenen Ergebnisse nun auf die untersuchte Funktion f an. Aus dem Beweis von Satz 2.2.3 folgt

$$\frac{\partial^2 f_i}{\partial a_k \partial a_m}(a) = d_q^2 \lambda_i(q_k, q_m) = B_i(q(x; a))(q_k, q_m).$$

Mit Hilfe des Mittelwertsatzes schätzen wir die Lipschitz-Konstante von f' in einer Umgebung des Nullpunkts des \mathbb{R}^{n+1} ab. In der anschließenden Rechnung verwenden wir die folgenden Bezeichnungen:

$g_i(x; a)$ sei die normierte i -te Eigenfunktion des Potentials $q(x; a)$. y_{1i}, y_{2i} seien die Lösungen der Anfangswertprobleme (2.22) für $q = \hat{q}$ aus (2.1) und $\lambda = \hat{\lambda}_i$, wobei $\hat{\lambda}_i$ den i -ten Eigenwert des zu \hat{q} gehörigen Dirichletproblems bezeichnet. κ sei eine beliebige positive reelle Zahl.

Wir setzen

$$\delta := \sum_{j=0}^n \|q_j\|$$

und

$$D := \{a \in \mathbb{R}^{n+1} : \|a\| \leq \kappa\}.$$

Dann gilt für alle $a \in D$

$$\|q(x; a) - \hat{q}(x)\| = \left\| \sum_{j=0}^n a_j q_j(x) \right\| \leq \|a\| \sum_{j=0}^n \|q_j\| \leq \kappa \delta.$$

Da die Verbindungsstrecke zweier Punkte $a, b \in D$ ebenfalls ganz in D liegt, erhalten wir für die Lipschitzkonstante von f' in D die folgende Abschätzung:

$$\begin{aligned} \|f'(a) - f'(b)\| &= \left\| \left(\int_0^\pi (g_i^2(x; a) - g_i^2(x; b)) q_j dx \right)_{ij} \right\| \\ &= \max_{i=0}^n \sum_{j=0}^n \left| \int_0^\pi (g_i^2(x; a) - g_i^2(x; b)) q_j(x) dx \right| \\ &= \max_{i=0}^n \sum_{j=0}^n \left| (d_q \lambda_i(q(x; a)) - d_q \lambda_i(q(x; b))) q_j \right| \end{aligned}$$

$$\begin{aligned}
&\leq \max_{i=0}^n \|d_q \lambda_i(q(x; a)) - d_q \lambda_i(q(x; b))\| \sum_{j=0}^n \|q_j\| \\
&\leq \delta \max_{i=0}^n \sup_{0 \leq t \leq 1} \|d_q^2 \lambda_i(q(x; a + t(b-a)))\| \|q(x; a) - q(x; b)\| \\
&= \delta \|q(x; a) - q(x; b)\| \max_{i=0}^n \sup_{0 \leq t \leq 1} \|B_i(q(x; a + t(b-a)))\| \\
&\leq \delta \|q(x; a) - q(x; b)\| \max_{i=0}^n 16 \pi \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa\delta \|y_{1i}\| \|y_{2i}\|} \\
&\leq 16 \pi \delta \left\| \sum_{k=0}^n (a_k - b_k) q_k(x) \right\| \max_{i=0}^n \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa\delta \|y_{1i}\| \|y_{2i}\|} \\
&\leq 16 \pi \delta \|a - b\| \left(\sum_{j=0}^n \|q_j\| \right) \max_{i=0}^n \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa\delta \|y_{1i}\| \|y_{2i}\|} \\
&= 16 \pi \delta^2 \|a - b\| \max_{i=0}^n \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa\delta \|y_{1i}\| \|y_{2i}\|} \\
&= \gamma \|a - b\|
\end{aligned}$$

mit

$$\gamma := 16 \pi \delta^2 \max_{i=0}^n \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa\delta \|y_{1i}\| \|y_{2i}\|}.$$

Nach diesen Vorbereitungen sind wir in der Lage, den folgenden Konvergenzsatz zu beweisen:

Satz 2.9.2 *Wir betrachten das inverse Dirichletproblem mit den vorgegebenen Eigenwerten $\{\nu_i\}_{i=0}^n$.*

Dazu sei ein Näherungspotential $\tilde{q}(x)$ mit Eigenwerten $\{\tilde{\lambda}_i\}_{i=0}^n$ gegeben. Wir wählen die Basisfunktionen

$$\begin{aligned}
\hat{q}(x) &:= \tilde{q}(x), \\
q_j(x) &:= -2 \frac{d}{dx} v_j, \quad j = 0, 1, \dots, n
\end{aligned}$$

mit $v_j(x) = y_{1j}(x; \tilde{q}, \tilde{\lambda}_j) y_{2j}(x; \tilde{q}, \tilde{\lambda}_j)$, wobei y_{1j}, y_{2j} die Lösungen der Anfangswertprobleme (2.22) für $q = \tilde{q}$ und $\lambda = \tilde{\lambda}_j$ bezeichnen.

Es seien

$$\delta := \sum_{j=0}^n \|q_j\|,$$

$$\kappa > \frac{1}{16 \pi \delta^2 \max_{i=0}^n \|y_{1i}\| \|y_{2i}\|},$$

$$\gamma := 16 \pi \delta^2 \max_{i=0}^n \|y_{1i}\| \|y_{2i}\| e^{8\pi\kappa\delta \|y_{1i}\| \|y_{2i}\|}$$

und

$$D := \{a \in \mathbb{R}^{n+1} : \|a\| \leq \kappa\}.$$

Dann gilt: Ist

$$\max_{i=0}^n |\nu_i - \tilde{\lambda}_i| < \frac{1}{2\gamma},$$

so konvergieren die iterierten Potentiale des Newtonverfahrens (2.1) mit Startvektor $a^{(0)} = 0$ gegen eine Lösung $q^*(x) = q(x; a^*)$ des inversen Dirichletproblems, und es gilt die Abschätzung

$$\|a^* - a^{(1)}\| \leq 2\gamma \|a^{(1)}\|^2.$$

Beweis: Für die gewählten Ansatzfunktionen erhalten wir als Funktionalmatrix im Nullpunkt des \mathbb{R}^{n+1} die Einheitsmatrix, woraus

$$\beta = \|(f'(a^{(0)}))^{-1}\| = 1,$$

$$\eta = \|(f'(a^{(0)}))^{-1} f(a^{(0)})\| = \|f(a^{(0)})\| = \max_{i=0}^n |\nu_i - \tilde{\lambda}_i| < \frac{1}{2\gamma}$$

folgen.

Nach der im Anschluß an Satz 2.9.1 durchgeführten Rechnung ist γ eine obere Schranke der Lipschitz-Konstanten von f' in D . Also gilt

$$\beta\gamma\eta = \gamma\eta < \frac{1}{2}.$$

Weiter liegt

$$S := \{a \in \mathbb{R}^{n+1} : \|a\| \leq \frac{1 - \sqrt{1 - 2\gamma\eta}}{\gamma} =: \varrho\}$$

wegen

$$\varrho < \frac{1}{\gamma} = \frac{1}{16 \pi \delta^2 \max_{i=0}^n \|y_{1i}\| \|y_{2i}\| e^{(\dots)}} < \frac{1}{16 \pi \delta^2 \max_{i=0}^n \|y_{1i}\| \|y_{2i}\|} < \kappa$$

ganz in D . Damit folgt die Behauptung aus dem Satz von Kantorowitsch. \square

Wir betrachten nun das Intervall-Newtonverfahren für die Funktion f aus (2.3). In den folgenden beiden Abschnitten stellen wir zuerst die Grundlagen der Intervallrechnung und das Intervall-Newtonverfahren selbst vor.

2.10 Intervallrechnung

In der Intervallrechnung wird mit Intervallen anstelle von Zahlen gerechnet. Sie ist ein Werkzeug, um auf einfache Weise simultane arithmetische Operationen für alle Elemente gegebener Zahlenintervalle durchzuführen. Ausführliche Darstellungen der Intervallrechnung finden sich in den Büchern von Alefeld und Herzberger [4], [5], Moore [34] und Neumaier [36].

Die reelle Intervallarithmetik ist für kompakte reelle Intervalle definiert. Derartige Intervalle schreiben wir als

$$[a] := [\underline{a}, \bar{a}] := \{a \in \mathbb{R} : \underline{a} \leq a \leq \bar{a}\}.$$

Wir übertragen die arithmetischen Operationen $+$, $-$, \cdot , $/$ von reellen Zahlen auf Intervalle durch die Vorschrift

$$[\underline{a}, \bar{a}] * [\underline{b}, \bar{b}] := \{a * b : a \in [\underline{a}, \bar{a}], b \in [\underline{b}, \bar{b}]\}$$

mit $*$ $\in \{+, -, \cdot, /\}$ und $0 \notin [\underline{b}, \bar{b}]$ im Fall der Division. Als Ergebnis erhalten wir das Intervall aller reellen Zahlen, die bei paarweiser Verknüpfung sämtlicher Elemente aus $[\underline{a}, \bar{a}]$ und $[\underline{b}, \bar{b}]$ entstehen. Die reellen Intervalle mit den oben definierten Verknüpfungen nennen wir *Raum der reellen Intervalle* und bezeichnen diesen mit \mathbb{IR} . Reelle Zahlen a identifizieren wir mit Punktintervallen (entarteteten Intervallen) $[a, a]$. Damit sind Verknüpfungen reeller Zahlen mit Intervallen erklärt.

Wesentlich für die Anwendbarkeit der Intervallrechnung zur garantierten Einschließung von Lösungen von gegebenen numerischen Aufgaben ist die *Inklusionsmonotonie* (Teilmengeneigenschaft) der Intervalloperationen. Es gilt für die Verknüpfungen $*$ $\in \{+, -, \cdot, /\}$

$$[\underline{a}, \bar{a}] \subseteq [\underline{c}, \bar{c}], [\underline{b}, \bar{b}] \subseteq [\underline{d}, \bar{d}] \Rightarrow [\underline{a}, \bar{a}] * [\underline{b}, \bar{b}] \subseteq [\underline{c}, \bar{c}] * [\underline{d}, \bar{d}],$$

wobei bei der Division $0 \notin [\underline{d}, \bar{d}]$ vorausgesetzt wird. Diese Eigenschaft führt dazu, daß sich Einschließungen $[\underline{c}, \bar{c}]$ für $[\underline{a}, \bar{a}]$ bzw. $[\underline{d}, \bar{d}]$ für $[\underline{b}, \bar{b}]$ auf den Ausdruck $[\underline{a}, \bar{a}] * [\underline{b}, \bar{b}]$ übertragen.

Der Raum der Intervalle wird zum vollständigen metrischen Raum, wenn man den *Abstand*

$$q([a], [b]) := \max\{|\underline{b} - \underline{a}|, |\bar{b} - \bar{a}|\}$$

eingührt. Grenzwerte von Intervallfolgen im Sinne dieser Metrik sind äquivalent zur Konvergenz der Intervallgrenzen in \mathbb{R} . Der *Betrag* eines Intervalls ist gegeben durch

$$|[a]| := q([a], 0) = \max\{|\underline{a}|, |\bar{a}|\}.$$

Intervallvektoren und Intervallmatrizen ergeben sich aus reellen Vektoren und Matrizen, indem man Intervalle als Vektorkomponenten bzw. Matrixelemente zuläßt. Die Räume der Intervallvektoren und Intervallmatrizen bezeichnen wir mit $I\mathbb{R}^n$ bzw. $I\mathbb{R}^{n \times n}$. Wie im eindimensionalen Fall identifizieren wir reelle Vektoren und Matrizen mit Punkt(intervall)vektoren bzw. Punkt(intervall)matrizen.

Lineare Gleichungssysteme mit einer Intervallmatrix $[A]$ als Koeffizientenmatrix und einem Intervallvektor $[b]$ als rechter Seite interpretieren wir als simultan gestellte reelle Gleichungssysteme und suchen einen Intervallvektor

$$[x] \supseteq \{x : Ax = b, A \in [A], b \in [b]\}, \quad (2.25)$$

wobei i.a. nur $[A] \cdot [x] \supseteq [b]$ gilt.

Ein solcher Vektor kann mit dem Intervall-Gauß-Algorithmus IGA, der aus der formalen Anwendung des Gauß-Algorithmus auf das Paar $([A], [b])$ besteht (siehe Alefeld und Herzberger [5], Kapitel 15), berechnet werden. Wir bezeichnen diesen Vektor mit

$$\text{IGA}([A], [b])$$

und nennen $\text{IGA}([A], [b])$ *durchführbar*, falls bei der Anwendung des Gauß-Algorithmus keine Division durch ein Intervall auftritt, das die Null enthält.

Eine Einschließung der inversen Matrizen aller $A \in [A]$ enthält man durch „formales Invertieren“ von $[A]$ mit Hilfe des Gauß-Algorithmus (was äquivalent ist zur spaltenweisen Berechnung von $\text{IGA}([A], e^i)$, wobei e^i den i -ten Einheitsvektor im \mathbb{R}^n bezeichnet). Wir verwenden für diese Einschließung die Schreibweise $\text{IGA}([A])$. Falls der Gauß-Algorithmus auf $[A]$ anwendbar ist, heißt $\text{IGA}([A])$ *durchführbar*.

Ähnlich wie die arithmetischen Grundoperationen können auch reellwertige Funktionen intervallmäßig erweitert werden. Im folgenden verstehen wir unter der Intervallauswertung $f([x])$ einer Funktion f auf dem Intervall $[x]$ ein (durch einen vorgeschriebenen Algorithmus berechnetes) Intervall, das den Wertebereich von f im Intervall $[x]$ enthält (wir setzen voraus, daß der Wertebereich beschränkt ist und die Funktionswerte numerisch berechnet oder eingeschlossen werden können). Algorithmen zur Intervallauswertung findet man z.B. im Buch von Ratschek und Rokne [40].

Zuletzt führen wir noch den Begriff der *inklusionsmonotonen* Intervallauswertung ein:

Definition 2.10.1 *Eine Intervallauswertung $f([x])$ einer Funktion f auf einem Intervall $I \subseteq \mathbb{R}$ heißt **inklusionsmonoton**, wenn für alle $[x], [y] \subseteq I$ gilt:*

$$[x] \subseteq [y] \Rightarrow f([x]) \subseteq f([y]). \quad (2.26)$$

Die Übertragung der Intervallauswertung auf mehrdimensionale Funktionen erfolgt komponentenweise. Offensichtlich treten dabei keine zusätzlichen Schwierigkeiten auf, so daß wir diesen Fall nicht gesondert behandeln müssen.

Funktionen, die als endliche Linearkombination gegebener Funktionen w_i , $i = 1, \dots, m$, geschrieben werden können, erweitern wir zu *Intervallfunktionen*, indem wir anstelle reeller Koeffizienten Intervalle zulassen:

$$\begin{aligned} [w](x) &:= \sum_{i=1}^m [a_i] w_i(x) \\ &:= \{ w(x) : w(x) = \sum_{i=1}^m a_i w_i(x), \quad a_i \in [a_i] \text{ für } i = 1, \dots, m \}. \end{aligned}$$

Sind alle Funktionen w_i differenzierbar oder stückweise differenzierbar mit endlich vielen Bruchstellen, so definieren wir die (stückweise) Ableitung $[w']$ von $[w]$ durch

$$[w'](x) := \{ w'(x) : w(x) \in [w](x) \}.$$

Bei dieser Festlegung gilt

$$[w'](x) = \sum_{i=1}^m [a_i] w'_i(x),$$

d.h. die Intervallkoeffizienten verhalten sich bei der Differentiation wie reelle Konstanten.

2.11 Das Intervall-Newtonverfahren

Das Intervall-Newtonverfahren berechnet eine Folge von Intervall(vektor)en $[x]^{(k)}$, die unter gewissen Voraussetzungen gegen eine Nullstelle von f konvergieren. Von Alefeld wurden in [2] und [3] Konvergenzaussagen für das Intervall-Newtonverfahren formuliert. Da wir einige der Beweisideen später verwenden werden, geben wir die Sätze zum Teil mit Beweisen an.

Es wird dabei von der folgenden Darstellung des Mittelwertsatzes 2.1.2 im \mathbb{R}^n Gebrauch gemacht (siehe auch Ortega und Rheinbold [38], Seite 68f):

Satz 2.11.1 *Es sei D eine offene, konvexe Teilmenge des \mathbb{R}^n , die Funktion $f : D \rightarrow \mathbb{R}^n$ sei differenzierbar in ganz D . Dann gibt es für alle $y, z \in D$ eine Matrix $B(y, z) \in \mathbb{R}^{n \times n}$, so daß*

$$f(z) - f(y) = B(y, z)(z - y) \quad (2.27)$$

gilt.

Bemerkung: Nach Satz 2.1.2 kann die Matrix B in der Form

$$B(y, z) = \int_0^1 f'(y + t(z - y)) dt.$$

dargestellt werden. B ist stetig von y und z abhängig und stimmt i.a. nicht mit der Ableitung von f an einer Zwischenstelle überein. Falls aber $f'([x])$ für ein $[x] \subset D$ existiert, so gilt

$$B(y, z) \in f'([x]) \quad \text{für alle } y, z \in [x].$$

□

Wir wollen nun die Aufgabe lösen, zu einer gegebenen Funktion

$$f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$$

im Intervall $[x]^{(0)} \subset D$ eine Nullstelle zu bestimmen. Wir setzen voraus, daß f in D differenzierbar ist, und daß auf $[x]^{(0)}$ eine inklusionsmonotone Intervallauswertung $f'([x])$ der Ableitung von f existiert. Außerdem sei für alle $[x] \subseteq [x]^{(0)}$

$$m : [x] \mapsto \mathbb{R}^n, \quad m([x]) \in [x]$$

eine Auswahlfunktion.

Dann ist das *Intervall-Newtonverfahren* (IN) zur Bestimmung einer Nullstelle x^* von f in $[x]^{(0)}$ durch die folgende Iteration gegeben:

Für $k = 0, 1, \dots$:

$$(IN) \begin{cases} m^{(k)} & := m([x]^{(k)}) \\ \text{IN}([x]^{(k)}) & := m^{(k)} - \text{IGA}(f'([x]^{(k)}), f(m^{(k)})) \\ [x]^{(k+1)} & := \text{IN}([x]^{(k)}) \cap [x]^{(k)}. \end{cases}$$

Wir beweisen zunächst einen Existenzsatz (siehe Alefeld [3]):

Satz 2.11.2 1. Falls $\text{IGA}(f'([x]^{(0)}))$ durchführbar ist und

$$\text{IN}([x]^{(0)}) \subseteq [x]^{(0)} \quad (2.28)$$

gilt, besitzt f in $\text{IN}([x]^{(0)})$ eine in $[x]^{(0)}$ eindeutige Nullstelle.

2. Falls $[x]^{(0)} \cap \text{IN}([x]^{(0)}) = \emptyset$, dann besitzt f in $[x]^{(0)}$ keine Nullstelle.

Beweis: Da $\text{IGA}(f'([x]^{(0)}))$ durchführbar ist, sind alle Punktmatrizen aus $f'([x]^{(0)})$, insbesondere $B(x, m^{(0)})$ für alle $x \in [x]^{(0)}$ nichtsingulär.

Zum Nachweis einer Nullstelle von f in $[x]^{(0)}$ zeigen wir unter der Voraussetzung (2.28), daß die Abbildung

$$p : [x]^{(0)} \rightarrow \mathbb{R}^n, \quad p(x) := x - B(x, m^{(0)})^{-1} f(x) \quad (2.29)$$

in $[x]^{(0)}$ einen Fixpunkt besitzt. Mit (2.27) gilt für alle $x \in [x]^{(0)}$:

$$\begin{aligned} p(x) &= x - B(x, m^{(0)})^{-1} f(m^{(0)}) + B(x, m^{(0)})^{-1} (f(m^{(0)}) - f(x)) \\ &= x - B(x, m^{(0)})^{-1} f(m^{(0)}) + m^{(0)} - x \\ &\in m^{(0)} - \text{IGA}(f'([x]^{(0)}), f(m^{(0)})) = \text{IN}([x]^{(0)}). \end{aligned}$$

Falls (2.28) erfüllt ist, wird durch p eine stetige Selbstabbildung auf $[x]^{(0)}$ definiert, die nach dem Fixpunktsatz von Brouwer in $[x]^{(0)}$ einen Fixpunkt x^* besitzt. Da $B(x^*, m^{(0)})^{-1}$ nichtsingulär ist, folgt aus (2.29), daß x^* eine Nullstelle von f ist.

Enthält $[x]^{(0)}$ zwei Nullstellen $x_1^* \neq x_2^*$ von f , dann folgt mit Satz 2.11.1

$$0 = f(x_2^*) - f(x_1^*) = B(x_1^*, x_2^*)(x_2^* - x_1^*).$$

Diese Gleichung kann nur erfüllt werden, wenn die Matrix $B(x_1^*, x_2^*) \in f'([x]^{(0)})$ singulär ist, im Widerspruch zur Durchführbarkeit von IGA($f'([x]^{(0)})$). Damit ist die Eindeutigkeit von x^* in $[x]^{(0)}$ bewiesen.

Um die zweite Aussage des Satzes zu beweisen, nehmen wir an, daß die Funktion f in $[x]^{(0)}$ eine Nullstelle x^* besitzt. Wieder gilt mit der oben definierten Funktion $p(x)$

$$p(x^*) = x^* - B(x^*, m^{(0)})^{-1}f(x^*) = x^* \in \text{IN}([x]^{(0)}),$$

und wegen $x^* \in [x]^{(0)}$ führt $[x]^{(0)} \cap \text{IN}([x]^{(0)}) = \emptyset$ zum Widerspruch. \square

Korollar 2.11.3 *Gilt für ein $k \in \mathbb{N}_0$ in der Intervall-Newtoniteration*

$$\text{IN}([x]^{(k)}) \subseteq [x]^{(k)},$$

so liegt in $[x]^{(k+1)}$ eine (in $[x]^{(0)}$ eindeutige) Lösung von $f(x) = 0$.

Abschließend formulieren wir den folgenden Konvergenzsatz aus der Arbeit von Alefeld [3]:

Satz 2.11.4 *Gegeben sei eine differenzierbare Abbildung $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$. Zur Ableitung f' existiere eine auf $[x]^{(0)} \subset D$ inklusionsmonotone Intervallauswertung $f'([x])$, und IGA($f'([x]^{(0)})$) sei durchführbar. Weiter setzen wir*

$$A := |I - \text{IGA}(f'([x]^{(0)}))f'([x]^{(0)})|, \quad (2.30)$$

wobei I die Einheitsmatrix bezeichnet und der Betrag in jeder Komponente zu bilden ist.

Falls f in $[x]^{(0)}$ eine Nullstelle x^ besitzt und der Spektralradius von A kleiner als 1 ist, dann ist die durch (IN) berechnete Folge wohldefiniert und es gilt $\lim_{k \rightarrow \infty} [x]^{(k)} = x^*$.*

2.12 Rekonstruktion mit dem Intervall-Newtonverfahren

Wenn man das Intervall-Newtonverfahren auf die untersuchte Funktion f aus (2.3) anwendet, erhält man ein Intervallverfahren zur Behandlung des inversen Dirichletproblems, das durch den nachstehenden Algorithmus 2.2 beschrieben wird.

Für dieses Verfahren ergibt sich aus Satz 2.11.3:

Satz 2.12.1 *Gilt bei der Durchführung des Rekonstruktionsverfahrens 2.2 für ein $k \in \mathbb{N}$*

$$\text{IN}([a]^{(k)}) \subseteq [a]^{(k)}, \quad (2.31)$$

so enthält $[a]^{(k+1)}$ genau eine Lösung a^* der Gleichung $f(a) = 0$. Die zugehörige Intervallfunktion $[q]^{(k+1)}(x)$ schließt genau ein Potential der Gestalt (2.1), nämlich

$$q^*(x) := q(x; a^*)$$

ein, das die vorgegebenen Eigenwerte $\{\nu_i\}_{i=0}^n$ besitzt.

Algorithmus 2.2: Rekonstruktion für das inverse Dirichletproblem
Intervall-Newtonverfahren im $I\mathbb{R}^{n+1}$

1. Wähle symmetrische Basisfunktionen $\widehat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$.
2. Wähle $[a]^{(0)} \in I\mathbb{R}^{n+1}$.
3. Für $k = 0, 1, \dots$:

(i) Für $i = 0, 1, \dots, n$:

(a) Berechne $f_i(m^{(k)}) = \lambda_i(q^{(k)}) - \nu_i$,

(b) Berechne $g_i(x; [a]^{(k)}) = g_i(x; [q]^{(k)}(x))$,

wobei

$$m^{(k)} := m([a]^{(k)}),$$

$$q^{(k)}(x) := q(x; m^{(k)}) := \widehat{q}(x) + \sum_{j=0}^n m_j^{(k)} q_j(x),$$

$$[q]^{(k)}(x) := q(x; [a]^{(k)}) := \widehat{q}(x) + \sum_{j=0}^n [a_j]^{(k)} q_j(x).$$

(ii) Berechne die Funktionalmatrix

$$\left(\frac{\partial f_i}{\partial a_j}([a]^{(k)}) \right) = \left(\int_0^\pi g_i^2(x; [a]^{(k)}) q_j(x) dx \right),$$

$i, j = 0, 1, \dots, n.$

(iii) Führe den Intervall-Newtonschritt

$$\text{IN}([a]^{(k)}) := m^{(k)} - \text{IGA}\left(\left(\frac{\partial f_i}{\partial a_j}([a]^{(k)})\right), f(m^{(k)})\right),$$

$$[a]^{(k+1)} := \text{IN}([a]^{(k)}) \cap [a]^{(k)}$$

durch.

Kapitel 3

Numerische Realisierung

In diesem Kapitel beschreiben wir die numerischen Algorithmen der einzelnen Teilschritte der Rekonstruktionsverfahren 2.1 bzw. 2.2 sowie die Implementierung auf dem Rechner.

Da für den Funktionswert $f(a^{(k)})$ im k -ten Iterationsschritt von Algorithmus 2.1 das direkte Dirichletproblem für das Potential $q^{(k)}(x)$ zu lösen ist, benötigen wir zuerst ein Berechnungsverfahren für Eigenwerte des Dirichletproblems. Dies ist gleichzeitig der aufwendigste Teil im gesamten Rekonstruktionsverfahren.

Wir werden drei unterschiedliche Implementierungen des Rekonstruktionsverfahrens beschreiben, zwei näherungsweise Realisierungen von Algorithmus 2.1, sowie eine Realisierung des Intervallverfahrens 2.2, die auf dem Rechner unter Berücksichtigung aller Fehler garantierte Lösungseinschlüsse berechnet.

Dabei sind u.a. die folgenden Teilprobleme zu lösen:

- Lineare Gleichungssysteme.
- Die Berechnung von Eigenwerten und Eigenvektoren symmetrischer Matrizen.
- Die numerische Integration eindimensionaler, reellwertiger stetiger Funktionen.
- Die numerische Lösung von Anfangswertproblemen gewöhnlicher Differentialgleichungen.

Algorithmen zur Behandlung dieser Aufgaben finden sich z.B. in den Büchern von Stoer [44] sowie Stoer und Bulirsch [45]. Wir gehen nur auf einige speziell an das Rekonstruktionsverfahren angepaßte Varianten näher ein.

Im verbleibenden Teil dieser Arbeit setzen wir voraus, daß $q(x)$ eine auf dem Intervall $[0, \pi]$ symmetrische, stetige Funktion ist.

3.1 Näherungslösungen mit Hilfe des Ritz-Verfahrens

Falls $q(x) > 0$ auf $[0, \pi]$ gilt, so gehört das Dirichletproblem zu einer Klasse von Eigenwertproblemen, für die das Rayleighsche Minimalprinzip anwendbar ist (siehe z.B. Collatz [11], Seite 211ff), und bei denen sich mit dem Ritzschen Ansatz ([11], Seite 214ff) Näherungen (genauer: obere Schranken) der kleinsten Eigenwerte berechnen lassen. Ohne tiefer in die zugrundliegende Theorie einzudringen, beschreiben wir im folgenden kurz das Ritz-Verfahren (häufig auch als Rayleigh-Ritz-Verfahren bezeichnet) für das Dirichletproblem.

Beim Ritz-Verfahren für das Dirichletproblem sind zunächst mit r linear unabhängigen Funktionen (sogenannten *Vergleichsfunktionen*) $u_i(x)$, $i = 1, 2, \dots, r$, Matrixelemente

$$m_{ij} = \int_0^\pi (-u_i'' + q(x) u_i) u_j dx, \quad n_{ij} = \int_0^\pi u_i u_j dx, \\ i, j = 1, 2, \dots, r$$

zweier Matrizen $M = (m_{ij})$, $N = (n_{ij})$ zu berechnen, für die danach das allgemeine Matrizen-Eigenwertproblem

$$Mx = \mu Nx \quad (3.1)$$

zu lösen ist.

Als Vergleichsfunktionen sind hier alle Funktionen zugelassen, die auf $[0, \pi]$ zweimal stetig differenzierbar sind und den Dirichletschen Randbedingungen genügen. Verwendet man normierte und paarweise orthogonale Vergleichsfunktionen, so ist N die Einheitsmatrix, und man erhält das gewöhnliche Matrizen-Eigenwertproblem

$$Mx = \mu x. \quad (3.2)$$

Sortiert man die erhaltenen Eigenwerte μ_i , $i = 1, 2, \dots, r$, von (3.1) bzw. (3.2) der Größe nach, so ist jeweils μ_i eine Obergrenze für den $i - 1$ -ten Eigenwert λ_{i-1} des Dirichletproblems zu $q(x)$ (zum Beweis siehe z.B. [11], Seite 228ff).

Die Approximation der λ_{i-1} durch die μ_i ist umso genauer, je besser sich die Eigenfunktionen des Dirichletproblems durch Linearkombinationen der Vergleichsfunktionen approximieren lassen (wählt man speziell die – unbekannt – Eigenfunktionen als Vergleichsfunktionen, so erhält man für M eine Diagonalmatrix aus den Eigenwerten des Dirichletproblems). Eine naheliegende Wahl ist

$$u_i(x) = \sqrt{\frac{2}{\pi}} \sin ix, \quad i = 1, 2, \dots, r.$$

Diese Vergleichsfunktionen sind paarweise orthogonal zueinander, sie approximieren die Eigenfunktionen des Dirichletproblems im Sinne der asymptotischen Entwicklung (1.6), und ihre Nullstellen trennen sich wie die Nullstellen der Eigenfunktionen.

Bemerkung:

1. Bei der Anwendung des Ritz-Verfahrens auf das Dirichletproblem kann die Bedingung $q(x) > 0$ fallengelassen werden. Andernfalls addiert man zu $q(x)$ eine geeignete Konstante c und wendet das Ritz-Verfahren auf die transformierte Eigenwertaufgabe mit um c erhöhten Eigenwerten an.
2. Ist $q(x)$ symmetrisch, sind auch die Eigenfunktionen mit geradem Index symmetrisch (siehe Pöschel und Trubowitz [39], Seite 41), wohingegen für die Eigenfunktionen mit ungeradem Index die Beziehung

$$u(x) = -u(\pi - x) \quad \text{für alle } x \in [0, \pi]$$

gilt (sie sind *ungerade* bzgl. dem Intervallmittelpunkt). In diesem Fall kann man das Ritz-Verfahren getrennt auf die Eigenwerte mit geradem und mit ungeradem Index anwenden (siehe Sauer und Szabó [43], Seite 486). Obergrenzen für die Eigenwerte mit geradem Index erhält man durch symmetrische Vergleichsfunktionen beim Ritz-Verfahren, wohingegen ungerade Vergleichsfunktionen Obergrenzen für die Eigenwerte mit ungeradem Index ergeben. Statt eines Matrizen-Eigenwertproblems der Dimension r sind (für gerades r) zwei Eigenwertprobleme der Dimension $\frac{r}{2}$ zu lösen, was den Aufwand i.a. erheblich vermindert. \square

Mit Hilfe des Ritz-Verfahrens lassen sich nicht nur Näherungen für die Eigenfunktionen berechnen, sondern sogar das gesamte Rekonstruktionsverfahren 2.1 approximativ lösen (siehe auch Hald [19]). Ist $z^i = (z_1^i, \dots, z_r^i)^T$ ein geeignet normierter Eigenvektor zu demjenigen Eigenwert der Matrix M , der den Dirichlet-Eigenwert λ_i approximiert, erhalten wir

$$\tilde{y}_i(x) = \sum_{j=1}^r z_j^i u_j(x)$$

als Näherung der i -ten Eigenfunktion des Dirichletproblems. Die Komponenten der Funktionalmatrix im Algorithmus 2.1 können dann durch die Integrale

$$\int_0^\pi \tilde{y}_i^2(x) q_l(x) dx = \sum_{j,k=1}^r z_j^i z_k^i \int_0^\pi u_j(x) u_k(x) q_l(x) dx, \quad (3.3)$$

$i, l = 0, 1, \dots, n$

approximiert werden. Diese Integrale treten bereits bei der Berechnung der Ritz-Matrix M auf.

Bemerkung: Definiert man die Matrizen

$$\begin{aligned} \overline{M} &:= \left(- \int_0^\pi u_j''(x) u_k(x) dx \right), & \widehat{M} &:= \left(\int_0^\pi u_j(x) u_k(x) \widehat{q}(x) dx \right), \\ M^l &:= \left(\int_0^\pi u_j(x) u_k(x) q_l(x) dx \right), & j, k &= 1, 2, \dots, r, \quad l = 0, 1, \dots, n, \end{aligned}$$

so ist

$$M := \overline{M} + \widehat{M} + \sum_{l=0}^n a_l M^l \quad (3.4)$$

die Ritz-Matrix des Potentials

$$q(x) = \widehat{q}(x) + \sum_{l=0}^n a_l q_l(x),$$

und die Integrale (3.3) können als Matrix-Vektor-Produkte geschrieben werden:

$$\int_0^\pi \tilde{y}_i^2(x) q_l(x) dx = (z^i)^T M^l z^i. \quad (3.5)$$

Die Matrizen M^l sind von den vorgegebenen Eigenwerten unabhängig. Sie müssen für jede Wahl von Basisfunktionen nur einmal berechnet werden und können danach für andere Rekonstruktionsaufgaben auf dem Rechner bereitgehalten werden.

Durch (3.4) wird ein *additives inverses Eigenwertproblem für Matrizen* definiert. Gesucht ist $a = (a_l) \in \mathbb{R}^{n+1}$, so daß die Matrix M aus (3.4) die vorgegebenen Eigenwerte ν_i , $i = 1, 2, \dots, n$, besitzt (wir weisen allerdings darauf hin, daß beim inversen Eigenwertproblem für Matrizen die Dimension der Matrizen im Regelfall mit der Zahl der vorgegebenen Eigenwerte übereinstimmt, wohingegen beim Ritz-Verfahren typischerweise erheblich mehr Vergleichsfunktionen verwendet werden, als Eigenwerte gesucht sind).

Newtonverfahren für das inverse Eigenwertproblem für Matrizen wurden beispielsweise von Friedland, Nocedal und Overton [14] untersucht. Für die Ableitung des Eigenwerts λ_i der Matrix M aus (3.4) gilt

$$\frac{\partial \lambda_i}{\partial a_l} = (z^i)^T M^l z^i, \quad i, l = 1, 2, \dots, n.$$

Die approximative Lösung des Rekonstruktionsverfahrens 2.1 unter Verwendung des Ritz-Verfahrens entspricht also der Lösung eines inversen Eigenwertproblems für Matrizen mit dem Newtonverfahren. \square

3.2 Näherungslösungen durch Lösen von Anfangswertproblemen

In diesem Abschnitt stellen wir ein zweites Verfahren zur näherungsweise Durchführung des Rekonstruktionsverfahrens 2.1 vor. Wieder behandeln wir zuerst das Problem, Eigenwerte zu berechnen.

Greenberg und Babuška [17] bestimmen die Eigenwerte von wesentlich allgemeineren Eigenwertproblemen mit einem Schießverfahren. Wir wenden dieses Verfahren auf das Dirichletproblem an und berechnen damit Näherungen sowie Ober- und Unterschranken der Eigenwerte des Dirichletproblems (siehe auch [35]).

Wie im Abschnitt 1.1 gezeigt wurde, lassen sich Eigenwertschranken durch Lösen des mit dem Dirichletproblem verbundenen Anfangswertproblems

$$\begin{aligned} -u'' + q(x)u &= \mu u, & x \in [0, \pi] \\ u(0) &= 0, & u'(\pi) = 1 \end{aligned} \tag{3.6}$$

für $u = u(x, \mu)$ berechnen, indem man die Anzahl $N(\mu)$ der Nullstellen von $u(x, \mu)$ im Intervall $(0, \pi]$ zählt.

Hat man zum n -ten Eigenwert λ_n des Dirichletproblems eine untere Schranke $\underline{\lambda}$ und eine obere Schranke $\bar{\lambda}$ gefunden, kann man (3.6) für ein $\mu \in [\underline{\lambda}, \bar{\lambda}]$ lösen und Korollar 1.1.2 auf μ und $\underline{\lambda}$ bzw. μ und $\bar{\lambda}$ anwenden, um eine verbesserte Einschließung für λ_n zu erhalten. Dieser Prozeß wird so lange fortgesetzt, bis die gefundenen Schranken nahe genug bei λ_n liegen.

Gilt $N(\underline{\lambda}) = n$ und $N(\bar{\lambda}) = n + 1$, kann man mit dem Sekantenverfahren die (im Intervall $[\underline{\lambda}, \bar{\lambda}]$ eindeutige) Nullstelle der Funktion

$$g(\mu) := u(\pi, \mu)$$

und damit den i -ten Eigenwert des Dirichletproblems näherungsweise bestimmen.

Ist für den i -ten Eigenwert eine Näherung $\tilde{\lambda}_i$ bekannt, erhält man durch Lösen des Anfangswertproblems (3.6) mit $\mu = \tilde{\lambda}_i$ eine Näherung der i -ten Eigenfunktion. Nach geeigneter Normierung lassen sich die Komponenten der Funktionalmatrix beim Rekonstruktionsverfahren 2.1 durch numerische Integration bestimmen, und somit das Rekonstruktionsverfahren durchführen.

Diese Implementierung vermeidet den *methodischen Fehler*, der beim Ritz-Verfahren dadurch entsteht, daß die Eigenwerte der Ritz-Matrix nicht mit den Eigenwerten des Dirichletproblems übereinstimmen.

3.3 Lösungseinschließung auf dem Computer

In den Anwendungen ist das Anfangswertproblem (3.6) i.a. nicht explizit lösbar. Auf einem Computer berechnete Näherungslösungen $\tilde{u}(x, \mu)$ sind aber bei der Verwendung einer Gleitpunktarithmetik mit Rundungsfehlern behaftet, die das Ergebnis verfälschen können. Es kann i.a. beispielsweise nicht gewährleistet werden, daß $\tilde{u}(x, \mu)$ im Intervall $(0, \pi]$ gleich viele Nullstellen besitzt wie die exakte Lösung $u(x, \mu)$ von (3.6).

Dieses Problem hat seine Ursache darin, daß auf einem Computer nur endlich viele Zahlen darstellbar sind. Die Menge der reellen Zahlen muß durch Rundung auf eine Teilmenge, die sogenannten *Gleitpunktzahlen* oder *Maschinenzahlen*, abgebildet werden. Ebenso sind die arithmetischen Grundoperationen $+$, $-$, \cdot , und $/$ auf dem Rechner durch Gleitpunktoperationen zu

ersetzen, da z.B. die Multiplikation von Gleitpunktzahlen nicht notwendigerweise eine Gleitpunktzahl mit gleicher Mantissenlänge liefert. Zum kontrollierten Rechnen auf dem Computer ist daher eine mathematisch exakte Definition der Rechnerarithmetik notwendig, wie sie von Kulisch [24] sowie Kulisch und Miranker [25] gegeben wurde.

Um auf dem Computer garantierte Lösungen zu berechnen, kann man die Intervallrechnung mittels *Maschinenintervallen* auf den Rechner übertragen. Statt gerundeten Operationen mit Gleitpunktzahlen sind dann Verknüpfungen zwischen Maschinenintervallen auszuführen (siehe ebenfalls [24], [25] für eine exakte Behandlung der Maschinenintervallarithmetik). Das wesentliche Element besteht bei einer derartigen Realisierung darin, daß die auftretenden Rundungsfehler stets in das Ergebnisintervall mit eingeschlossen werden, und daß eine Fehlermeldung ausgegeben wird, wenn keine Einschließung berechnet werden kann.

3.3.1 PASCAL-XSC

Es gibt heute eine ganze Reihe von Programmiersprachen, die eine Intervallarithmetik zur Verfügung stellen, und die Intervalle, reelle und intervallwertige Vektoren und Matrizen als Datentypen zulassen und unterstützen. Wir haben in unseren Anwendungen das am Institut für Angewandte Mathematik der Universität Karlsruhe entwickelte PASCAL-XSC [26] (eine Erweiterung von Standard-PASCAL für wissenschaftliches Rechnen) verwendet.

Durch einen portablen Compiler nach ANSI-C ist PASCAL-XSC für eine Vielzahl von Rechnern verfügbar. PASCAL-XSC arbeitet mit der binären Gleitpunktarithmetik des IEEE 754 - Standards in Software-Emulation mit Hardware-Unterstützung und garantiert somit identische Resultate unabhängig vom eingesetzten Rechner.

PASCAL-XSC unterstützt die Implementierung mathematischer Algorithmen durch modulare Programmierung, dynamische Felder, benutzerdefinierte Datentypen und Operatoren, Operatoren und Funktionen mit allgemeinem Ergebnistyp sowie durch das mögliche Überladen von Funktionen, Prozeduren und Operatoren. Mathematische Formeln lassen sich dadurch meist unverändert in den Programmtext übernehmen.

Dies gilt sowohl für skalare als auch für vektorielle oder matrixwertige Datentypen, deren Einsatz durch vordefinierte Arithmetikmodule unterstützt

wird. Außerdem ist ein genaues Skalarprodukt verfügbar, mit dem sich Produkte von Vektoren mit nur einer Rundung berechnen lassen.

Die implementierte Intervallarithmetik umfaßt reelle und komplexe Intervalle, dynamische Intervallvektoren und -matrizen sowie die Grundoperatoren zur Addition und Multiplikation. Für die mathematischen Standardfunktionen $\sin x$, e^x , usw., werden darüber hinaus Intervallauswertungen bereitgestellt, die Einschließungen des Funktionswerts liefern. Rundungsfehler werden bei Operationen mit Intervallen automatisch mit in das Ergebnis eingeschlossen, man erhält stets garantierte Lösungsschranken.

Für eine Vielzahl numerischer Standardprobleme (lineare Gleichungssysteme, Polynomauswertung, Eigenwerte symmetrischer Matrizen, usw.) stehen Module mit Prozeduren zur Verfügung, die nicht nur Näherungslösungen, sondern Einschließungsintervalle der wahren Lösung berechnen und gleichzeitig die Existenz und die Eindeutigkeit der Lösung im angegebenen Intervall nachweisen.

3.3.2 Garantierte Lösung von Anfangswertproblemen

Von Lohner [29] wurde ein Einschließungsverfahren zur verifizierten Lösung gewöhnlicher Anfangswertprobleme

$$\begin{aligned}y' &= f(x, y) \\ y(x^0) &= y^0\end{aligned}\tag{3.7}$$

(mit $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$) entwickelt und in PASCAL-XSC als Numerikpaket AWA implementiert. Es beruht auf einem Einschnittverfahren mit Entwicklung der Lösung nach lokalen Fehlern und ist für eine große Klasse von Differentialgleichungen anwendbar. Vorausgesetzt wird im wesentlichen, daß f als Komposition aus mathematischen Standardfunktionen und -operationen dargestellt werden kann und hinreichend oft differenzierbar ist. Für die bei der Lösung des inversen Dirichletproblems auftretenden Anfangswertprobleme sind diese Voraussetzungen erfüllt, falls die gewählten Basisfunktionen hinreichend oft differenzierbar sind.

Bemerkung: Bei einer vereinfachten Version des Verfahrens von Lohner erhält man folgendermaßen eine kontinuierliche Einschließung der Lösung $y(x)$ des Anfangswertproblems in stückweise zusammengesetzte Intervallpolynome: Das Intervall, über dem die Lösung eingeschlossen werden soll,

wird durch Stützstellen in Teilintervalle unterteilt. In jedem Teilintervall $[x_l, x_{l+1}]$, $l = 0, 1, \dots, l_{max} - 1$, werden Intervalle $[c_k]$, $k = 0, 1, \dots, r$ berechnet, die die entsprechenden Taylorkoeffizienten von $y(x)$ bis zur Ordnung $r - 1$ und das zugehörige Restglied einschließen. Es gilt dann

$$y(x) \in [y](x) = \sum_{k=0}^{r-1} [c_k](x - x_l)^k + [c_r] \left(\frac{x - x_l}{x_{l+1} - x_l} \right)^r, \quad x \in [x_l, x_{l+1}], \quad (3.8)$$

wobei $[y](x)$ hier *nicht* als Polynom mit Intervallkoeffizienten zu verstehen ist, sondern als Funktionenschlauch

$$\begin{aligned} [y](x) &= [\underline{y}(x), \bar{y}(x)] \\ &= \{ y(x) \in C^0[x_l, x_{l+1}] : \underline{y}(x) \leq y(x) \leq \bar{y}(x) \text{ auf } [x_l, x_{l+1}] \}. \end{aligned}$$

□

3.4 Implementierung des Rekonstruktionsverfahrens

3.4.1 Eigenwertschranken auf dem Computer

Um auf dem Computer garantierte Eigenwertschranken zu erhalten, muß das Anfangswertproblem (3.6) mit einem Intervallverfahren gelöst werden, das unter Einschluß sämtlicher Fehler einen Funktionenschlauch

$$\begin{aligned} [u](x, \mu) &:= [\underline{u}(x, \mu), \bar{u}(x, \mu)] \\ &:= \{ u(x) \in C^0[0, \pi] : \underline{u}(x, \mu) \leq u(x) \leq \bar{u}(x, \mu) \text{ in } [0, \pi] \} \end{aligned}$$

mit $\underline{u}(x, \mu), \bar{u}(x, \mu) \in C^0[0, \pi]$ berechnet, der die exakte Lösung $u(x, \mu)$ enthält.

Gilt nun für $0 < \alpha < \beta < \pi$

$$\underline{u}(\alpha, \mu) \geq 0, \quad \bar{u}(\beta, \mu) \leq 0 \quad (3.9)$$

(bzw. $\bar{u}(\alpha, \mu) \leq 0$, $\underline{u}(\beta, \mu) \geq 0$), so besitzt die Funktion $u(x, \mu) \in [u](x, \mu)$ sicher eine Nullstelle im Intervall $[\alpha, \beta]$. Beim Zählen der Nullstellen von $u(x, \mu)$ müssen wir aber überprüfen, ob $u(x, \mu)$ in einem solchen Intervall

genau eine Nullstelle besitzt. Dies erreichen wir dadurch, daß wir zusätzlich eine äquivalente Einschließung $[u'](x, \mu)$ von $u'(x, \mu)$ berechnen und $([u], [u'])$ auffassen als Funktionenmenge

$$([u], [u'])(x, \mu) := \{u(x) \in C^1[0, \pi] : u(x) \in [u](x, \mu), u'(x) \in [u'](x, \mu)\}.$$

Gilt dann neben (3.9) für alle $x \in [\alpha, \beta]$

$$0 \in [u](x, \mu) \Rightarrow 0 \notin [u'](x, \mu), \quad (3.10)$$

so besitzt jedes $u(x) \in ([u], [u'])$ genau eine einfache Nullstelle in $[\alpha, \beta]$. Wir sagen in diesem Fall, daß $([u], [u'])$ eine einfache Nullstelle besitzt. Da Lösungen $u(x, \mu)$ von (3.6) im Intervall $[0, \pi]$ nur einfache Nullstellen besitzen, kann (3.10) erzielt werden, falls $[u]$ und $[u']$ auf dem Computer genau genug berechnet werden.

Falls $([u], [u'])$ i Nullstellen im Intervall $(0, \pi)$ besitzt, ist μ nach Satz 1.1.1 eine Oberschranke des $(i - 1)$ -ten Eigenwerts des Dirichletproblems. μ ist eine Unterschranke von λ_i , falls $0 \notin [u](\pi, \mu)$. Ist $0 \in [u](\pi, \mu)$, kann über das Vorzeichen von $u(x, \mu)$ nicht entschieden werden. $N(u(x, \mu))$ könnte sowohl den Wert i als auch $i+1$ annehmen, und daher kann nicht festgestellt werden, ob μ eine Unterschranke von λ_i ist.

Dieser Fall tritt in der Praxis natürlich gerade für $\mu = \lambda_i$ bzw. $\mu \approx \lambda_i$ auf, da dann $u(\pi, \mu) = 0$ bzw. $u(\pi, \mu) \approx 0$ gilt. Die beim Rechnen auf dem Computer unvermeidbaren Fehler (die ja in $[u](\pi, \mu)$ eingeschlossen werden) führen in diesem Fall zwangsläufig zu einem Intervall, das die Null enthält. Die maximal erzielbare Genauigkeit der Eigenwertschranken hängt also davon ab, wie genau sich die Lösungen der Anfangswertprobleme (3.6) einschließen lassen.

Bemerkung: Wenn man $q(x)$ in (3.6) durch eine Intervallfunktion $[q](x)$ ersetzt, und das entstehende Anfangswertproblem formal mit dem Verfahren von Lohner behandelt, werden wegen der Inklusionsmonotonie der Intervalloperationen simultan die Eigenwerte aller Funktionen $q(x) \in [q](x)$ eingeschlossen. Beim praktischen Rechnen hat sich aber gezeigt, daß sich auf diese Weise i.a. keine engen Eigenwerteinschließungen berechnen lassen, wenn der *Durchmesser*

$$d([q]) := \max_{x \in [0, \pi]} (\bar{q}(x) - \underline{q}(x))$$

von $[q](x)$ groß ist. Um zu simultanen Eigenwerteinschließungen zu gelangen, haben wir daher vom Monotoniesatz 1.1.4 Gebrauch gemacht.

Mit zwei Eigenwertschranken $\underline{\lambda}_i(q_{mid})$, $\bar{\lambda}_i(q_{mid})$ der *Mittelpunktsfunktion*

$$q_{mid}(x) := \frac{1}{2}(\underline{q}(x) + \bar{q}(x))$$

von $[q](x)$ folgt aus dem Monotoniesatz 1.1.4 für die Eigenwerte jeder Funktion $q(x) \in [q](x)$ wegen

$$\|q(x) - q_{mid}(x)\|_\infty \leq \frac{1}{2} d([q])$$

die Abschätzung

$$\underline{\lambda}_i(q_{mid}) - \frac{1}{2} d([q]) \leq \lambda_i(q) \leq \bar{\lambda}_i(q_{mid}) + \frac{1}{2} d([q]). \quad (3.11)$$

Wir haben in den gerechneten Beispielen die Auswahlfunktion m im Intervall-Newtonschritt des Rekonstruktionsverfahrens 2.2 so definiert, daß sie den Mittelpunkt des Intervallvektors $[a]^{(k)}$ auswählt. In diesem Fall sind die Eigenwertschranken von $q_{mid}^{(k)}(x)$ zur Durchführung der Iteration ohnehin zu berechnen. Zur Einschließung der Eigenwerte von $[q]^{(k)}(x)$ mit Hilfe von (3.11) ist also nur zusätzlich der Durchmesser von $[q]^{(k)}(x)$ zu bestimmen bzw. abzuschätzen. Bei den gerechneten Beispielen hat sich gezeigt, daß durch diese Methode mit geringem Aufwand enge Einschließungen der Eigenwerte aller in $[q]^{(k)}(x)$ enthaltenen Potentiale berechenbar sind. Diese Eigenwerteinschließungen werden bei der Einschließung der Eigenfunktionen $g_i(x)$ benötigt. \square

3.4.2 Einschließung von Eigenfunktionen

Wenn wir bei der formalen Anwendung des Verfahrens von Lohner auf das Anfangswertproblem (3.6) anstelle des reellen Parameters μ ein Intervall $[\mu]$ zulassen, wird auf Grund der Inklusionsmonotonie der Intervalloperationen ein Funktionenschlauch $u(x, [\mu])$ berechnet, so daß

$$u(x, \mu) \in u(x, [\mu]) \quad \text{für alle } x \in [0, \pi] \text{ und alle } \mu \in [\mu].$$

Diese Eigenschaft haben wir benutzt, um in zwei Schritten Schranken von Eigenfunktionen von $q(x)$ zu erhalten.

Zuerst bestimmen wir eine Einschließung $[\lambda_i]$ des i -ten Eigenwerts λ_i von $q(x)$. Danach wenden wir das Verfahren von Lohner formal auf das Anfangswertproblem (3.6) für $[\mu] = [\lambda_i]$ an. Wegen

$$u_i(x) = u(x, \lambda_i) \in u(x, [\lambda_i]) \quad \text{für alle } x \in [0, \pi]$$

ist der Funktionenschlauch $u(x, [\lambda])$ eine Einschließung der auf $u'(0) = 1$ normierten Eigenfunktion $u_i(x)$.

Wenn wir bei der Behandlung der Anfangswertprobleme (3.6) mit dem Verfahren von Lohner sowohl für $q(x)$ als auch für μ Intervallgrößen einsetzen, erhalten wir einen Funktionenschlauch $[u] := u(x; [q], [\mu])$, so daß

$$u(x; q, \mu) \in u(x; [q], [\mu])$$

für alle $x \in [0, \pi]$, alle $q(x) \in [q](x)$ und alle $\mu \in [\mu]$ gilt. Um simultan die normierten Eigenfunktionen aller $q(x) \in [q](x)$ einzuschließen, haben wir zuerst mit der Methode aus Kapitel 3.4.1 ein Intervall $[\lambda_i]$ bestimmt, das die i -ten Eigenwerte aller $q(x) \in [q](x)$ enthält, und anschließend formal $u(x; [q], [\lambda_i])$ in der Darstellung (3.8) berechnet.

Durch formales Quadrieren von (3.8) erhält man in jedem Teilintervall $[x_l, x_{l+1}] \subset [0, \pi]$ eine Einschließung für $u_i^2(x)$ in der Form

$$u_i^2(x) \in \sum_{k=0}^{2r} [d_k](x - x_l)^k, \quad x \in [x_l, x_{l+1}].$$

Da $(x - x_l)^k$ in $[x_l, x_{l+1}]$ das Vorzeichen nicht wechselt, dürfen die Intervallkoeffizienten $[d_k]$ bei der Integration vor das Integral gezogen werden (siehe Klein [23], Seite 29, Bemerkung 4). Summation der Integrale

$$\int_{x_l}^{x_{l+1}} u_i^2(x) dx \in \sum_{k=0}^{2r} [d_k] \int_{x_l}^{x_{l+1}} (x - x_l)^k dx$$

über alle Teilintervalle von $[0, \pi]$ liefert ein Intervall $[w]$, in dem $\|u_i\|_2^2$ enthalten ist. Durch den Divisionsschritt

$$[g_i](x) := \frac{[u_i](x)}{\sqrt{[w]}}$$

erhalten wir Einschließungen der L^2 -normierten Eigenfunktionen $g_i(x)$ aller $q(x) \in [q](x)$ in der Darstellung (3.8). Diese haben wir benutzt, um die Funktionalmatrix im Algorithmus 2.2 in eine Intervallmatrix einzuschließen.

3.4.3 Einschließung der partiellen Ableitungen

Die Elemente der Funktionalmatrix im Intervall-Newtonschritt haben wir durch formale Auswertung der Integrale

$$\int_0^\pi [g_i]^2(x) q_j(x) dx \quad (3.12)$$

wie im vorigen Abschnitt eingeschlossen. Wir haben für die Basisfunktionen $q_j(x)$, $j = 0, 1, \dots, n$, Einschließungen der Gestalt (3.8) mit den gleichen Stützstellen wie bei $[g_i](x)$ bestimmt. Den Ausdruck (3.12) haben wir dann durch formale Multiplikation dieser Funktionenschläuche und anschließende formale Integration wie oben ausgewertet.

3.4.4 Das Maschinen-Intervall-Newtonverfahren

Die Durchführung des k -ten Intervall-Newtonschritts

$$\text{IN}([a]^{(k)}) := m^{(k)} - \text{IGA}\left(\left(\frac{\partial f_i}{\partial a_j}([a]^{(k)})\right), f(m^{(k)})\right) \quad (3.13)$$

auf dem Rechner wird u.a. durch die folgenden Modifikationen an die darstellbaren Maschinenzahlen angepaßt:

1. Die Auswahlfunktion m wird so definiert, daß sie zu einem Maschinenintervall $[x]$ eine Maschinenzahl $x \in [x]$ liefert.
2. Falls die Komponenten von $f(m^{(k)})$ nicht selbst Maschinenzahlen sind, werden sie in Maschinenintervalle eingeschlossen.
3. Funktionswerte von f sind eventuell nicht durch endlich viele rundungsfehlerfreie Maschinenoperationen berechenbar. Statt f wird in diesem Fall auf dem Rechner eine Funktion \tilde{f} ausgewertet. Um den Existenznachweis einer Nullstelle von f mit Hilfe des Intervall-Newtonverfahrens auf dem Rechner zu führen, werden bei der Auswertung von \tilde{f} Maschinenintervalle berechnet, die die Funktionswerte von f einschließen.

Wir setzen im folgenden voraus, daß bei der Auswertung von $\tilde{f}(m^{(k)})$ auf dem Rechner ein Maschinenintervall berechnet wird, das $f(m^{(k)})$ enthält. Ebenso setzen wir voraus, daß auf dem Rechner eine Funktion \tilde{f}' ausgewertet wird, so daß $\tilde{f}'([x]) \supseteq f'([x])$ für jedes Maschinenintervall $[x]$ gilt.

Statt (3.13) sind auf dem Rechner also vielmehr die Lösungen $a^{(k+1)}$ der Gleichungssysteme

$$A^{(k)} (a^{(k+1)} - m^{(k)}) = -b^{(k)}$$

für alle

$$A^{(k)} \in \left(\frac{\partial \tilde{f}_i}{\partial a_j}([a]^{(k)}) \right), \quad b^{(k)} \in \tilde{f}(m^{(k)})$$

in einen Intervallvektor $[a]^{(k+1)}$ einzuschließen. Beim praktischen Rechnen haben wir dazu nicht den Intervall-Gauß-Algorithmus eingesetzt, sondern das in der PASCAL-XSC-Numerikbibliothek enthaltene Modul LSS, das diese Aufgabe ebenfalls erfüllt. Damit wir diese Prozedur anstelle von IGA in unserem Rekonstruktionsverfahren verwenden können, müssen wir lediglich den Einschließungssatz 2.11.2 etwas modifizieren.

Im folgenden bezeichnen wir für eine Intervallmatrix $[A]$ und einen Intervallvektor $[b]$ mit

$$\text{LSS}([A], [b])$$

einen Algorithmus, der einen Intervallvektor $[x]$ berechnet, so daß $[x]$ alle Lösungen x der reellen Gleichungssysteme

$$Ax = b, \quad A \in [A], \quad b \in [b] \quad (3.14)$$

einschließt. Falls eine solche Einschließung existiert und mit dem gegebenen Algorithmus berechnet werden kann, heißt $\text{LSS}([A], [b])$ *durchführbar*, sonst *nicht durchführbar*.

Ist nun also die Auswahlfunktion

$$m : [x] \mapsto \mathbb{R}^n, \quad m([x]) \in [x]$$

so definiert, daß $m([x])$ für alle Maschinenintervalle $[x] \subseteq [x]^{(0)}$ eine Maschinenzahl liefert, so ist das *Maschinen-Intervall-Newtonverfahren* (MIN) zur Einschließung einer Nullstelle x^* der differenzierbaren Funktion

$$f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$$

im Maschinenintervall $[x]^{(0)} \subset D$ auf dem Rechner durch die folgende Iteration gegeben (wobei \tilde{f}, \tilde{f}' die oben beschriebenen Eigenschaften besitzen sollen):

Für $k = 0, 1, \dots$:

$$(MIN) \begin{cases} m^{(k)} := m([x]^{(k)}) \\ MIN([x]^{(k)}) := m^{(k)} - LSS(\tilde{f}'([x]^{(k)}), \tilde{f}(m^{(k)})) \\ [x]^{(k+1)} := MIN([x]^{(k)}) \cap [x]^{(k)}. \end{cases}$$

Aus dem Beweis von Satz 2.11.2 ist ersichtlich, daß die Existenzaussage des Satzes auch für LSS anstelle von IGA gilt, da zum Nachweis einer Lösung lediglich die Selbstabbildung von $[x]^{(0)}$ durch die Funktion p aus (2.29) zu zeigen ist. Wir können also die folgenden Korollare formulieren:

Korollar 3.4.1 *Es sei $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ differenzierbar in D . Für $m^{(0)} \in [x]^{(0)} \subset D$ existiere eine Maschinenintervallauswertung \tilde{f} von f mit $f(m^{(0)}) \in \tilde{f}(m^{(0)})$ sowie eine Maschinenintervallauswertung \tilde{f}' der Ableitung von f mit $f'([x]^{(0)}) \subseteq \tilde{f}'([x]^{(0)})$. Dann gilt:*

Falls $LSS(\tilde{f}'([x]^{(0)}), \tilde{f}(m^{(0)}))$ durchführbar ist und

$$MIN([x]^{(0)}) \subseteq [x]^{(0)}$$

gilt, besitzt f in $MIN([x]^{(0)})$ eine in $[x]^{(0)}$ eindeutige Nullstelle.

Korollar 3.4.2 *Gilt für ein $k \in \mathbb{N}_0$ in der Maschinen-Intervall-Newton-iteration*

$$MIN([x]^{(k)}) \subseteq [x]^{(k)},$$

so liegt in $[x]^{(k+1)}$ eine (in $[x]^{(0)}$ eindeutige) Lösung von $f(x) = 0$.

Bevor wir das Maschinen-Intervall-Newtonverfahren zur Lösung des inversen Dirichletproblems auf dem Rechner angeben, machen wir auf ein letztes Problem aufmerksam. In der Regel sind auch die Basisfunktionen auf dem Rechner nicht exakt auswertbar. In diesem Fall sind die Funktionswerte der Basisfunktionen in allen Rechnungen durch garantierte Schranken einzuschließen.

Damit erhalten wir das *Maschinen-Intervall-Newtonverfahren für das inverse Dirichletproblem*:

Algorithmus 3.1: Rekonstruktion für das inverse Dirichletproblem
Maschinen-Intervall-Newtonverfahren im $I\mathbb{R}^{n+1}$

1. Wähle symmetrische Basisfunktionen $\widehat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$.
2. Wähle $[a]^{(0)} \in I\mathbb{R}^{n+1}$.
3. Für $k = 0, 1, \dots$:

(i) Für $i = 0, 1, \dots, n$:

(a) Berechne $\tilde{f}_i(m^{(k)}) \ni \lambda_i(q^{(k)}(x)) - \nu_i$,

(b) Berechne $g_i(x; [a]^{(k)}) = g_i(x; [q]^{(k)}(x))$,

wobei

$$m^{(k)} := m([a]^{(k)}),$$

$$q^{(k)}(x) := q(x; m^{(k)}) := \widehat{q}(x) + \sum_{j=0}^n m_j^{(k)} q_j(x),$$

$$[q]^{(k)}(x) := q(x; [a]^{(k)}) := \widehat{q}(x) + \sum_{j=0}^n [a_j]^{(k)} q_j(x).$$

(ii) Berechne die Funktionalmatrix

$$\left(\frac{\partial \tilde{f}_i}{\partial a_j}([a]^{(k)}) \right) \ni \left(\int_0^\pi g_i^2(x; [a]^{(k)}) q_j(x) dx \right),$$

$i, j = 0, 1, \dots, n.$

(iii) Führe den Maschinen-Intervall-Newtonschritt

$$\text{MIN}([a]^{(k)}) := m^{(k)} - \text{LSS}\left(\frac{\partial \tilde{f}_i}{\partial a_j}([a]^{(k)}), \tilde{f}(m^{(k)})\right),$$

$$[a]^{(k+1)} := \text{MIN}([a]^{(k)}) \cap [a]^{(k)}$$

durch.

3.4.5 Der vollständige Algorithmus

Da die Rechnung mit Intervallen auf dem Computer aufwendig ist, will man bei dem Problem, mit dem Intervall-Newtonverfahren eine Nullstelle x^* einer gegebenen Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ einzuschließen, möglichst nur einen Iterationsschritt durchführen. In der Regel berechnet man daher zuerst mit dem gewöhnlichen Newtonverfahren eine Näherungslösung, bis ein Abbruchkriterium erfüllt ist. Hierbei wird die Rechnung in Gleitpunktarithmetik ohne Berücksichtigung von Rundungsfehlern durchgeführt.

Von grundlegendem Interesse ist nun die folgende Aufgabe: aus der Kenntnis von $r + 1$ Iterierten $x^{(k)}, k = 0, 1, \dots, r$, des reellen Newtonverfahrens ist ein Intervall $[x]$ gesucht, in dem eine Nullstelle x^* von f vermutet wird. Zum Existenznachweis der Nullstelle ist anschließend eine geeignete Intervalliteration mit Startintervall $[x]^{(0)} = [x]$ durchzuführen. Neben dem Intervall-Newtonverfahren steht dazu u.a. auch die Iteration mit dem *Krawczyk-Operator* (siehe [6]) zur Verfügung.

In [6] geben Alefeld, Gienger und Potra ein neues Abbruchkriterium für das Newtonverfahren an, bei dem Eigenschaften des Krawczyk-Operators mit einem Korollar des Satzes von Kantorowitsch kombiniert werden. Wenn das Abbruchkriterium erfüllt ist, berechnen sie ausgehend von den letzten drei Iterierten des Newtonverfahrens eine Konstante $\varepsilon > 0$, so daß

$$[x] := x^{(r)} + \varepsilon [-1, 1] \quad (3.15)$$

(Addition in jeder Komponente von $x^{(r)}$) ein geeignetes Startintervall für die Iteration mit dem Krawczyk-Operator ist. Wir haben dieses Intervall als Ausgangsnäherung für das Maschinen-Intervall-Newtonverfahren verwendet.

Das in [6] konstruierte Intervall $[x]$ besitzt die folgende Eigenschaft: falls $[x]$ eine Nullstelle von f enthält, liegt der relative Fehler jeder Näherung $\tilde{x} \in [x]$ in der Größenordnung der Maschinenkonstante des verwendeten Rechners. Die Einschließung einer Lösung mit dieser Genauigkeit ist aber beim Maschinen-Intervall-Newtonverfahren nur dann möglich, wenn sich der Funktionswert von f an der Stelle $m^{(0)}$ auf dem Rechner ebenso genau einschließen läßt. Falls sich das Intervall $\tilde{f}(m^{(0)})$ – wie bei der von uns im Rekonstruktionsverfahren 3.1 verwendeten Funktion – über mehrere Maschinezahlen erstreckt, muß mindestens die Beziehung

$$m^{(0)} - \text{LSS}(\tilde{f}'(m^{(0)}), \tilde{f}(m^{(0)})) \subseteq [x]^{(0)}$$

erfüllt sein, damit die Inklusion $\text{MIN}([x]^{(0)}) \subseteq [x]^{(0)}$ möglich ist.

Diese „innere Einschließung“ kann man mit Hilfe der im reellen Newtonverfahren berechneten Näherung zu $f'(x^{(r-1)})$ approximieren, indem man überprüft, ob für das nach den Kriterien aus [6] ausgewählte Intervall $[x]^{(0)}$ die Beziehung

$$m^{(0)} - \text{LSS}(f'(x^{(r-1)}), \tilde{f}(m^{(0)})) \subseteq [x]^{(0)} \quad (3.16)$$

erfüllt ist. Da $\tilde{f}(m^{(0)})$ zur Durchführung des Maschinen-Intervall-Newtonverfahrens sowieso (unabhängig von ε) berechnet werden muß, besteht der zusätzliche Aufwand aus der Einschließung der Lösungen des Intervall-Gleichungssystems in (3.16). Wir haben diesen Test benutzt, um eine zu kleine Wahl von ε zu verhindern, und in diesem Fall von einer ebenfalls in [6] angegebenen modifizierten Berechnung von ε Gebrauch gemacht. Dadurch gelang bei den gerechneten Beispielen jeweils im ersten Schritt des Maschinen-Intervall-Newtonverfahrens der Existenznachweis einer Lösung des inversen Dirichletproblems.

Die Implementierung des vollständigen Rekonstruktionsverfahrens zur Lösung des inversen Dirichletproblems bei vorgegebenen, paarweise verschiedenen Eigenwerten ν_i , $i = 0, 1, \dots, n$, beschreiben wir nun im Algorithmus 3.2. Dieser Algorithmus zerfällt in drei Teiliterationen.

Zuerst berechnen wir unter Verwendung des Ritz-Verfahrens eine erste Näherungslösung \tilde{a} . In vielen gerechneten Beispielen haben sich mit dem Ritz-Verfahren mit geringem Aufwand gute Näherungen berechnen lassen. Mit dem aufwendigeren, aber unserer Erfahrung nach genaueren Näherungsverfahren aus Abschnitt 3.2 verbessern wir \tilde{a} , bis ein Abbruchkriterium erfüllt ist. Alle Rechnungen werden bis zu diesem Zeitpunkt in Gleitpunktarithmetik ohne Berücksichtigung von Rundungsfehlern durchgeführt.

Zu der so erhaltenen Näherung \tilde{a} haben wir mit Hilfe des Testintervalls aus (3.16) und den Verfahren aus [6] eine Konstante ε berechnet, mit der bei den gerechneten Beispielen ausgehend von

$$[a]^{(0)} := \tilde{a} + \varepsilon[-1, 1]$$

jeweils im ersten Schritt des Maschinen-Intervall-Newtonverfahrens der Existenznachweis einer Lösung des inversen Dirichletproblems gelang.

Dazu war zu überprüfen, ob die Inklusion

$$\text{MIN}([a]^{(0)}) \subseteq [a]^{(0)}$$

Algorithmus 3.2: Rekonstruktion für das inverse Dirichletproblem
Implementierung auf dem Computer

1. Wähle symmetrische Basisfunktionen $\hat{q}(x)$, $q_j(x)$, $j = 0, 1, \dots, n$.
2. Wähle einen Startvektor $a^{(0)}$ mit zugehörigem Startpotential $q^{(0)}(x) := q(x; a^{(0)}) = \hat{q}(x) + \sum_{j=0}^n a_j q_j(x)$.
3. Berechne mit Hilfe des Ritz-Verfahrens mit Sinus-Vergleichsfunktionen und Gleitpunktarithmetik einen Näherungsvektor \check{a} mit Näherungspotential $\check{q}(x) := q(x; \check{a})$.
4. Starte das reelle Newtonverfahren (2.1) mit \check{a} und berechne in Gleitpunktarithmetik eine verbesserte Näherung \tilde{a} mit Näherungspotential $\tilde{q}(x) := q(x; \tilde{a})$.
5. Erzeuge eine Intervallfunktion $[q]^{(0)}(x)$, in der eine Lösung des inversen Dirichletproblems nachgewiesen werden soll. Setze dazu speziell $\tilde{q}(x) := \check{q}(x)$ und $[a]_i^{(0)} := [-\varepsilon, \varepsilon]$ für $i = 0, 1, \dots, n$. Wähle ε nach den Kriterien aus [6], unter Berücksichtigung von (3.16).
6. Führe das Maschinen-Intervall-Newtonverfahren (3.1) mit Startwert $[a]^{(0)}$ durch, bis entweder Inklusion oder Abbruch eintritt oder sich die Iterierten wegen $[a]^{(k)} \subseteq \text{MIN}([a]^{(k)})$ nicht mehr ändern.

erfüllt war. In diesem Fall schließt $[q]^{(1)}(x) = q(x; [a]^{(1)})$ genau eine reellwertige Funktion der Gestalt $q(x; a)$, nämlich $q^*(x) = q(x; a^*)$ ein, die Lösung des inversen Dirichletproblems ist.

Im letzten Kapitel dieser Arbeit demonstrieren wir nun die Anwendbarkeit des Rekonstruktionsverfahrens 3.2 an Beispielen.

Kapitel 4

Numerische Beispiele

Die numerischen Berechnungen der nachfolgenden Beispiele für das direkte und das inverse Dirichletproblem mit symmetrischen Potentialen wurden in PASCAL-XSC durchgeführt. Dabei stehen 16 dezimale Mantissenstellen zur Verfügung. Bei der intervallmäßigen Rechnung wurden sämtliche Rundungsfehler mit in das Ergebnis eingeschlossen.

Die Einschließungen der Lösungen der Anfangswertprobleme

$$\begin{aligned} -u'' + q(x)u &= \mu u, & x \in [0, \pi] \\ u(0) &= 0, & u'(0) = 1 \end{aligned} \tag{4.1}$$

wurden mit dem Verfahren von Lohner [29] berechnet. Wir haben stets Taylorkoeffizienten bis zur Ordnung 15 verwendet. Die jeweils größten und kleinsten benutzten Schrittweiten sind bei den gerechneten Beispielen angegeben. Eine größere Zahl von Taylorkoeffizienten oder kleinere Schrittweiten führten i.a. nicht zu signifikant besseren Einschließungen.

Zur Lösung der linearen Intervall-Gleichungssysteme im Newtonschritt haben wir die PASCAL-XSC-Prozedur LSS eingesetzt. Die Berechnung der Eigenwerte der Ritz-Matrizen haben wir mit der PASCAL-XSC-Prozedur SEIG_APP durchgeführt, die auf ein von Lohner [30] entwickeltes Verfahren zur Eigenwerteinschließung bei symmetrischen Matrizen zurückgeht. Die Eigenwertnäherungen werden dabei mit dem klassischen Jacobi-Verfahren berechnet, siehe [30]. SEIG_APP liefert auch Näherungen der Eigenvektoren, mit denen wir beim Rekonstruktionsverfahren 2.1 die Komponenten der Funktionalmatrix näherungsweise bestimmten.

4.1 Beispiele zum direkten Dirichletproblem

Mit den 60 Vergleichsfunktionen $\sin jx$, $j = 1, 2, \dots, 60$ haben wir mit dem Ritz-Verfahren Näherungen für die Eigenwerte berechnet. Eine größere Zahl an Vergleichsfunktionen änderte die Ergebnisse i.a. nicht, weniger Vergleichsfunktionen führten aber bei einigen Beispielen zu schlechteren Eigenwertapproximationen.

Wir haben die Symmetrie der Potentiale ausgenutzt, um getrennt Näherungen für die Eigenwerte von $q(x)$ mit geradem und mit ungeradem Index zu bestimmen, und durch diese Aufspaltung die Dimension der zu lösenden Matrizen-Eigenwertprobleme halbiert.

Eigenwerteinschließungen stellen wir zum Vergleich gegenüber. Die angegebenen Eigenwerteinschließungen sind die bestmöglichen, die sich mit dem Verfahren von Lohner mit 15 Taylorkoeffizienten und den jeweils verwendeten Schrittweiten berechnen ließen. Für Werte von μ , die zwischen den angegebenen Eigenwertschranken liegen, galt für die Lösung $[u](x, \mu)$ des intervallmäßig gelösten Anfangswertproblems (4.1) stets $0 \in [u](\pi, \mu)$, und daher konnte für diese μ nicht entschieden werden, ob sie untere oder obere Schranken der gesuchten Eigenwerte darstellten.

Die Beispiele zeigen, daß sich das Ritz-Verfahren zur näherungsweise Berechnung der Eigenwerte des Dirichletproblems eignet, und auch das Einschließungsverfahren enge Eigenwertschranken liefert. Dies gilt auch für Potentiale mit großer Amplitude und hoher Frequenz.

Beispiel 1: Eigenwerte von $q(x) = \left(\frac{2x}{\pi} - 1\right)^2$

Schrittweiten: $h = \frac{\pi}{30}$ für λ_0 , $h = \frac{\pi}{90}$ für λ_{10} .

i	Ritz-Verfahren	$[\lambda_i]$
0	1.127 841 267 830 505	1.127 841 267 830 $2\frac{80}{60}$
1	4.279 936 021 990 841	4.279 936 021 989 $9\frac{81}{22}$
2	9.311 329 657 043 055	9.311 329 657 040 $7\frac{41}{680}$
3	16.321 416 799 784 26	16.321 416 799 780 $5\frac{2}{41}$
10	121.331 830 876 211 1	121.331 830 876 $17\frac{4}{3} \frac{8}{9}$

Mit dem Ritz-Verfahren erhält man ca. 12 gültige Mantissenstellen der Eigenwerte. Obwohl intervallmäßige Anfangswertprobleme zu lösen sind, unterscheiden sich die berechneten oberen und unteren Schranken der Eigenwerte erst in der 15. Mantissenstelle.

Beispiel 2: Eigenwerte von $q(x) = 100 \left(\frac{2x}{\pi} - 1 \right)^2$

Schrittweiten: $h = \frac{\pi}{70}$ für λ_3 , $h = \frac{\pi}{110}$ für λ_{12} .

i	Ritz-Verfahren	$[\lambda_i]$
0	6.366 206 010 508 605	6.366 206 010 508 $\frac{365}{259}$
1	19.098 834 811 439 70	19.098 834 811 432 $\frac{38}{17}$
2	31.834 245 487 182 14	31.834 245 487 07 $\frac{8}{7}$ $\frac{22}{97}$
3	44.590 112 711 172 87	44.590 112 710 479 $\frac{46}{14}$
4	57.444 129 433 397 12	57.444 129 429 570 $\frac{76}{37}$
5	70.618 245 021 320 42	70.618 245 009 585 $\frac{61}{19}$
6	84.526 399 473 011 25	84.526 399 439 60 $\frac{2}{1}$ $\frac{21}{68}$
7	99.677 369 043 699 70	99.677 368 984 16 $\frac{7}{6}$ $\frac{19}{12}$
8	116.492 163 000 867 5	116.492 162 888 43 $\frac{5}{4}$ $\frac{5}{6}$
9	135.213 642 811 457 5	135.213 642 661 64 $\frac{6}{5}$ $\frac{7}{3}$
10	155.943 213 821 489 6	155.943 213 587 86 $\frac{9}{8}$ $\frac{5}{5}$
11	178.710 474 858 316 2	178.710 474 584 35 $\frac{9}{8}$ $\frac{2}{1}$
12	203.517 859 771 657 3	203.517 859 378 26 $\frac{8}{7}$ $\frac{2}{1}$

Auch für das hundertfache Potential aus Beispiel 1 erhält man Eigenwertnäherungen und -einschließungen vergleichbarer Genauigkeit.

Beispiel 3: Eigenwerte von $q(x) = 1000 \left(\frac{2x}{\pi} - 1 \right)^2$

Schrittweiten: $h = \frac{\pi}{170}$ für λ_{10} , $h = \frac{\pi}{210}$ für λ_0 .

i	Ritz-Verfahren	$[\lambda_i]$
0	20.131 684 841 795 00	20.131 684 841 $79\frac{5}{4}$ $\frac{13}{49}$
1	60.395 054 525 384 42	60.395 054 525 $38\frac{5}{3}$ $\frac{03}{84}$
2	100.658 424 208 974 0	100.658 424 208 $97\frac{4}{3}$ $\frac{8}{3}$
3	140.921 793 892 563 7	140.921 793 892 $56\frac{4}{2}$ $\frac{6}{8}$
10	422.765 382 297 261 6	422.765 382 297 $25\frac{9}{6}$ $\frac{7}{6}$

Sogar für das tausendfache Potential aus Beispiel 1 lassen sich Eigenwert-einschließungen mit 14 gültigen Mantissenstellen berechnen, obwohl hier erheblich kleinere Schrittweiten verwendet wurden als in Beispiel 1, und dadurch auch eine größere Zahl an Rundungsfehlern eingeschlossen werden mußte.

Bei den folgenden Beispielen 4 bis 7 dienen die Mittelwerte der berechneten Eigenwert-einschließungen in den Beispielen 22 bis 25 als Eingangsdaten zur Rekonstruktion von $q(x)$. Die erzielte Einschließungsgenauigkeit ist vergleichbar mit den Beispielen 1 bis 3.

Beispiel 4: Eigenwerte von $q(x) = \cos(2x)$

Schrittweiten: $h = \frac{\pi}{30}$ für λ_0 , $h = \frac{\pi}{90}$ für λ_{10} .

i	Ritz-Verfahren	$[\lambda_i]$
0	0.470 654 354 933 839 1	0.470 654 354 933 $8\frac{45}{32}$ $\frac{4}{8}$
1	3.979 189 215 751 357	3.979 189 215 751 $3\frac{73}{41}$
2	9.013 719 838 920 393	9.013 719 838 920 $\frac{418}{367}$
3	16.008 310 459 709 48	16.008 310 459 709 $\frac{53}{43}$
10	121.001 041 672 579 0	121.001 041 672 $57\frac{9}{8}$ $\frac{5}{5}$

Beispiel 5: Eigenwerte von $q(x) = 10 \cos(2x)$ Schrittweiten: $h = \frac{\pi}{50}$ für λ_0 , $h = \frac{\pi}{90}$ für λ_{10} .

i	Ritz-Verfahren	$[\lambda_i]$
0	-5.790 080 598 637 771	-5.790 080 598 637 $\frac{743}{801}$
1	2.099 460 445 486 665	2.099 460 445 486 $\frac{708}{622}$
2	9.236 327 713 693 701	9.236 327 713 693 $\frac{751}{651}$
3	16.648 219 937 169 78	16.648 219 937 169 $\frac{85}{70}$
10	121.104 225 893 255 9	121.104 225 893 $25\frac{7}{4}\frac{1}{6}$

Beispiel 6: Eigenwerte von $q(x) = 10 \cos(10x)$ Schrittweiten: $h = \frac{\pi}{40}$ für λ_0 , $h = \frac{\pi}{110}$ für λ_{12} .

i	Ritz-Verfahren	$[\lambda_i]$
0	0.481 708 764 497 397 7	0.481 708 76 $\frac{6}{2}\frac{237}{435}\frac{651}{621}\frac{4}{2}$
1	3.408 979 283 480 962	3.408 979 283 $\frac{636}{315}\frac{302}{904}$
2	8.229 914 987 561 617	8.229 914 987 $\frac{691}{419}\frac{988}{544}$
3	14.683 553 678 815 08	14.683 553 67 $\frac{9}{8}\frac{997}{514}\frac{31}{15}$
4	19.878 096 702 096 58	19.878 096 702 $\frac{112}{078}\frac{28}{14}$
5	37.063 858 253 605 06	37.063 858 253 $60\frac{7}{2}\frac{36}{78}$
6	49.509 561 546 370 04	49.509 561 546 $3\frac{72}{67}\frac{59}{47}$
7	64.316 053 188 488 11	64.316 053 188 $48\frac{8}{7}\frac{80}{43}$
8	81.219 850 487 643 54	81.219 850 487 $64\frac{6}{0}\frac{26}{75}$
9	99.916 681 128 455 11	99.916 681 128 $45\frac{6}{3}\frac{47}{76}$
10	121.131 023 803 482 1	121.131 023 803 $4\frac{85}{79}\frac{1}{0}$
11	144.105 276 583 735 2	144.105 276 583 $73\frac{7}{3}\frac{2}{1}$
12	169.086 901 717 837 8	169.086 901 717 $83\frac{9}{5}\frac{8}{8}$

Beispiel 7: Eigenwerte von $q(x) = 100 \cos(2x)$

Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{120}$ für λ_{10} .

i	Ritz-Verfahren	$[\lambda_i]$
0	-86.112 538 528 407 24	-86.112 538 528 407 $\frac{07}{40}$
1	-58.867 402 991 504 06	-58.867 402 991 $50\frac{3}{4}$ $\frac{83}{28}$
2	-32.717 760 511 650 37	-32.717 760 511 650 $\frac{09}{63}$
3	-7.744 936 375 464 172	-7.744 936 375 $46\frac{3}{4}$ $\frac{877}{460}$
10	131.970 676 456 726 4	131.970 676 456 $72\frac{7}{5}$ $\frac{0}{8}$

Beispiel 8: Eigenwerte von $q(x) = 1000 \cos(2x)$

Schrittweiten: $h = \frac{\pi}{230}$ für λ_{10} , $h = \frac{\pi}{290}$ für λ_0 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-955.530 062 027 493 7	-955.530 062 027 $49\frac{2}{5}$ $\frac{5}{1}$
1	-867.098 865 357 049 2	-867.098 865 357 $0\frac{47}{50}$ $\frac{6}{9}$
2	-779.694 175 891 901 3	-779.694 175 891 $\frac{899}{903}$ $\frac{4}{2}$
3	-693.334 860 235 705 9	-693.334 860 235 $70\frac{3}{8}$ $\frac{8}{2}$
10	-119.964 806 058 980 6	-119.964 806 058 $9\frac{77}{83}$ $\frac{7}{4}$

Trotz hoher Amplitude des vorgegebenen Potentials erhält man auch in diesem Beispiel enge Eigenwerteinschließungen.

Bei den folgenden Beispielen treten Eigenwertcluster (Eigenwerthäufungen) auf. Wir haben bei weiteren Berechnungen beobachtet, daß die Eigenwerte von $q(x) = a \cos(2jx)$, $j \in \mathbb{N}$, für große Werte von a Cluster von je j Eigenwerten bilden, wohingegen für stark negatives a abwechselnd Cluster von $j - 1$ und $j + 1$ Eigenwerten auftreten. Die Cluster für negative Werte

von a ergeben sich aus denjenigen für positives a , da sich wegen der Symmetrieeigenschaften der Potentiale $\pm a \cos(2jx)$ nur die Eigenwerte $\lambda_{j-1+2kj}$, $k \in \mathbb{N}_0$, unterscheiden.

Beispiel 9: Eigenwerte von $q(x) = 100 \cos(4x)$

Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{100}$ für λ_0 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-72.755 573 974 754 84	$-72.755\ 573\ 974\ 75\frac{4}{5}\ \frac{51}{20}$
1	-72.755 344 901 353 24	$-72.755\ 344\ 901\ 35\frac{2}{3}\ \frac{98}{50}$
2	-20.542 501 409 356 12	$-20.542\ 501\ 409\ 35\frac{5}{6}\ \frac{62}{62}$
3	-20.531 995 155 614 95	$-20.531\ 995\ 155\ 61\frac{4}{5}\ \frac{53}{36}$
4	26.668 886 736 186 06	$26.668\ 886\ 736\ 18\frac{6}{5}\ \frac{57}{55}$
5	26.871 797 226 587 37	$26.871\ 797\ 226\ 58\frac{7}{6}\ \frac{88}{84}$
6	67.383 516 154 074 56	$67.383\ 516\ 154\ 07\frac{5}{3}\ \frac{13}{97}$
7	69.469 340 792 567 76	$69.469\ 340\ 792\ 56\frac{8}{7}\ \frac{42}{10}$

Beispiel 10: Eigenwerte von $q(x) = -100 \cos(4x)$

Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{100}$ für λ_0 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-72.755 573 974 754 84	$-72.755\ 573\ 974\ 75\frac{4}{5}\ \frac{54}{13}$
1	-20.552 992 758 836 28	$-20.552\ 992\ 758\ 83\frac{5}{6}\ \frac{87}{71}$
2	-20.542 501 409 356 12	$-20.542\ 501\ 409\ 35\frac{5}{6}\ \frac{70}{62}$
3	-20.531 995 155 614 95	$-20.531\ 995\ 155\ 61\frac{4}{5}\ \frac{55}{39}$
4	26.668 886 736 186 06	$26.668\ 886\ 736\ 18\frac{6}{5}\ \frac{61}{52}$
5	65.678 982 684 445 22	$65.678\ 982\ 684\ 44\frac{5}{4}\ \frac{70}{69}$
6	67.383 516 154 074 56	$67.383\ 516\ 154\ 07\frac{6}{2}\ \frac{08}{99}$
7	69.469 340 792 567 76	$69.469\ 340\ 792\ 56\frac{8}{7}\ \frac{42}{09}$

Beispiel 11: Eigenwerte von $q(x) = 100 \cos(6x)$ Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{110}$ für λ_8 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-59.983 577 651 975 83	-59.983 577 651 $97\frac{5}{6} \frac{25}{41}$
1	-59.953 464 052 696 48	-59.953 464 052 $69\frac{5}{7} \frac{91}{03}$
2	-59.938 372 694 193 63	-59.938 372 694 $19\frac{3}{4} \frac{16}{09}$
3	14.051 434 973 798 47	14.051 434 973 $79\frac{9}{6} \frac{89}{93}$
4	14.842 127 316 889 88	14.842 127 316 $8\frac{91}{88} \frac{45}{16}$
5	15.257 557 390 725 69	15.257 557 390 $72\frac{7}{4} \frac{21}{03}$
6	71.850 398 235 405 79	71.850 398 235 $40\frac{6}{4} \frac{88}{71}$
7	78.100 829 111 658 47	78.100 829 111 $65\frac{9}{7} \frac{73}{21}$
8	82.719 475 725 946 67	82.719 475 725 $94\frac{7}{5} \frac{91}{44}$

Beispiel 12: Eigenwerte von $q(x) = -100 \cos(6x)$ Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{110}$ für λ_8 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-59.983 577 651 975 83	-59.983 577 651 $97\frac{5}{6} \frac{36}{30}$
1	-59.953 464 052 696 48	-59.953 464 052 $69\frac{6}{6} \frac{02}{94}$
2	13.674 114 791 737 10	13.674 114 791 $73\frac{8}{5} \frac{46}{49}$
3	14.051 434 973 798 47	14.051 434 973 $\frac{800}{796} \frac{39}{32}$
4	14.842 127 316 889 88	14.842 127 316 $8\frac{91}{87} \frac{82}{69}$
5	15.257 557 390 725 69	15.257 557 390 $72\frac{7}{4} \frac{16}{01}$
6	71.850 398 235 405 79	71.850 398 235 $40\frac{7}{4} \frac{16}{39}$
7	78.100 829 111 658 47	78.100 829 111 $65\frac{9}{6} \frac{95}{99}$
8	108.520 487 703 676 5	108.520 487 703 $67\frac{7}{5} \frac{3}{6}$

Beispiel 13: Eigenwerte von $q(x) = 100 \cos(8x)$ Schrittweiten: $h = \frac{\pi}{100}$ für λ_0 , $h = \frac{\pi}{110}$ für λ_8 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-48.056 006 080 762 19	$-48.056\ 006\ 080\ 76\frac{0}{3}\frac{68}{86}$
1	-47.809 476 867 329 27	$-47.809\ 476\ 867\ 3\frac{27}{31}\frac{42}{28}$
2	-47.558 552 463 412 17	$-47.558\ 552\ 463\ 4\frac{09}{15}\frac{46}{07}$
3	-47.453 284 599 180 74	$-47.453\ 284\ 599\ 1\frac{79}{82}\frac{41}{24}$
4	41.475 338 389 824 34	$41.475\ 338\ 389\ 8\frac{27}{0}\frac{40}{96}$
5	44.988 902 349 064 01	$44.988\ 902\ 349\ 0\frac{69}{57}\frac{85}{80}$
6	49.306 788 092 145 10	$49.306\ 788\ 092\ 1\frac{52}{37}\frac{24}{52}$
7	51.482 930 861 794 71	$51.482\ 930\ 861\ 7\frac{98}{0}\frac{56}{44}$
8	103.259 157 878 586 6	$103.259\ 157\ 878\ 5\frac{8}{4}\frac{2}{9}$

Beispiel 14: Eigenwerte von $q(x) = -100 \cos(8x)$ Schrittweiten: $h = \frac{\pi}{100}$ für λ_0 , $h = \frac{\pi}{110}$ für λ_8 .

i	Ritz-Verfahren	$[\lambda_i]$
0	-48.056 006 080 762 19	$-48.056\ 006\ 080\ 76\frac{0}{3}\frac{87}{68}$
1	-47.809 476 867 329 27	$-47.809\ 476\ 867\ 3\frac{27}{30}\frac{85}{86}$
2	-47.558 552 463 412 17	$-47.558\ 552\ 463\ 4\frac{10}{3}\frac{84}{67}$
3	40.176 272 988 546 07	$40.176\ 272\ 988\ 5\frac{48}{3}\frac{57}{33}$
4	41.475 338 389 824 34	$41.475\ 338\ 389\ 8\frac{30}{17}\frac{75}{52}$
5	44.988 902 349 064 01	$44.988\ 902\ 349\ 0\frac{74}{53}\frac{88}{34}$
6	49.306 788 092 145 10	$49.306\ 788\ 092\ 1\frac{51}{38}\frac{35}{42}$
7	51.482 930 861 794 71	$51.482\ 930\ 861\ 7\frac{98}{0}\frac{45}{53}$
8	103.259 157 878 586 6	$103.259\ 157\ 878\ 5\frac{90}{82}\frac{3}{8}$

Beispiel 15: Eigenwerte von $q(x) = 100 \cos(10x)$

Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{130}$ für λ_{12} .

i	Ritz-Verfahren	$[\lambda_i]$
0	-37.575 466 563 675 70	-37.575 466 563 $6\frac{69}{84} \frac{41}{87}$
1	-36.840 425 617 895 78	-36.840 425 617 $8\frac{70}{923} \frac{28}{67}$
2	-35.890 355 745 787 23	-35.890 355 745 $6\frac{96}{890} \frac{33}{43}$
3	-35.084 400 836 100 67	-35.084 400 836 $0\frac{22}{189} \frac{45}{71}$
4	-34.766 912 530 633 04	-34.766 912 530 $6\frac{01}{76} \frac{79}{21}$
5	61.476 539 932 500 51	61.476 539 932 $5\frac{12}{486} \frac{58}{22}$
6	67.228 837 504 420 99	67.228 837 504 $4\frac{47}{391} \frac{59}{37}$
7	76.006 625 771 173 22	76.006 625 771 $7\frac{22}{119} \frac{62}{78}$
8	86.083 865 527 590 04	86.083 865 527 $6\frac{08}{563} \frac{88}{80}$
9	91.805 817 662 429 77	91.805 817 662 $4\frac{38}{19} \frac{99}{98}$
10	137.685 602 866 786 2	137.685 602 866 $7\frac{0}{0} \frac{5}{9}$
11	155.782 958 866 912 2	155.782 958 866 $9\frac{01}{890} \frac{2}{1}$
12	178.178 139 296 243 7	178.178 139 296 $2\frac{35}{28} \frac{2}{8}$

Auch bei diesem Potential mit hoher Frequenz und großer Amplitude erhält man (ebenso wie im nächsten Beispiel) bei relativ großer Schrittweite mit dem Verfahren von Lohner Eigenwerteinschließungen mit bis zu 14 gültigen Mantissenstellen.

Beispiel 16: Eigenwerte von $q(x) = -100 \cos(10x)$

Schrittweiten: $h = \frac{\pi}{90}$ für λ_3 , $h = \frac{\pi}{130}$ für λ_{12} .

i	Ritz-Verfahren	$[\lambda_i]$
0	-37.575 466 563 675 70	-37.575 466 563 $6\frac{69}{84}$ $\frac{85}{41}$
1	-36.840 425 617 895 78	-36.840 425 617 $\frac{876}{917}$ $\frac{63}{29}$
2	-35.890 355 745 787 23	-35.890 355 745 $\frac{694}{890}$ $\frac{64}{51}$
3	-35.084 400 836 100 67	-35.084 400 836 $\frac{068}{143}$ $\frac{97}{22}$
4	59.479 997 012 217 29	59.479 997 012 $2\frac{28}{03}$ $\frac{95}{68}$
5	61.476 539 932 500 51	61.476 539 932 $\frac{547}{451}$ $\frac{25}{49}$
6	67.228 837 504 420 99	67.228 837 504 $\frac{489}{349}$ $\frac{87}{39}$
7	76.006 625 771 173 22	76.006 625 771 $\frac{256}{085}$ $\frac{67}{70}$
8	86.083 865 527 590 04	86.083 865 527 $\frac{613}{559}$ $\frac{54}{33}$
9	91.805 817 662 429 77	91.805 817 662 $4\frac{38}{20}$ $\frac{75}{08}$
10	137.685 602 866 786 2	137.685 602 866 $7\frac{75}{52}$ $\frac{5}{9}$
11	155.782 958 866 912 2	155.782 958 866 $\frac{901}{890}$ $\frac{2}{1}$
12	178.178 139 296 243 7	178.178 139 296 $2\frac{35}{28}$ $\frac{2}{8}$

4.2 Beispiele zum inversen Dirichletproblem

Die in den folgenden Beispielen aufgeführten Potentiale haben wir mit dem Algorithmus 3.2 rekonstruiert. Zur Rekonstruktion haben wir die arithmetischen Mittel der zuvor berechneten Eigenwertschranken des durch $q(x)$ definierten Dirichletproblems vorgegeben.

Wir haben zwei verschiedene Ansätze für das Potential getestet, Splines und trigonometrische Funktionen. Mit beiden haben wir gute Einschließungen für die gesuchten Potentiale erhalten.

Wir haben symmetrische, kubische, zweimal stetig differenzierbare Splines (siehe z.B. Ahlberg, Nilson und Walsh [1]) mit äquidistanten Stützstellen im Intervall $[0, \pi]$ verwendet, und diese in Intervallsplines eingeschlossen.

Durch die Symmetriebedingung sind die Funktionswerte der im Intervall $(\frac{\pi}{2}, \pi]$ liegenden Stützstellen aus den Funktionswerten der Stützstellen im Intervall $[0, \frac{\pi}{2}]$ bestimmt. Als Basisfunktion $q_j(x)$ haben wir denjenigen Spli-

ne gewählt, der an der j -ten Stützstelle im Intervall $[0, \frac{\pi}{2}]$ den Wert 1 annimmt, und an den übrigen Stützstellen in $[0, \frac{\pi}{2}]$ den Wert 0.

Mit vier Eigenwerten lassen sich also beispielsweise Splines mit sieben oder acht äquidistanten Stützstellen in $[0, \pi]$ bestimmen. Bereits diese Splines lieferten gute Ergebnisse bei der Rekonstruktion, sofern sie zur Approximation der vorgegebenen Potentiale überhaupt geeignet waren.

Da die Splines nur stückweise unendlich oft differenzierbar sind, konnten wir die Anfangswertprobleme nur stückweise mit dem Lohnerschen Verfahren lösen. Wir haben dabei die in [29] beschriebene „Auswertung 4“ verwendet. Gute Ergebnisse erhielten wir, solange die Norm der Potentiale nicht zu groß war. Andernfalls konnten die Eigenfunktionen mit den von uns getesteten Schrittweiten nicht mehr genau genug eingeschlossen werden, um beim Rekonstruktionsverfahren eine nichtsinguläre Funktionalmatrix zu erhalten. Dieses Problem trat verstärkt auf, wenn die Zahl der Stützstellen der Splines zunahm, da dann mehr Bruchstellen zu berücksichtigen waren.

Mit (auf dem gesamten Intervall unendlich oft differenzierbaren) trigonometrischen Funktionen konnten auch Einschließungen bei etwas größerer Norm des Potentials erzielt werden. Wir haben speziell

$$q_0(x) = 1, \quad q_j(x) = \cos(2jx), \quad j = 1, \dots, n,$$

verwendet, um die führenden Terme der Fourierreihen der gesuchten Potentiale zu bestimmen.

Als Startnäherung konnte das Nullpotential eingesetzt werden, solange die Norm des zu rekonstruierenden Potentials nicht zu groß war. Die in den übrigen Fällen verwendeten Startpotentiale sind in den jeweiligen Beispielen angegeben.

In den folgenden Tabellen sind bei den Splinepotentialen die Funktionswerte des zu rekonstruierenden Potentials $q(x)$ in den Stützstellen x_i des Splines und die Funktionswerte des berechneten Intervallsplines $[q_s]$ in diesen Stützstellen aufgelistet. Bei der Rekonstruktion mit trigonometrischen Basisfunktionen vergleichen wir die Fourierkoeffizienten c_j des Potentials

$$q(x) = c_0 + \sum_{j=1}^{\infty} c_j \cos(2jx)$$

mit den berechneten Koeffizienten a_j des rekonstruierten Potentials

$$a_0 + \sum_{j=1}^n a_j \cos(2jx).$$

Wir haben bei der Lösung der Anfangswertprobleme ungefähr die gleichen Schrittweiten wie bei den direkten Problemen verwendet. Geringfügige Anpassungen waren bei den Splinepotentialen notwendig, da bei diesen wegen der Sprungstellen in der dritten Ableitung die Stützstellen der Splines auch Stützstellen im Verfahren von Lohner sein mußten.

Beispiel 17: Rekonstruktion von $q(x) = \left(\frac{2}{\pi}x - 1\right)^2$

Anzahl der vorgegebenen Eigenwerte: 4 (siehe Beispiel 1).

Ansatz für das Potential: Spline mit acht Stützstellen,

$$x_i = \frac{i\pi}{7}, i = 0, 1, \dots, 7.$$

Startnäherung: $q^{(0)}(x) = 0$.

i	$q(x_i)$	$[q_s](x_i)$
0	1.000 000 000 000 000 ...	1.000 000 000 003 736 0.999 999 999 996 606 2
1	0.510 204 081 632 653 ...	0.510 204 081 632 $\frac{967}{323}$ $\frac{2}{1}$
2	0.183 673 469 387 755 ...	0.183 673 469 $38\frac{8}{7}$ $\frac{037}{485}$ $\frac{4}{9}$
3	0.020 408 163 265 306 ...	0.020 408 163 265 $\frac{450}{156}$ $\frac{00}{55}$

Das vorgegebene Potential ist als Spline exakt darstellbar, und man erhält eine Einschließung auf ca. 12 gültige Mantissenstellen.

Beispiel 18: Rekonstruktion von $q(x) = 100\left(\frac{2}{\pi}x - 1\right)^2$

Anzahl der vorgegebenen Eigenwerte: 4 (siehe Beispiel 2).

Ansatz für das Potential: Spline mit acht Stützstellen,

$$x_i = \frac{i\pi}{7}, i = 0, 1, \dots, 7.$$

Startnäherung: $q^{(0)}(x) = 0$.

i	$q(x_i)$	$[q_s](x_i)$
0	100.000 000 000 000 000 ...	$\frac{100.000\ 000\ 000\ 149\ 3}{99.999\ 999\ 999\ 851\ 92}$
1	51.020 408 163 265 306 ...	$51.020\ 408\ 163\ 2\frac{75}{55}\frac{49}{22}$
2	18.367 346 938 775 510 ...	$18.367\ 346\ 938\ 77\frac{6}{4}\frac{28}{75}$
3	2.040 816 326 530 612 ...	$2.040\ 816\ 326\ 530\ \frac{814}{413}$

Das hier vorliegende Potential ist das Hundertfache des Potentials aus Beispiel 17 und somit ebenfalls als Spline exakt darstellbar. Wie Beispiel 2 zeigt, sind die Eigenwerte im Vergleich zum Nullpotential stark gestört. Dennoch konvergiert das Rekonstruktionsverfahren mit der Startnäherung $q(x) = 0$, und der relative Fehler der erzielten Einschließung ist sogar kleiner als in Beispiel 17.

Beispiel 19: Rekonstruktion von $q(x) = 100\left(\frac{2}{\pi}x - 1\right)^2$

Anzahl der vorgegebenen Eigenwerte: 2 (siehe Beispiel 2).

Ansatz für das Potential: $q(x) = a_0 + a_1 \cos(2x)$.

Startnäherung: $q^{(0)}(x) = 0$.

j	Fourierkoeffizienten c_j	Rekonstruierte Koeff. a_j
0	33.333 333 333 333 ...	$23.592\ 977\ 955\ 09\frac{9}{8}\frac{52}{83}$
1	40.528 473 456 935 108 ...	$23.877\ 058\ 039\ 38\frac{1}{0}\frac{16}{26}$

Bereits bei der Vorgabe von nur zwei Eigenwerten läßt das rekonstruierte Potential die ungefähre Gestalt des Ausgangspotentials erkennen. Besonders in der Mitte des Intervalls $[0, \pi]$ erhält man eine gute Approximation von $q(x)$. Wir weisen darauf hin, daß das berechnete Intervallpotential nicht nur als Approximation von $q(x) = 100\left(\frac{2}{\pi} - 1\right)^2$ betrachtet werden kann, sondern auch ein Potential der Gestalt $b_0 + b_1 \cos(2x)$ einschließt, das die vorgegebenen Eigenwerte exakt annimmt.

Beispiel 20: Rekonstruktion von $q(x) = 100\left(\frac{2}{\pi}x - 1\right)^2$

Anzahl der vorgegebenen Eigenwerte: 6 (siehe Beispiel 2).

Ansatz für das Potential: $q(x) = a_0 + \sum_{j=1}^5 a_j \cos(2jx)$.

Startnäherung: $q^{(0)}(x) = 0$.

j	Fourierkoeffizienten c_j	Rekonstruierte Koeff. a_j
0	33.333 333 333 333 333 ...	32.388 117 081 $1\frac{84}{77} \frac{90}{15}$
1	40.528 473 456 935 108 ...	38.649 323 296 $3\frac{41}{26} \frac{54}{35}$
2	10.132 118 364 233 777 ...	8.290 774 189 $\frac{512}{497} \frac{073}{795}$
3	4.503 163 717 437 234 ...	2.738 305 284 $9\frac{59}{47} \frac{882}{347}$
4	2.533 029 591 058 444 ...	0.903 153 465 $7\frac{44}{35} \frac{336}{269} \frac{8}{5}$
5	1.621 138 938 277 404 ...	0.205 265 810 $7\frac{14}{08} \frac{500}{902} \frac{8}{4}$

Durch sechs Eigenwerte lassen sich die führenden Fourierkoeffizienten des Potentials bereits recht genau bestimmen. Der relative Fehler beträgt ca. 3 % bei a_0 , ca. 5 % bei a_1 .

Beispiel 21: Rekonstruktion von $q(x) = 100\left(\frac{2}{\pi}x - 1\right)^2$

Anzahl der vorgegebenen Eigenwerte: 13 (siehe Beispiel 2).

Ansatz für das Potential: $q(x) = a_0 + \sum_{j=1}^{12} a_j \cos(2jx)$.

Startnäherung: $q^{(0)}(x) = 0$.

j	Fourierkoeffizienten c_j	Rekonstruierte Koeff. a_j
0	33.333 333 333 333 333 ...	33.194 805 008 $1\frac{54}{45}$ $\frac{86}{89}$
1	40.528 473 456 935 108 ...	40.251 422 374 $7\frac{73}{53}$ $\frac{90}{62}$
2	10.132 118 364 233 777 ...	9.855 085 409 $\frac{722}{699}$ $\frac{726}{404}$
3	4.503 163 717 437 234 ...	4.226 165 750 $8\frac{27}{04}$ $\frac{658}{607}$
4	2.533 029 591 058 444 ...	2.256 091 594 $8\frac{95}{70}$ $\frac{975}{261}$
5	1.621 138 938 277 404 ...	1.344 300 408 $6\frac{83}{55}$ $\frac{051}{607}$

Werden dreizehn Eigenwerte vorgegeben, verbessert sich der relative Fehler auf weniger als 0.5 % bei a_0 . Sogar a_5 ist auf ca. 17 % genau, obwohl der Anteil von $c_5 \cos 10(x)$ am Potential nur ungefähr 1.6 % beträgt.

Beispiel 22: Rekonstruktion von $q(x) = \cos(2x)$

Anzahl der vorgegebenen Eigenwerte: 4 (siehe Beispiel 4).

Ansatz für das Potential: Spline mit acht Stützstellen,

$$x_i = \frac{i\pi}{7}, i = 0, 1, \dots, 7.$$

Startnäherung: $q^{(0)}(x) = 0$.

i	$q(x_i)$	$[q_s](x_i)$
0	1.000 000 000 000 000 0...	0.990 579 065 $9\frac{61}{54}$ $\frac{872}{175}$ $\frac{2}{7}$
1	0.623 489 801 858 733 5...	0.624 418 556 $3\frac{10}{09}$ $\frac{550}{885}$ $\frac{3}{9}$
2	-0.222 520 933 956 314 4...	-0.222 771 163 $94\frac{1}{2}$ $\frac{496}{098}$ $\frac{6}{5}$
3	-0.900 968 867 902 419 1...	-0.901 949 916 271 $\frac{284}{587}$ $\frac{1}{9}$

Das berechnete Potential approximiert die Funktion $\cos(2x)$ schon bei der der Vorgabe von nur vier Eigenwerten recht gut (und zwar auf dem gesamten Intervall). Wie in Beispiel 19 ist der gefundene Intervallspline nicht nur eine Approximation von $\cos(2x)$. Er schließt einen reellen Spline ein, der die vorgegebenen Eigenwerte besitzt.

Beispiel 23: Rekonstruktion von $q(x) = 10 \cos(2x)$

Anzahl der vorgegebenen Eigenwerte: 4 (siehe Beispiel 5).

Ansatz für das Potential: Spline mit acht Stützstellen,

$$x_i = \frac{i\pi}{7}, i = 0, 1, \dots, 7.$$

Startnäherung: $q^{(0)}(x) = 0$.

i	$q(x_i)$	$[q_s](x_i)$
0	10.000 000 000 000 000 ...	9.915 642 387 3_{20}^{43} 824^{148}
1	6.234 898 018 587 335 ...	6.244 098 617 01_3^4 874^{777}
2	-2.225 209 339 563 144 ...	-2.227 686 004 54_3^2 343^{896}
3	-9.009 688 679 024 191 ...	-9.019 509 515 53_4^3 046^{815}

Die Verzehnfachung der Amplitude hat keinen entscheidenden Einfluß auf die Genauigkeit der Approximation.

Beispiel 24: Rekonstruktion von $q(x) = 10 \cos(10x)$

Anzahl der vorgegebenen Eigenwerte: 13 (siehe Beispiel 6).

Ansatz für das Potential: Spline mit 25 Stützstellen,

$$x_i = \frac{i\pi}{24}, i = 0, 1, \dots, 24.$$

Startnäherung: $q^{(0)}(x) = 0$.

i	$q(x_i)$	$[q_s](x_i)$
0	10.000 000 000 000 000 ...	10.722 013 3_{38}^{44} 309^{836} 58^{92}
3	-7.071 067 811 865 475 ...	-7.113 333 05_1^0 953^{920} 722^{819}
6	0.000 000 000 000 000 ...	-0.000 093 17_6^4 970^{839} 366^{780}
9	7.071 067 811 865 475 ...	7.113 695 57_5^7 774^{932} 395^{883}
12	-10.000 000 000 000 000 ...	-10.060 191 21_4^2 917^{427} 69^{11}

Mit nur 25 Stützstellen im Intervall $[0, \pi]$ erhält man eine Approximation mit einem relativen Fehler von ca. 7.2 % an den Intervallenden und ca. 0.6 % in der Intervallmitte. Dabei ist zu berücksichtigen, daß für eine volle Periode der Kosinusschwingung nur fünf Stützstellen zur Verfügung stehen.

Beispiel 25: Rekonstruktion von $q(x) = 100 \cos(2x)$

Anzahl der vorgegebenen Eigenwerte: 4 (siehe Beispiel 7).

Ansatz für das Potential: $q(x) = a_0 + \sum_{j=1}^3 a_j \cos(2jx)$.

Startnäherung: $q^{(0)}(x) = 99 \cos(2x)$.

Rekonstruiertes Potential
$100 \cos(2x) + [-5.618 \cdot 10^{-11}, 5.616 \cdot 10^{-11}]$ $+ [-9.364 \cdot 10^{-11}, 9.370 \cdot 10^{-11}] \cos(2x)$ $+ [-5.149 \cdot 10^{-11}, 5.153 \cdot 10^{-11}] \cos(4x)$ $+ [-1.500 \cdot 10^{-11}, 1.504 \cdot 10^{-11}] \cos(6x)$

Das zu rekonstruierende Potential liegt in der linearen Hülle der Basisfunktionen. Man erhält eine Einschließung auf ca. 12 gültige Mantissenstellen. Das Rekonstruktionsverfahren divergiert allerdings, wenn man es mit dem Nullpotential startet. Dies ist vermutlich dadurch zu erklären, daß die vorgegebenen Eigenwerte stark von denen des Nullpotentials abweichen (siehe Beispiel 7).

Beispiel 26: Rekonstruktion von $q(x) = 100 \cos(10x)$

Anzahl der vorgegebenen Eigenwerte: 6 (siehe Beispiel 15).

Ansatz für das Potential: $q(x) = a_0 + \sum_{j=1}^5 a_j \cos(2jx)$.

Startnäherung: $q^{(0)}(x) = 99 \cos(10x)$.

Rekonstruiertes Potential
$100 \cos(10x) + [-2.475 \cdot 10^{-09}, 2.454 \cdot 10^{-09}]$
$+ [-6.277 \cdot 10^{-09}, 6.224 \cdot 10^{-09}] \cos(2x)$
$+ [-6.229 \cdot 10^{-09}, 6.175 \cdot 10^{-09}] \cos(4x)$
$+ [-6.811 \cdot 10^{-09}, 6.754 \cdot 10^{-09}] \cos(6x)$
$+ [-8.063 \cdot 10^{-09}, 7.998 \cdot 10^{-09}] \cos(8x)$
$+ [-3.624 \cdot 10^{-09}, 3.595 \cdot 10^{-09}] \cos(10x)$

Dieses Beispiel zeigt, daß auch bei Potentialen mit hoher Frequenz und großer Amplitude enge Einschließungen erzielbar sind. Wie in Beispiel 25 weichen die Eigenwerte stark von denen des Nullpotentials ab, und das Rekonstruktionsverfahren divergiert, wenn man es mit $q^{(0)}(x) = 0$ startet.

Anhang

Programmdokumentation

Wir beschreiben hier die für das Rekonstruktionsverfahren mit trigonometrischen Basisfunktionen maßgeblichen Teile des Programmtexts, mit denen auch die Beispiele 19-21, 25 und 26 gerechnet wurden. Das gesamte Programmsystem umfaßt das Hauptprogramm IEPT und die Module DEPT, IEP_RAWA, IEP_IAWA, IEP_TAYK, SERVICE und SEIG_APP.

In SERVICE stehen Ein-/Ausgabe-Routinen für (Intervall-)vektoren und -matrizen sowie einige Funktionen (Potenzfunktion, Maximumsnorm bei Vektoren, etc.), die in PASCAL-XSC nicht vordefiniert sind. Da deren Definition aus den gewählten Prozedurnamen ableitbar ist, verzichten wir auf die Veröffentlichung dieses Moduls.

In IEP_TAYK werden die dynamischen Datentypen

```
GLOBAL TYPE RTKO = GLOBAL DYNAMIC ARRAY[*] OF REAL;  
             RTVEC = GLOBAL DYNAMIC ARRAY[*] OF RTKO;  
             RTMAT = GLOBAL DYNAMIC ARRAY[*] OF RTVEC;  
             ITKO = GLOBAL DYNAMIC ARRAY[*] OF INTERVAL;  
             ITVEC = GLOBAL DYNAMIC ARRAY[*] OF ITKO;  
             ITMAT = GLOBAL DYNAMIC ARRAY[*] OF ITVEC;
```

zur Speicherung von Taylorkoeffizienten einer (mehrerer) reellwertigen Funktion(en) f_j in einer (mehreren) reellen Stützstelle(n) x_i festgelegt und arithmetische Operatoren für die jeweiligen Datentypen definiert. Außerdem enthält IEP_TAYK Prozeduren zur Berechnung von Taylorkoeffizienten einiger Standardfunktionen. Die entsprechenden mathematischen Definitionen lassen sich ohne wesentliche Modifikationen als PASCAL-XSC-Programm

formulieren, so daß wir auch den Programmtext dieses Moduls nicht angeben.

IEP_IAWA dient der verifizierten Lösung der Anfangswertprobleme (4.1). Es handelt sich dabei um die an die spezielle Gestalt der Anfangswertprobleme (4.1) angepaßte Version von Lohners Programm AWA, siehe [29]. Zur approximativen Lösung der Anfangswertprobleme wurden aus dem Programmtext von IEP_IAWA diejenigen Anweisungen, die der Fehlerbehandlung dienen, gestrichen und die entstehenden Prozeduren im Modul IEP_RAWA zusammengefaßt. Wir veröffentlichen nur den Programmtext der Intervallprozeduren.

Das Modul DEPT enthält die Prozeduren, die zur Behandlung des direkten Dirichletproblems benötigt werden, d.h. zur Berechnung der Eigenwerte und Eigenfunktionen der iterierten Potentiale im Rekonstruktionsverfahren. DEPT schließt auch Prozeduren für das Ritzverfahren ein. Die Berechnung der beim Ritzverfahren zu bestimmenden Eigenwerte und Eigenvektoren von symmetrischen Matrizen wird im Modul SEIG_APP durchgeführt. Zur Berechnung der Eigenpaarnäherungen wird das Jacobi-Verfahren verwendet (siehe auch Kapitel 4 bzw. Lohner [30]).

Der vollständigen Rekonstruktionsalgorithmus 3.2 ist im Hauptprogramm IEPT implementiert. Ausgehend von einer einzulesenden Startnäherung $a^{(0)}$, die im Programm in der Variablen FCP_AKT abgespeichert ist, werden zur Durchführung der drei Iterationen (Ritzverfahren, reelles Newtonverfahren 2.1, Maschinen-Intervall-Newtonverfahren 2.2) die entsprechenden Prozeduren in den angegebenen Modulen aufgerufen. Zuvor werden im Hauptprogramm die zur numerischen Lösung der Anfangswertprobleme (4.1) benötigten Taylorkoeffizienten des iterierten Potentials $q(x; a^{(k)})$ in den Variablen TRVECP bzw. TIVECP, TIVECP_MID, TIVECP_RST und TIVECP_RST_MID bereitgestellt.

Bemerkung: Bei der Veröffentlichung der Programmteile haben wir sämtliche Ausgabeanweisungen aus den Programmtexten gestrichen, um die Darstellung übersichtlich zu halten. Dies gilt auch für einige „Notbremsen“, um eventuell endlose Iterationen abubrechen oder Speicherüberläufe (bei dynamischen Feldern) zu verhindern.

Wir weisen darauf hin, daß bei der angegebenen Programmversion in einigen Prozeduren die Endergebnisse in Files geschrieben werden müssen, aus denen sie in anderen Prozeduren später wieder als Eingabedaten eingelesen werden. Diese Art der Datenübergabe erübrigt sich, falls auf dem

verwendeten Rechner genügend Hauptspeicher zur Verfügung steht, um alle benötigten Zwischengrößen ständig bereitzuhalten. □

Wir beginnen die Veröffentlichung des Programmtextes mit dem Hauptprogramm IEPT:

```

{-----}
{ PASCAL-XSC - PROGRAMM IEPT }
{ VOLLSTAENDIGES REKONSTRUKTIONSVERFAHREN MIT BASISFUNKTIONEN cos(2jx) }
{-----}

PROGRAM IEPT_AGP( INPUT , EIN , OUTPUT );
USE i_ari , mv_ari , mvi_ari , lss , ilss , seig_app;
USE dept , service , iep_tayk , iep_rawa , iep_iawa;
CONST M = 10;      { ANZAHL DER STUETZSTELLEN ZUR SCHRITTWEITENSTEUERUNG}
      N = 5;      { N = M DIV 2 = ANZAHL DER EW - 1 }
      P = 15;    { ORDNUNG DER TAYLORENTWICKLUNG VON U }

VAR EW , I , J , COUNT , ERR , ST      : INTEGER;
    DIVI , DIVI_2                      : INTEGER;
    HNORM , ETA , ETA_ALT              : REAL;
    PI , H , T , VR_U2 , WIDTH_AKT    : REAL;
    NCL                                : BOOLEAN;
    IPI , IH , IHI , IT                : INTERVAL;
    EIG_ZIEL , EIG_AKT , FCP_AKT      : RVECTOR[0..N];
    ZUWACHS_MID                       : RVECTOR[0..N];
    ZUWACHS                            : IVECTOR[0..N];
    FC_Q , FUNKT_MAT                   : RMATRIX[ 0..N , 0..N ];
    DIVISOR                            : ARRAY[0..N] OF INTEGER;
    VRU                                : RVECTOR[1..2];
    VR_U2Q                             : RVECTOR[0..N];
    TRVECP                             : RTVEC[ 0..6000 , 0..P ];
    TRMATQ                             : RTMAT[ 0..N , 0..6000 , 0..P ];
    VIU                                : IVECTOR[1..2];
    VI_U2                              : INTERVAL;
    VI_U2Q                             : IVECTOR[0..N];
    IFCP_AKT , IFCP_ALT , IFCP_MID     : IVECTOR[0..N];
    IEIG_AKT , IEIG_AKT_MID           : IVECTOR[0..N];
    IFC_Q , ILAST_FUNKT_MAT , IFUNKT_MAT : IMATRIX[ 0..N , 0..N ];
    TIVECP , TIVECP_RST               : ITVEC[ 0..6000 , 0..P ];
    TIVECP_MID , TIVECP_RST_MID       : ITVEC[ 0..6000 , 0..P ];
    TIMATQ                             : ITMAT[ 0..N , 0..6000 , 0..P ];
    EIN                                : TEXT;
{ EW : NR DES EIGENWERTS }
{ VR_U2 = U^2 , VR_U2Q[I] = Q[I] * U^2 }
{ 6000 : MAXIMAL ZULAESSIGE DIMENSION }

FUNCTION TAYKFCPT( T : REAL; FCP_AKT : RVECTOR ) : RTKO[0..P];
{ BERECHNET DIE TAYLORKOEFFIZIENTEN DER }
{ TRIGONOMETRISCHEN POTENTIALFUNKTION P(X) }

VAR I : INTEGER;
    TRP : RTKO[0..P];

BEGIN
    TRP := TKO( P , FCP_AKT[0] );
    FOR I := 1 TO N DO
        TRP := TRP + FCP_AKT[I] * COSKO( P , T , 2 * I / DIVI );
    FOR I := 0 TO P DO
        TRP[I] := TRP[I] / DIVI_2;
    TAYKFCPT := TRP;
END;
```

```

FUNCTION TAYKFCPT( IT : INTERVAL; IFCP_AKT : IVECTOR ) : ITKO[0..P];
{ BERECHNET DIE TAYLORKOEFFIZIENTEN DER      }
{ TRIGONOMETRISCHEN POTENTIALFUNKTION P(X) }
VAR I   : INTEGER;
    TIP : ITKO[0..P];
BEGIN
  TIP := TKO( P , IFCP_AKT[0] );
  FOR I := 1 TO N DO
    TIP := TIP + IFCP_AKT[I] * COSKO( P , IT , 2 * I / INTVAL( DIVI ) );
  FOR I := 0 TO P DO
    TIP[I] := TIP[I] / DIVI_2;
  TAYKFCPT := TIP;
END;

FUNCTION RESTGLIED( IT : INTERVAL; IFCP_AKT : IVECTOR ) : INTERVAL;
{ BERECHNET DEN P-TEN TAYLORKOEFFIZIENTEN DER }
{ TRIGONOMETRISCHEN POTENTIALFUNKTION P(X)   }
VAR I   : INTEGER;
    TIP : ITKO[0..P];
BEGIN
  TIP := TKO( P , IFCP_AKT[0] );
  FOR I := 1 TO N DO
    TIP := TIP + IFCP_AKT[I] * COSKO( P , IT , 2 * I / INTVAL( DIVI ) );
  RESTGLIED := TIP[P]; { HIER KEINE DIVISION DURCH DIVI_2! }
END;

{-----}
{ HAUPTPROGRAMM }
{-----}

BEGIN RESET( EIN );
  { VARIABLEN VORBESETZEN }
  PI := 3.141592653589793;
  IPI := INTVAL( RVAL('3.141592653589793',-1), RVAL('3.141592653589794',1) );
  ETA := 1;
  { GLOBALE SCHRITTWEITE SETZEN }
  H := PI / M;
  IH := IPI / M;
  IHI := INTVAL( 0 , IH.SUP );
  { EIGENWERTE UND NAEHERUNGEN DER FUNKTIONSWERTE }
  { IN DEN SIGNIFIKANTEN STUETZSTELLEN EINLESEN }
  FOR I := 0 TO N DO BEGIN
    READLN( EIG_ZIEL[I] );
    READLN( EIN , FCP_AKT[I] );
  END;

{-----}
{ SCHRITT 1 : BERECHNUNG EINES GROBEN NAEHERUNGSPOTENTIALS MIT DEM }
{ RITZ-VERFAHREN }
{-----}

COUNT := 0;
REPEAT { "ENDLOS"-SCHLEIFE }
  COUNT := COUNT + 1;
  { RITZ-EIGENWERTE UND RITZ-NAEHERUNG FUER DIE FUNKTIONALMATRIX }
  { DES POTENTIALS MIT 30 ANSATZFUNKTIONEN BEIM RITZ-VERFAHREN }
  { IN DER PROZEDUR DEPT_RITZ IM MODUL DEPT BERECHNEN }
  DEPT_RITZ( FCP_AKT , N , 30 , EIG_AKT , FUNKT_MAT );
  { NAECHSTE ITERIERTE DES POTENTIALS BERECHNEN }
  LSS( FUNKT_MAT , EIG_ZIEL - EIG_AKT , ZUWACHS , ERR );
  EIG_AKT := EIG_ZIEL;
  ZUWACHS_MID := MID( ZUWACHS );
  FCP_AKT := FCP_AKT + ZUWACHS_MID;
  { HILFSGROESSEN FUER DEN EINSCHLIESSUNGSSCHRITT BERECHNEN }
  ETA_ALT := ETA;

```

```

ETA := MAX_NORM( ZUWACHS_MID );
UNTIL MAX_NORM( MID ( ZUWACHS ) ) < 1.0E-04 * MAX_NORM( FCP_AKT );
-----
{ SCHRITT 2 : BERECHNUNG EINES GUTEN NAEHERUNGSPOTENTIALS DURCH
  NAEHERUNGSWEISE LOESUNG VON ANFANGSWERTPROBLEMEN
}
-----
{ BASISPOTENTIALIA BERECHNEN }
FOR I := 0 TO N DO BEGIN
  FOR J := 0 TO N DO
    FC_Q[I,J] := 0;
  FC_Q[I,I] := 1;
END;
{ HEURISTISCH OPTIMALEN DIVISOR BERECHNEN }
{ DIVISOREN STRECKEN DAS ZEITINTERVALL DER DGL, }
{ STEuern ALSO INDIREKT DIE SCHRITTWEITE }
FOR EW := 0 TO N DO BEGIN
  HNORM := ABS( FCP_AKT[0] - EIG_ZIEL[EW] );
  FOR I := 1 TO N DO
    HNORM := HNORM + ABS( FCP_AKT[I] );
  HNORM := SQRT( HNORM );
  DIVISOR[EW] := 1 + ROUND( ( 6 * HNORM + 15 ) / M );
END;
COUNT := 0;
REPEAT { "ENDLOS"-SCHLEIFE }
  COUNT := COUNT + 1;
  FOR EW := 0 TO N DO BEGIN
    DIVI := DIVISOR[EW];
    DIVI_2 := SQR( DIVI );
    { TAYLORKOEFFIZIENTEN DER POTENTIALFUNKTION BEREITSTELLEN }
    IF ( EW = 0 ) OR ( DIVISOR[EW] <> DIVISOR[EW-1] ) THEN BEGIN
      FOR ST := 0 TO M * DIVI - 1 DO BEGIN
        T := H * ST;
        TRVECP[ST] := TAYKFCPT( T , FCP_AKT );
        FOR J := 0 TO N DO
          TRMATQ[ J , ST ] := DIVI_2 * TAYKFCPT( T , FC_Q[J] );
        END;
      END;
      { EIGENWERTE VON FCP MIT PROZEDUR EW_APP IM MODUL DEPT BERECHNEN }
      EIG_AKT[EW] := EW_APP( M , P , EW , DIVI , TRVECP , EIG_ZIEL[EW] );
      { BERECHNUNG DER FUNKTIONALMATRIX }
      AWA_NWT( M * DIVI , P , H , EIG_AKT[EW] / DIVI_2 ,
        TRVECP , TRMATQ , VRU , VR_U2 , VR_U2Q );
      FOR I := 0 TO N DO
        FUNKT_MAT[EW,I] := VR_U2Q[I] / VR_U2;
      END; { FOR , EW }
      { NAECHSTE ITERIERTE DES POTENTIALS BERECHNEN }
      LSS( FUNKT_MAT , EIG_ZIEL - EIG_AKT , ZUWACHS , ERR );
      EIG_AKT := EIG_ZIEL;
      ZUWACHS_MID := MID( ZUWACHS );
      FCP_AKT := FCP_AKT + ZUWACHS_MID;
      { HILFSGROESSEN FUER DEN EINSCHLIESSUNGSSCHRITT BERECHNEN }
      ETA_ALT := ETA;
      ETA := MAX_NORM( ZUWACHS_MID );
      UNTIL 8 * ( ETA**3 ) / ( MAX_NORM( FCP_AKT ) * SQR( ETA_ALT ) ) < 1.0E-14;
    }
  }
  { SCHRITT 3 : BERECHNUNG EINES EINSCHLIESSENDEN INTERVALLPOTENTIALS
    ANFANGSWERTPROBLEME MIT INTERVALLEINSCHLIESSUNG GELOEST
  }
  -----
  FOR I := 0 TO N DO
    IFCP_MID[I] := FCP_AKT[I];
  { LETZTE FUNKTIONALMATRIX BESETZEN }
  FOR J := 0 TO N DO
    FOR I := 0 TO N DO
      ILAST_FUNKT_MAT[I,J] := FUNKT_MAT[I,J];
    }

```

```

{ BASISPOTENTIALE BERECHNEN }
FOR I := 0 TO N DO BEGIN
  FOR J := 0 TO N DO
    IFC_Q[I,J] := 0;
  IFC_Q[I,I] := 1;
END;
COUNT := 0;
REPEAT { "ENDLOS"-SCHLEIFE }
  COUNT := COUNT + 1;
  FOR EW := 0 TO N DO BEGIN
    DIVI := DIVISOR[EW];
    DIVI_2 := SQR( DIVI );
    { TAYLORKOEFFIZIENTEN DER POTENTIALFUNKTION BEREITSTELLEN }
    IF ( EW = 0 ) OR ( DIVISOR[EW] <> DIVISOR[EW-1] ) THEN BEGIN
      FOR ST := 0 TO M * DIVI - 1 DO BEGIN
        IT := IH * ST;
        TIVECP_MID[ST] := TAYKFCPT( IT , IFCP_MID );
        TIVECP_RST_MID[ST] := TAYKFCPT( IT + IHI , IFCP_MID );
      END;
    END;
    { EIGENWERTE VON MID(IFCP) BERECHNEN }
    { PROZEDUR EW_BDS IM MODUL DEPT BERECHNET EIGENWERTSCHRANKEN }
    { GENAUIGKEITS-PARAMETER 0.98 = HOHE GENAUIGKEIT }
    IEIG_AKT_MID[EW] := EIG_ZIEL[EW];
    IEIG_AKT_MID[EW] := EW_BDS( M , P , EW , DIVI , TIVECP_MID ,
      TIVECP_RST_MID , IEIG_AKT_MID[EW] , 0.98 );
  END; { FOR , EW }
  { INTERVALLPOTENTIAL GENERIEREN, IN DEM DIE LOESUNG VERMUTET WIRD }
  { VERFAHREN VON ALEFELD-GIENGER-POTRA VERWENDEN }
  { ZUM VERGLEICH ZUERST INNERE MINDESTEINSCHLIESSUNG BERECHNEN }
  LSS( ILAST_FUNKT_MAT , EIG_ZIEL - IEIG_AKT_MID , ZUWACHS , ERR );
  WIDTH_AKT := ZUWACHS[0].SUP -> ZUWACHS[0].INF;
  FOR I := 1 TO N DO
    IF WIDTH_AKT < ZUWACHS[I].SUP -> ZUWACHS[I].INF THEN
      WIDTH_AKT := ZUWACHS[I].SUP -> ZUWACHS[I].INF;
  IF WIDTH_AKT > 2 * ETA THEN BEGIN
    ETA := SQR( ETA * ETA_ALT );
    IF WIDTH_AKT > 2 * ETA THEN
      ETA := 50 * WIDTH_AKT; {100-FACHE MINDESTEINSCHLIESSUNG}
  END;
  { BEI DER EINSCHLIESSUNG DER EIGENFUNKTIONEN IST DIE KONSTANTE }
  { BASISFUNKTION q_0(x) = 1 UNERHEBLICH! }
  IFCP_AKT[0] := IFCP_MID[0];
  FOR I := 1 TO N DO
    IFCP_AKT[I] := IFCP_MID[I] + ETA * INTVAL( -1 , 1 );
  WIDTH_AKT := 2 * ETA;
  { EIGENWERTSCHRANKEN DES INTERVALLPOTENTIALS MIT DEM MONOTONIE- }
  { PRINZIP AUS DEN SCHRANKEN DES MITTELPUNKTPOTENTIALS BERECHNEN }
  { MAXIMALEN RADIUS DURCH MAXIMAL MOEGLICHEN RADIUS ABSCHAETZEN }
  FOR EW := 0 TO N DO BEGIN
    IEIG_AKT[EW] := IEIG_AKT_MID[EW] + INTVAL( -N * ETA , N * ETA );
    DIVI := DIVISOR[EW];
    DIVI_2 := SQR( DIVI );
    { TAYLORKOEFFIZIENTEN DER POTENTIALFUNKTION BEREITSTELLEN }
    IF ( EW = 0 ) OR ( DIVISOR[EW] <> DIVISOR[EW-1] ) THEN BEGIN
      FOR ST := 0 TO M * DIVI - 1 DO BEGIN
        IT := IH * ST;
        TIVECP[ST] := TAYKFCPT( IT , IFCP_AKT );
        TIVECP_RST[ST] := TAYKFCPT( IT+IHI , IFCP_AKT );
        FOR J := 0 TO N DO BEGIN
          TIMATQ[ J , ST ] := DIVI_2 * TAYKFCPT( IT , IFC_Q[J] );
          TIMATQ[ J , ST , P ] := RESTGLIED( IT + IHI , IFC_Q[J] );
        END;
      END;
    END;
  END;
  END;
  { BERECHNUNG DER FUNKTIONALMATRIX }

```

```

    AWA_NWT( M * DIVI , P , IH , IEIG_AKT[EW] / DIVI_2 ,
            TIVECP , TIVECP_RST , TIMATQ , VIU , VI_U2 , VI_U2Q );
    FOR I := 0 TO N DO
        IFUNKT_MAT[EW,I] := VI_U2Q[I] / VI_U2;
    END; { FOR , EW }
    { NAECHSTE ITERIERTE DES POTENTIALS BERECHNEN }
    { ZUERST IFCP_AKT[0] EBENFALLS AUFBLAEHEN }
    IFCP_AKT[0] := IFCP_MID[0] + ETA * INTVAL( -1 , 1 );
    LSS( IFUNKT_MAT , EIG_ZIEL - IEIG_AKT_MID , ZUWACHS , ERR );
    IFCP_ALT := IFCP_AKT;
    IFCP_AKT := IFCP_MID + ZUWACHS;
    NCL := TRUE;
    FOR I := 0 TO N DO
        IF NOT ( IFCP_AKT[I] IN IFCP_ALT[I] ) THEN NCL := FALSE;
    UNTIL NCL;
END.

```

Modul DEPT:

```

{-----}
{ PASCAL-XSC - MODUL DEPT }
{-----}
MODULE dept;
USE i_ari , mv_ari , mvi_ari;
USE iep_rawa , iep_iawa , iep_tayk , seig_app , service;
{-----}
{ PASCAL-XSC - PROZEDUR DEPT_RITZ }
{ BERECHNET FUER EINE ENDLICHE COSINUSREIHE NAEHERUNGEN FUER DIE ERSTEN }
{ N+1 EIGENWERTE MIT DEM RITZ-VERFAHREN ZU DEN ANSATZFUNKTIONEN SIN(kx) }
{ SOWIE DIE FUNKTIONALMATRIX BEIM NEWTON-VERFAHREN FUER DAS INVERSE }
{ DIRICHLETPROBLEM AUS DER DISSERTATION }
{ DIE KOEFFIZIENTEN DER COSINUSREIHE STEHEN IN FCP_AKT }
{-----}
GLOBAL PROCEDURE DEPT_RITZ( FCP_AKT : RVECTOR; N , ANZ : INTEGER;
                           VAR EIGVAL : RVECTOR; VAR FUNKT_MAT : RMATRIX );
{
  { AUSGEGEBEN WERDEN DIE KLEINSTE N+1 EIGENWERTE IN EIGVAL UND DIE }
  { FUNKTIONALMATRIX BEIM NEWTON-VERFAHREN ZUM INVERSEN DP IN FUNKT_MAT }
  { N : N + 1 = ANZAHL DER VORGEgebenEN EIGENWERTE VON P(X) }
  { ANZ = ANZAHL DER ANSATZFUNKTIONEN BEIM RITZ-VERFAHREN }
  { (JE FUER GERADE UND UNGERADE EIGENWERTE) }
}
CONST PI = 3.141592653589793;
TYPE MATSARRAY = DYNAMIC ARRAY[*,*,*] OF REAL;
VAR I , J , K : INTEGER;
    H1 : REAL;
    VALID : BOOLEAN;
    RITZEIGVAL , VR1 : RVECTOR[1..ANZ];
    RITZMAT , RITZEIGVEC : RMATRIX[1..ANZ,1..ANZ];
    BASIS_RITZMATS : MATSARRAY[0..N,1..ANZ,1..ANZ];
    FILENAMEG , FILENAMEU : STRING;
    FILEG , FILEU : TEXT;
BEGIN
  { ERSTER TEILSCHRITT FUER DIE EIGENWERTE VON P(X) }
  { MIT GERADEM INDEX: BASIS-RITZ-MATRIZEN ERSTELLEN }
  BASIS_RITZMATS[0,*,*] := ID( ANZ );
  FOR K := 1 TO N DO BEGIN
    BASIS_RITZMATS[K,*,*] := NULL( ANZ );
    FOR I := K TO K DO
      FOR J := I + 1 TO ANZ DO BEGIN
        BASIS_RITZMATS[K,J,J-I] := 0.5;

```

```

        BASIS_RITZMATS[K,J-I,J] := 0.5;
    END;
    FOR I := K TO K DO
        FOR J := 1 TO I DO
            BASIS_RITZMATS[K,I+1-J,J] := BASIS_RITZMATS[K,I+1-J,J] - 0.5;
        END;
    END;
{ ZWEITER TEILSCHRITT FUER DIE EIGENWERTE VON P(X) }
{ MIT GERADEM INDEX : RITZ-MATRIX BERECHNEN }
RITZMAT := FCP_AKT[0] * BASIS_RITZMATS[0,*,*];
FOR I := 1 TO N DO
    RITZMAT := RITZMAT + FCP_AKT[I] * BASIS_RITZMATS[I,*,*];
{ DIAGONALTERM " - U_J'' U_I " ADDIEREN }
FOR I := 1 TO 30 DO
    RITZMAT[I,I] := RITZMAT[I,I] + SQR( 2 * I - 1 );
{ DRITTER TEILSCHRITT FUER DIE EIGENWERTE VON P(X) MIT }
{ GERADEM INDEX : MODUL SEIG_APP ZUR EIGENWERTBERECHNUNG AUFRUFEN }
SEIG_APP( RITZMAT , RITZEIGVAL , RITZEIGVEC );
{ KLEINSTE N DIV 2 + 1 EIGENWERTE SORTIEREN }
K := 0;
REPEAT VALID := TRUE;
    K := K + 1;
    FOR I := ANZ DOWNT0 K + 1 DO
        IF RITZEIGVAL[I] < RITZEIGVAL[I-1] THEN BEGIN
            VALID := FALSE;
            H1 := RITZEIGVAL[I];
            VR1 := RITZEIGVEC[*,I];
            RITZEIGVAL[I] := RITZEIGVAL[I-1];
            RITZEIGVEC[*,I] := RITZEIGVEC[*,I-1];
            RITZEIGVAL[I-1] := H1;
            RITZEIGVEC[*,I-1] := VR1;
        END;
    UNTIL VALID OR ( K = N DIV 2 + 1 );
    FOR I := 0 TO N DIV 2 DO
        EIGVAL[2*I] := RITZEIGVAL[I+1];
    { FUNKTIONALMATRIX BERECHNEN }
    FOR I := 0 TO N DIV 2 DO
        FOR J := 0 TO N DO
            FUNKT_MAT[2*I,J] := RITZEIGVEC[*,I+1] *
                ( BASIS_RITZMATS[J,*,*] * RITZEIGVEC[*,I+1] );
        { ERSTER TEILSCHRITT FUER DIE EIGENWERTE VON P(X) }
        { MIT UNGERADEM INDEX: BASIS-RITZ-MATRIZEN ERSTELLEN }
        BASIS_RITZMATS[0,*,*] := ID( ANZ );
        FOR K := 1 TO N DO BEGIN
            BASIS_RITZMATS[K,*,*] := NULL( ANZ );
            FOR I := K TO K DO
                FOR J := I + 1 TO ANZ DO BEGIN
                    BASIS_RITZMATS[K,J,J-I] := 0.5;
                    BASIS_RITZMATS[K,J-I,J] := 0.5;
                END;
            END;
            FOR I := K TO K DO
                FOR J := 1 TO I - 1 DO
                    BASIS_RITZMATS[K,I-J,J] := BASIS_RITZMATS[K,I-J,J] - 0.5;
                END;
            END;
{ ZWEITER TEILSCHRITT FUER DIE EIGENWERTE VON P(X) }
{ MIT UNGERADEM INDEX : RITZ-MATRIX BERECHNEN }
RITZMAT := FCP_AKT[0] * BASIS_RITZMATS[0,*,*];
FOR I := 1 TO N DO
    RITZMAT := RITZMAT + FCP_AKT[I] * BASIS_RITZMATS[I,*,*];
{ DIAGONALTERM " - U_J'' U_I " ADDIEREN }
FOR I := 1 TO 30 DO
    RITZMAT[I,I] := RITZMAT[I,I] + SQR( 2 * I );
{ DRITTER TEILSCHRITT FUER DIE EIGENWERTE VON P(X) MIT }
{ UNGERADEM INDEX : MODUL SEIG_APP ZUR EIGENWERTBERECHNUNG AUFRUFEN }
SEIG_APP( RITZMAT , RITZEIGVAL , RITZEIGVEC );
{ KLEINSTE ( N + 1 ) DIV 2 EIGENWERTE SORTIEREN }
K := 0;
REPEAT VALID := TRUE;

```

```

      K := K + 1;
      FOR I := ANZ DOWNT0 K + 1 DO
        IF RITZEIGVAL[I] < RITZEIGVAL[I-1] THEN BEGIN
          VALID := FALSE;
          H1 := RITZEIGVAL[I];
          VR1 := RITZEIGVEC[* , I];
          RITZEIGVAL[I] := RITZEIGVAL[I-1];
          RITZEIGVEC[* , I] := RITZEIGVEC[* , I-1];
          RITZEIGVAL[I-1] := H1;
          RITZEIGVEC[* , I-1] := VR1;
        END;
      UNTIL VALID OR ( K = ( N + 1 ) DIV 2 );
      FOR I := 1 TO ( N + 1 ) DIV 2 DO
        EIGVAL[2*I-1] := RITZEIGVAL[I];
      { FUNKTIONALMATRIX BERECHNEN }
      FOR I := 1 TO ( N + 1 ) DIV 2 DO
        FOR J := 0 TO N DO
          FUNKT_MAT[2*I-1, J] := RITZEIGVEC[* , I] *
            ( BASIS_RITZMATS[J , * , *] * RITZEIGVEC[* , I] );
        END;
      END;
{-----}
{ PASCAL-XSC - PROZEDUR EW_APP }
{ BERECHNET FUER DIE FUNKTION TRVECP NAEHERUNGEN FUER EINEN EIGENWERT }
{ IN DER NAEHE DES EINGABEPARAMETERS EIG MIT DEM SEKANTENVERFAHREN }
{-----}
GLOBAL FUNCTION EW_APP( M , P , EW , DIVI : INTEGER; TRVECP : RTVEC;
  EIG : REAL ) : REAL;
  { P : ORDNUNG DER TAYLORENTWICKLUNG VON U }
  { M : SCHRITTWEITE H = PI / M }
  { EW : NUMMER DES EIGENWERTS }
  { TRVECP ENTHAELT DIE TAYLORKOEFFIZIENTEN DER FUNKTION }
  { IN DEN STUETZSTELLEN (GEEIGNET NORMIERT) }
VAR DIVI_2 , COUNT : INTEGER;
  EIGALT , STEIGUNG , PI , H : REAL;
  VRU , VRV : RVECTOR[1..2];
BEGIN
  { GLOBALE SCHRITTWEITE SETZEN }
  PI := 3.141592653589793;
  H := PI / M;
  DIVI_2 := SQR( DIVI );
  { ZUM EINSTIEG IN DAS SEKANTENVEFAHREN }
  { FUNKTIONSWERTE IN DER NAEHE VON EIG BERECHNEN }
  { EIGALT ENTHAELT DEN WERT DES EIGENWERTS IM "LETZTEN" SCHRITT }
  COUNT := 0;
  EIGALT := 1.000001 * EIG;
  { ZUM START DES SEKANTENVERFAHRENS WIRD DER FUNKTIONSWERT }
  { AN DER STELLE EIGALT (IN DER NAEHE VON EIG BERECHNET) }
  VRU := AWA( M * DIVI , P , H , EIGALT / DIVI_2 , TRVECP );
  { VRV ENTHAELT DEN "FUNKTIONSWERT" DES "LETZTEN" SCHRITTS }
  VRV := VRU;
  { SEKANTENVERFAHREN }
  REPEAT
    COUNT := COUNT + 1;
    VRU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TRVECP );
    STEIGUNG := ( VRU[1] - VRV[1] ) / ( EIG - EIGALT );
    EIGALT := EIG;
    IF STEIGUNG <> 0 THEN
      EIG := EIG - VRU[1] / STEIGUNG;
    VRV := VRU;
  UNTIL ( ( ABS( EIG - EIGALT ) < 3.0E-12 * ABS( EIGALT ) )
    OR ( STEIGUNG = 0 ) );
  VRU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TRVECP );
  EW_APP := EIG;

```

```

END;
{-----}
{ PASCAL-XSC - PROZEDUR EW_BDS }
{ BERECHNET FUER DIE FUNKTION TRVECP SCHRANKEN FUER EINEN EIGENWERT }
{-----}

GLOBAL FUNCTION EW_BDS( M, P, EW, DIVI : INTEGER; TIVECP, TIVECP_RST : ITVEC;
                      EIG_APP : INTERVAL; PREC : REAL ) : INTERVAL;
  { P : ORDNUNG DER TAYLORENTWICKLUNG VON U }
  { M : SCHRITTWEITE H = PI / M }
  { EW : NUMMER DES EIGENWERTS }
  { PREC : GENAUIGKEITSANFORDERUNGEN IN % }
  { TIVECP ENTHAELT DIE TAYLORKOEFFIZIENTEN DER FUNKTION }
  { IN DEN STUETZSTELLEN (GEEIGNET NORMIERT) }

VAR COUNT , I : INTEGER;
    EIG , STEIGUNG , INCREASE : REAL;
    EIGLOW , EIGHIGH , FCULOW , FCUHIGH : INTERVAL;
    PI , H , DIVI_2 , EIGBDS : INTERVAL;
    VIU : IVECTOR[1..2];

PROCEDURE IST_USCHRANKE;
BEGIN EIGLOW.INF := EIG; FCULOW.INF := VIU[1].INF; END;

PROCEDURE KEINE_USCHRANKE;
BEGIN EIGLOW.SUP := EIG; FCULOW.SUP := VIU[1].INF; END;

PROCEDURE IST_OSCHRANKE;
BEGIN EIGHIGH.SUP := EIG; FCUHIGH.SUP := VIU[1].SUP; END;

PROCEDURE KEINE_OSCHRANKE;
BEGIN EIGHIGH.INF := EIG; FCUHIGH.INF := VIU[1].SUP; END;

FUNCTION VALID : BOOLEAN;
{ ABRUCHBEDINGUNG BEI DER ITERATIONSSCHLEIFE }
BEGIN VALID := TRUE;
  IF ( ( PREC > 0 ) AND ( PREC < 1 ) AND
      ( EIGHIGH.INF -> EIGLOW.SUP > PREC * ( EIGHIGH.SUP -< EIGLOW.INF ) ) )
  THEN VALID := FALSE;
  { BEI GANZ ENGER EINSCHLIESSUNG ZUSAETZLICHES KRITERIUM NOETIG }
  IF ( EIGLOW.INF +> 1.0E-99 >= EIGLOW.SUP -< 1.0E-99 ) AND
      ( EIGHIGH.INF +> 1.0E-99 >= EIGHIGH.SUP -< 1.0E-99 ) THEN
    VALID := FALSE;
END;

{--- HAUPTPROGRAMM EW_BDS -----}
BEGIN
  { FUER DEN EIGENWERT WERDEN MIT DEM BISEKTIONSVERFAHREN }
  { ERST EINE UNTER-, DANN EINE OBERSCHRANKE BERECHNET }
  { DA DIE "FUNKTIONSWERTE" DER "EIGENFUNKTIONEN" WEGEN DEN }
  { AUFTRETENDEN RUNDUNGSFEHLERN UNSTETIG VON DEN EIGENWERT- }
  { NAEHERUNGEN ABHAENGEN, IST DAS SEKANTENVERFAHREN HIER }
  { NICHT BRAUCHBAR --> REGULA FALSI }
  { ZUR BERECHNUNG EINER GROBEINSCHLIESSUNG IST JEWEILS }
  { EINE VORITERATION IN DER NAEHE DES EIGENWERTS NOETIG }
  { ES WIRD VORAUSGESETZT, DASS EINE GUTE NAEHERUNG DES }
  { EIGENWERTS EINGELESEN WIRD }
  PI:= INTVAL( RVAL('3.141592653589793',-1) , RVAL('3.141592653589794',1) );
  H := PI / M;
  DIVI_2 := SQR( DIVI );
  EIG := MID( EIG_APP );
  FCULOW := 0;
  FCUHIGH := 0;
  COUNT := 1;
  { SCHRITTWEITE BEI DER VORITERATION MINDESTENS }
  { HALBER DURCHMESSER DER EIGENWERTANFANGSNAEHERUNG }
  INCREASE := ( 1 + ABS( EIG ) ) * 2.0E-14;
  INCREASE := INCREASE + ( EIG_APP.SUP - EIG_APP.INF ) / 2;
  { ZUM EINSTIEG IN DAS BISEKTIONSVERFAHREN }

```

```

{ FUNKTIONSWERTE FUER EIG BERECHNEN }
VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
{ ZUERST GROBEINSCHLIESSUNG DER UNTERSCHRANKE BERECHNEN }
IF VIU[1].INF >= 0 THEN BEGIN
  IST_USCHRANKE;
  KEINE_USCHRANKE;
  { JETZT IST EINE NICHT MEHR GARANTIERBARE (ZU GROSSE) }
  { UNTERSCHRANKE ZU BERECHNEN }
  REPEAT
    { HEURISTISCHE ERHOEHUNG UM DEN DURCHMESSER DER FUNKTION }
    { UND DEN WERT ABS( U(pi) ) / ABS( VIU[2].SUP ) }
    COUNT := COUNT + 1;
    EIG := EIG + INCREASE + VIU[1].INF / ABS( VIU[2].SUP );
    INCREASE := 2 * INCREASE;
    VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
    VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
    IF VIU[1].INF >= 0 THEN BEGIN
      IST_USCHRANKE;
      KEINE_USCHRANKE;
    END
  ELSE BEGIN
    KEINE_USCHRANKE;
    IF VIU[1].SUP <= 0 THEN IST_OSCHRANKE
    ELSE KEINE_OSCHRANKE;
  END;
  UNTIL VIU[1].INF < 0;
END
ELSE BEGIN
  KEINE_USCHRANKE;
  IF VIU[1].SUP <= 0 THEN IST_OSCHRANKE
  ELSE KEINE_OSCHRANKE;
  REPEAT
    { HEURISTISCHE VERMINDERUNG UM DEN DURCHMESSER DER FUNKTION }
    { UND DEN WERT ABS( U(pi) ) / ABS( VIU[2].SUP ) }
    COUNT := COUNT + 1;
    EIG := EIG - INCREASE + VIU[1].INF / ABS( VIU[2].SUP );
    INCREASE := 2 * INCREASE;
    VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
    VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
    IF VIU[1].INF >= 0 THEN BEGIN
      IST_USCHRANKE;
      IF FCUHIGH.INF = 0 THEN KEINE_OSCHRANKE;
    END
  ELSE BEGIN
    KEINE_USCHRANKE;
    IF VIU[1].SUP <= 0 THEN IST_OSCHRANKE
    ELSE IF FCUHIGH.INF = 0 THEN KEINE_OSCHRANKE;
  END;
  UNTIL VIU[1].INF >= 0;
END;
{ EVENTUELL MUSS NOCH EINE GARANTIERTE }
{ OBERSCHRANKE BERECHNET WERDEN }
IF FCUHIGH.SUP = 0 THEN BEGIN
  { STARTWERTE VORBESETZEN }
  INCREASE := ( 1 + ABS( EIG ) ) * 2.0E-14;
  INCREASE := INCREASE + ( EIG_APP.SUP - EIG_APP.INF ) / 2;
  EIG := EIGHIGH.INF;
  VIU[1].SUP := FCUHIGH.INF;
  VIU[2].SUP := 1.0E+14 * VIU[1].SUP; { DUMMY-WERT }
  { JETZT IST EINE GARANTIERTE OBERSCHRANKE ZU BERECHNEN }
  REPEAT
    { HEURISTISCHE ERHOEHUNG UM DEN DURCHMESSER DER FUNKTION }
    { UND DEN WERT ABS( U(pi) ) / ABS( VIU[2].SUP ) }
    COUNT := COUNT + 1;
    EIG := EIG + INCREASE + VIU[1].SUP / ABS( VIU[2].SUP );

```

```

INCREASE := 2 * INCREASE;
VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
IF VIU[1].SUP <= 0 THEN IST_OSCHRANKE
ELSE KEINE_OSCHRANKE;
UNTIL VIU[1].SUP <= 0;
END;
{ SOLANGE DIE EINSCHLIESSUNGEN DER UNTER- UND DER OBERSCHRANKE }
{ UEBEREINSTIMMEN, KANN SIMULTAN MIT DER REGULA FALSI ITERIERT WERDEN }
WHILE EIGHIGH = EIGLOW DO BEGIN
COUNT := COUNT + 1;
STEIGUNG:= ( FCUHIGH.SUP-FCULOW.INF ) / ( EIGHIGH.SUP-EIGLOW.INF );
{ STEIGUNG IST <> 0 , WEIL FCUHIGH.SUP * FCULOW.INF <> 0 }
EIG := EIGLOW.INF - FCULOW.INF / STEIGUNG;
{ HEURISTISCHE KORREKTUR BEI ALLZU UNSYMMETRISCHER TEILUNG }
IF ( EIG - EIGLOW.INF ) < ( EIGHIGH.SUP - EIGLOW.INF ) / 20 THEN
EIG := EIGLOW.INF -> 2 * FCULOW.INF /> STEIGUNG;
IF ( EIGHIGH.SUP - EIG ) < ( EIGHIGH.SUP - EIGLOW.INF ) / 20 THEN
EIG := 2 * EIG -< EIGHIGH.SUP;
VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
IF VIU[1].INF >= 0 THEN BEGIN
IST_USCHRANKE;
KEINE_OSCHRANKE;
END
ELSE IF VIU[1].SUP <= 0 THEN BEGIN
IST_OSCHRANKE;
KEINE_USCHRANKE;
END
ELSE BEGIN
KEINE_USCHRANKE;
KEINE_OSCHRANKE;
END;
END; { WHILE }
{ DIE GROBEINSCHLIESSUNGEN EIGLOW UND EIGHIGH SIND JETZT }
{ MIT DER REGULA FALSI SEPARAT ZU VERBESSERN }
COUNT := 1;
WHILE VALID DO BEGIN
IF EIGLOW.SUP - EIGLOW.INF >= EIGHIGH.SUP - EIGHIGH.INF THEN BEGIN
{ UNTERSCHRANKE VERBESSERN }
{ IN DER BEDINGUNG STEHT ">=" , DAMIT BEI GLEICHHEIT }
{ KEINE ENDLOSSCHLEIFE DURCHLAUFEN WIRD }
COUNT := COUNT + 1;
STEIGUNG:= ( FCULOW.SUP-FCULOW.INF ) / ( EIGLOW.SUP-EIGLOW.INF );
{ STEIGUNG IST <> 0 , WEIL FCULOW.SUP * FCULOW.INF <> 0 }
EIG := EIGLOW.INF - FCULOW.INF / STEIGUNG;
{ HEURISTISCHE KORREKTUR BEI ALLZU UNSYMMETRISCHER TEILUNG }
IF ( EIG - EIGLOW.INF ) < ( EIGLOW.SUP - EIGLOW.INF ) / 20 THEN
EIG := EIGLOW.INF - 2 * FCULOW.INF / STEIGUNG;
IF ( EIGLOW.SUP - EIG ) < ( EIGLOW.SUP - EIGLOW.INF ) / 20 THEN
EIG := 2 * EIG - EIGLOW.SUP;
IF EIG <= EIGLOW.INF THEN
EIG := EIGLOW.INF +> ( EIGLOW.SUP - EIGLOW.INF ) /> 4;
IF EIG >= EIGLOW.SUP THEN
EIG := EIGLOW.SUP -< ( EIGLOW.SUP - EIGLOW.INF ) /> 4;
VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
IF VIU[1].INF >= 0 THEN IST_USCHRANKE
ELSE KEINE_USCHRANKE;
END; { IF }
IF ( VALID AND ( EIGHIGH.SUP-EIGHIGH.INF >= EIGLOW.SUP-EIGLOW.INF ) )
THEN BEGIN
{ OBERSCHRANKE VERBESSERN }
COUNT := COUNT + 1;
STEIGUNG:= (FCUHIGH.SUP-FCUHIGH.INF) / (EIGHIGH.SUP-EIGHIGH.INF);
{ STEIGUNG IST <> 0 , WEIL FCUHIGH.SUP * FCUHIGH.INF <> 0 }

```

```

EIG := EIGHIGH.INF - FCUHIGH.INF / STEIGUNG;
{ HEURISTISCHE KORREKTUR BEI ALLZU UNSYMMETRISCHER TEILUNG }
IF ( EIG - EIGHIGH.INF ) < ( EIGHIGH.SUP - EIGHIGH.INF ) / 20 THEN
  EIG := EIGHIGH.INF - 2 * FCUHIGH.INF / STEIGUNG;
IF ( EIGHIGH.SUP - EIG ) < ( EIGHIGH.SUP - EIGHIGH.INF ) / 20 THEN
  EIG := 2 * EIG - EIGHIGH.SUP;
IF EIG <= EIGHIGH.INF THEN
  EIG := EIGHIGH.INF +> ( EIGHIGH.SUP - EIGHIGH.INF ) /> 4;
IF EIG >= EIGHIGH.SUP THEN
  EIG := EIGHIGH.SUP -< ( EIGHIGH.SUP - EIGHIGH.INF ) /> 4;
VIU := AWA( M * DIVI , P , H , EIG / DIVI_2 , TIVECP , TIVECP_RST );
VIU := VIU * ( 1 - 2 * ( EW MOD 2 ) );
IF VIU[1].SUP <= 0 THEN IST_OSCHRANKE
ELSE KEINE_OSCHRANKE;
END; { IF }
END; { WHILE }
EIGBDS.INF := EIGLOW.INF;
EIGBDS.SUP := EIGHIGH.SUP;
EW_BDS := EIGBDS;
END;
END.

```

Modul IEP_IAWA:

```

{-----}
{ PASCAL-XSC - MODUL IEP_IAWA }
{-----}
MODULE iep_iawa;
USE i_ari , mv_ari , mvi_ari , iep_tayk , service;
FUNCTION MIINV2X2( MRA : RMATRIX ) : IMATRIX[1..2,1..2];
{ INVERSION EINER 2x2-MATRIX }
VAR IDET : INTERVAL;
    MIB : IMATRIX[1..2,1..2];
BEGIN MIB[1,1] := MRA[2,2];
    MIB[1,2] := - MRA[1,2];
    MIB[2,1] := - MRA[2,1];
    MIB[2,2] := MRA[1,1];
    IDET := MIB[1,1] * MIB[2,2] - MIB[1,2] * MIB[2,1];
    MIB[1,1] := MIB[1,1] / IDET;
    MIB[1,2] := MIB[1,2] / IDET;
    MIB[2,1] := MIB[2,1] / IDET;
    MIB[2,2] := MIB[2,2] / IDET;
    MIINV2X2 := MIB;
END;
FUNCTION QR2X2( MRA : RMATRIX ) : RMATRIX[1..2,1..2];
{ QR - ZERLEGUNG EINER 2X2 - MATRIX }
VAR A , B , R1 , R2 , R3 : REAL;
    MRQ : RMATRIX[1..2,1..2];
BEGIN A := MRA[1,1];
    B := MRA[2,1];
    R1 := SQRT( SQR( A ) + SQR( B ) );
    IF A > 0 THEN
      R2 := A + R1
    ELSE
      R2 := A - R1;
    R3 := 2 / ( SQR( R2 ) + SQR( B ) );
    MRQ[1,1] := 1 - R3 * SQR( R2 );
    MRQ[1,2] := - B * R2 * R3;

```

```

MRQ[2,1] := MRQ[1,2];
MRQ[2,2] := 1 - SQR( B ) * R3;
QR2X2 := MRQ;
END;

FUNCTION PERM2X2( MRA : RMATRIX ; VIX : IVECTOR ) : RMATRIX[1..2,1..2];
{ PERMUTATION EINER 2X2 - MATRIX, FALLS NEBENBEDINGUNGEN ERFUELLT SIND }
VAR R1 , R2 : REAL;
    VRY      : RVECTOR[1..2];
    MRB      : RMATRIX[1..2,1..2];

BEGIN R1 := SQR( VIX[1].SUP -> VIX[1].INF );
R2 := SQR( VIX[2].SUP -> VIX[2].INF );
VRY[1] := ( SQR( MRA[1,1] ) + SQR( MRA[2,1] ) ) * R1;
VRY[2] := ( SQR( MRA[1,2] ) + SQR( MRA[2,2] ) ) * R2;
IF VRY[2] <= VRY[1] THEN
    { KEIN SPALTENTAUSSCH }
    MRB := MRA
ELSE BEGIN
    { TAUSCHE ERSTE UND ZWEITE SPALTEN VON MRA }
    MRB[1,1] := MRA[1,2];
    MRB[2,1] := MRA[2,2];
    MRB[1,2] := MRA[1,1];
    MRB[2,2] := MRA[2,1];
END;
PERM2X2 := MRB;
END;

FUNCTION TAYKOEFF( TIP: ITKO; VIU: IVECTOR ) : ITVEC[1..2,0..UB(TIP)];
VAR NR , P : INTEGER;
    TIVECU : ITVEC[1..2,0..UB(TIP)];

BEGIN P := UB( TIP );
TIVECU[1] := TKO( P , VIU[1] );
TIVECU[2] := TKO( P , VIU[2] );
FOR NR := 1 TO P DO BEGIN
    { U1' := U2 }
    TIVECU[1,NR] := TIVECU[2,NR-1] / NR;
    { U2' := FCP * U1 }
    TIVECU[2,NR] := TIPROD( TIP , TIVECU[1] , NR - 1 ) / NR;
END; { FOR }
TAYKOEFF := TIVECU;
END;

PROCEDURE PSI_ABLEITUNG( H : INTERVAL; TIP : ITKO;
                        VAR MI_AJ : IMATRIX; VAR TIVECU : ITVEC);
{ BERECHNET WIRD DIE MATRIX MI_AJ, SIEHE LOHNER SEITE 43 }
{ ALS NEBENPRODUKT ERHAELT MAN }
{ DIE TAYLORKOEFFIZIENTEN DER LOESUNG VIU SELBST MIT DAZU }

VAR I , J , NR , P : INTEGER;
    IU1 , IU2      : INTERVAL;
    TIMATU        : ITMAT[1..2,1..2,0..UB(TIP)];

BEGIN P := UB( TIP );
{ IU1, IU2 ENTHALTEN DEN FUNKTIONSWERT VON U AN DER STELLE TJ }
{ DIE MATRIX MI_AJ UND DIE LOESUNG U DER DGL ERFUELLEN DIE GLEICHE DGL }
{ MI_AJ HAT ANFANGSWERT ID, U(X) HAT ANFANGSWERT (IU1,IU2) }
IU1 := TIVECU[1,0];
IU2 := TIVECU[2,0];
TIMATU := TIMATID( P );
FOR NR := 1 TO P - 1 DO BEGIN
    { U11' := U21 }
    TIMATU[1,1,NR] := TIMATU[2,1,NR-1] / NR;
    { U12' := U22 }
    TIMATU[1,2,NR] := TIMATU[2,2,NR-1] / NR;
    { U21' := FCP * U11 }

```

```

TIMATU[2,1,NR] := TIPROD( TIP , TIMATU[1,1] , NR - 1 ) / NR;
{ U22' := FCP * U12 }
TIMATU[2,2,NR] := TIPROD( TIP , TIMATU[1,2] , NR - 1 ) / NR;
{ TAYLORKOEFFIZIENTEN VON VIU AUS TAYLORKOEFFIZIENTEN VON MI_AJ ! }
TIVECU[1,NR] := TIMATU[1,1,NR] * IU1 + TIMATU[1,2,NR] * IU2;
TIVECU[2,NR] := TIMATU[2,1,NR] * IU1 + TIMATU[2,2,NR] * IU2;
END; { FOR }
FOR I := 1 TO 2 DO
  FOR J := 1 TO 2 DO BEGIN
    MI_AJ[I,J] := TIMATU[I,J,P-1];
    FOR NR := P - 2 DOWNTO 0 DO
      MI_AJ[I,J] := MI_AJ[I,J] * H + TIMATU[I,J,NR];
    END; { FOR }
  END;
END;

PROCEDURE PSI_ABLEITUNG( H : INTERVAL; TIP : ITKO; VAR MI_AJ : IMATRIX;
  VAR TIVECU , TIVECUINCL : IVEC);
{ BERECHNET WIRD DIE MATRIX MI_AJ, SIEHE LOHNER SEITE 43 }
{ ALS NEBENPRODUKT ERHAELT MAN }
{ DIE TAYLORKOEFFIZIENTEN DER LOESUNG VIU SELBST MIT DAZU }
VAR I , J , NR , P : INTEGER;
    IU1 , IU2 , IU3 , IU4 : INTERVAL;
    TIMATU : ITMAT[1..2,1..2,0..UB(TIP)];
BEGIN P := UB( TIP );
  { IU1, IU2 ENTHALTEN EINE NAEHERUNG DES FUNKTIONSWERTS VON U AN DER }
  { STELLE TJ, IU3 UND IU4 EINE EINSCHLIESSUNG DAVON }
  { DIE MATRIX MI_AJ UND DIE LOESUNG U DER DGL ERFUELLEN DIE GLEICHE DGL }
  { MI_AJ HAT ANFANGSWERT ID, U(X) HAT ANFANGSWERT (IU1,IU2) !!! }
  IU1 := TIVECU[1,0];
  IU2 := TIVECU[2,0];
  IU3 := TIVECUINCL[1,0];
  IU4 := TIVECUINCL[2,0];
  TIMATU := TIMATID( P );
  FOR NR := 1 TO P - 1 DO BEGIN
    { U11' := U21 }
    TIMATU[1,1,NR] := TIMATU[2,1,NR-1] / NR;
    { U12' := U22 }
    TIMATU[1,2,NR] := TIMATU[2,2,NR-1] / NR;
    { U21' := FCP * U11 }
    TIMATU[2,1,NR] := TIPROD( TIP , TIMATU[1,1] , NR - 1 ) / NR;
    { U22' := FCP * U12 }
    TIMATU[2,2,NR] := TIPROD( TIP , TIMATU[1,2] , NR - 1 ) / NR;
    { TAYLORKOEFFIZIENTEN VON VIU AUS TAYLORKOEFFIZIENTEN VON MI_AJ }
    TIVECU[1,NR] := TIMATU[1,1,NR] * IU1 + TIMATU[1,2,NR] * IU2;
    TIVECU[2,NR] := TIMATU[2,1,NR] * IU1 + TIMATU[2,2,NR] * IU2;
    TIVECUINCL[1,NR] := TIMATU[1,1,NR] * IU3 + TIMATU[1,2,NR] * IU4;
    TIVECUINCL[2,NR] := TIMATU[2,1,NR] * IU3 + TIMATU[2,2,NR] * IU4;
  END; { FOR }
  FOR I := 1 TO 2 DO
    FOR J := 1 TO 2 DO BEGIN
      MI_AJ[I,J] := TIMATU[I,J,P-1];
      FOR NR := P - 2 DOWNTO 0 DO
        MI_AJ[I,J] := MI_AJ[I,J] * H + TIMATU[I,J,NR];
      END; { FOR }
    END;
  END;
END;

FUNCTION GROBEIN( HI , IP : INTERVAL; VIU : IVECTOR ) : IVECTOR[1..2];
VAR VIY , VIZ : IVECTOR[1..2];
BEGIN VIZ := VIU;
  REPEAT VIY := INTVAL( 0.9 , 1.1 ) * VIZ;
    VIZ[1] := VIU[1] + HI * VIY[2];
    VIZ[2] := VIU[2] + HI * IP * VIY[1];
  UNTIL VIZ <= VIY;
END;

```

```

    GROBEIN := VIZ;
END;
FUNCTION RESTGLIED ( H : INTERVAL; TIP: ITKO; VIU: IVECTOR ) : IVECTOR[1..2];
VAR P      : INTEGER;
    VIZ    : IVECTOR[1..2];
    TIVECU : ITVEC[1..2,0..UB(TIP)];
BEGIN P := UB( TIP );
    TIVECU := TAYKOEFF( TIP , VIU );
    VIZ[1] := TIVECU[1,P] * ( H ** P );
    VIZ[2] := TIVECU[2,P] * ( H ** P );
    RESTGLIED := VIZ;
END;
GLOBAL FUNCTION PUNKTAUSWERTUNG( H : INTERVAL; TIU : ITKO ) : INTERVAL;
VAR I , P : INTEGER;
    IX    : INTERVAL;
BEGIN P := UB( TIU );
    IX := TIU[P-1];
    FOR I := P - 2 DOWNT0 0 DO
        IX := H * IX + TIU[I];
    PUNKTAUSWERTUNG := IX;
END;
GLOBAL FUNCTION PUNKTAUSWERTUNG( H : INTERVAL; TIVECU : ITVEC ) :
    IVECTOR[ LB( TIVECU ) .. UB( TIVECU ) ];
VAR I : INTEGER;
    VIU : IVECTOR[ LB( TIVECU ) .. UB( TIVECU ) ];
BEGIN
    FOR I := LB( TIVECU ) TO UB( TIVECU ) DO
        VIU[I] := PUNKTAUSWERTUNG( H , TIVECU[I] );
    PUNKTAUSWERTUNG := VIU;
END;
{-----}
{ PASCAL-XSC - PROZEDUR AWA }
{ AWP-PROZEDUR FUER DIE STURM-LIOUVILLE-GLEICHUNG }
{ MIT UNENDLICH OFT DIFFERENZIERBAREN POTENTIALEN }
{-----}
GLOBAL FUNCTION AWA( UPBD , P : INTEGER; H , EIG : INTERVAL;
    TIVECP , TIVECP_RST : ITVEC ) : IVECTOR[1..2];
{ BEZEICHNUNGEN WIE BEI LOHNER }
{ SEITENANGABEN BEZIEHEN SICH AUF DESSEN DISSERTATION }
{ UPBD : ANZAHL DER STUETZSTELLEN IM INTERVALL [0,pi] }
{ P : ORDNUNG DER TAYLOR-ENTWICKLUNG }
{ H , HI : SCHRITTWEITE }
{ EIG : EIGENWERT }
{ TIVECP , TIVECP_RST : TAYLORKOEFFIZIENTEN DES POTENTIALS P(X) }
{ VIU : GESAMTEINSCHLIESSUNG }
{ VIU_GROB : GROBEINSCHLIESSUNGSSCHLANGE SEITE 40 }
{ VIU_SCHLANGE : INTERVALLGROESSE UJSCHLANGE SEITE 54 }
{ VIU_MID : PUNKTGROESSE UJSCHLANGE SEITE 54 }
{ VI_ZJPLUS1 : DISKRETISIERUNGSFEHLER ZJ+1 SEITE 48 }
{ VI_ZJPLUS1_SCHLANGE : " ZJ+1_SCHLANGE SEITE 54 }
{ (ENTHAELT AUCH DIE RUNDUNGSFEHLER) }
{ VI_RJ_DACH : GLOBALER FEHLER RJ_DACH SEITE 48 }
{ MI_AJ : MATRIX AJ SEITE 48 }
{ MRB : MATRIX MRBJ BZW MRBJ+1 SEITE 48 }
{ MIBINV : MRB_JPLUS1_INV SEITE 48 }
VAR I , J , ST : INTEGER;
    T : REAL;
    HI : INTERVAL;

```

```

VIU , VIU_GROB , VIU_MID , VIU_SCHLANGE , VIU_RST : IVECTOR[1..2];
VI_ZJPLUS1 , VI_ZJPLUS1_SCHLANGE , VI_RJ_DACH   : IVECTOR[1..2];
MRB                                             : RMATRIX[1..2,1..2];
MI_AJ , MI_HILFS , MIBINV                     : IMATRIX[1..2,1..2];
TIP , TIP_RST                                 : ITKO[0..P];
TIVECU                                         : ITVEC[1..2,0..P];

BEGIN HI := INTVAL( 0 , H.SUP );
VIU[1] := 0;
VIU[2] := 1;
VIU_MID := MID( VIU );
VI_RJ_DACH := VNULL( 2 );
MRB := ID( 2 );
FOR ST := 0 TO UPBD - 1 DO BEGIN
  TIP := TIVECP[ST];
  TIP[0] := TIP[0] - EIG;
  TIP_RST := TIVECP_RST[ST];
  TIP_RST[0] := TIP_RST[0] - EIG;
  VIU_GROB := GROBEIN( HI , TIP_RST[0] , VIU );
  VI_ZJPLUS1 := RESTGLIED( H , TIP_RST , VIU_GROB );
  TIVECU[1,0] := VIU_MID[1];
  TIVECU[2,0] := VIU_MID[2];
  { TAYLOR-KOEFFIZIENTEN VON U SOWIE DIE MATRIX AJ BERECHNEN }
  PSI_ABLEITUNG( H , TIP , MI_AJ , TIVECU );
  { FORMEL (2.4.1) VON LOHNER }
  VIU_SCHLANGE := PUNKTAUSWERTUNG( H , TIVECU ) + VI_ZJPLUS1;
  VIU_MID := MID( VIU_SCHLANGE );
  { FORMEL (2.4.2) VON LOHNER }
  VI_ZJPLUS1_SCHLANGE := VIU_SCHLANGE - VIU_MID;
  MI_HILFS := MI_AJ * MRB;
  { FORMEL (2.3.6) VON LOHNER }
  MRB := QR2X2( PERM2X2( MID( MI_HILFS ) , VI_RJ_DACH ) );
  MIBINV := MIINV2X2( MRB );
  { FORMEL (2.3.3) VON LOHNER }
  VI_RJ_DACH := (MIBINV * MI_HILFS) * VI_RJ_DACH;
  VI_RJ_DACH := VI_RJ_DACH + MIBINV * VI_ZJPLUS1_SCHLANGE;
  VIU_RST := MRB * VI_RJ_DACH;
  { FORMEL (2.3.1) VON LOHNER }
  VIU := VIU_MID + VIU_RST;
END; { FOR , ST };
AWA := VIU;
END;

{-----}
{ PASCAL-XSC - PROZEDUR AWA_NWT }
{ AWP-PROZEDUR FUER DIE STURM-LIOUVILLE-GLEICHUNG }
{ MIT UNENDLICH OFT DIFFERENZIERBAREN POTENTIALEN }
{ HIER WIRD AUCH DIE FUNKTIONALMATRIZ IM REKONSTRUKTIONSVERFAHREN }
{ FUER DAS INVERSE DIRICHLET-PROBLEM BERECHNET }
{-----}

GLOBAL PROCEDURE AWA_NWT( UPBD , P : INTEGER; H , EIG : INTERVAL;
  TIVECP , TIVECP_RST : ITVEC; VAR TIMATQ : ITMAT;
  VAR VIU : IVECTOR; VAR VI_U2 : INTERVAL; VAR VI_U2Q : IVECTOR );
{ BEZEICHNUNGEN WIE BEI LOHNER }
{ SEITENANGABEN BEZIEHEN SICH AUF DESSEN DISSERTATION }
{ UPBD : ANZAHL DER STUETZSTELLEN IM INTERVALL [0,pi] }
{ P : ORDNUNG DER TAYLOR-ENTWICKLUNG }
{ H , HI : SCHRITTWEITE }
{ EIG : EIGENWERT }
{ TIVECP , TIVECP_RST : TAYLORKOEFFIZIENTEN DES POTENTIALS P(X) }
{ VIU : GESAMTEINSCHLIESSUNG }
{ VIU_GROB : GROBEINSCHLIESSUNGSSCHLANGE SEITE 40 }
{ VIU_SCHLANGE : INTERVALLGROESSE UJSCHLANGE SEITE 54 }
{ VIU_MID : PUNKTGROESSE UJSCHLANGE SEITE 54 }

```

```

{ VI_ZJPLUS1      : DISKRETISIERUNGSFEHLER ZJ+1          SEITE 48 }
{ VI_ZJPLUS1_SCHLANGE : " ZJ+1_SCHLANGE                SEITE 54 }
{                  : (ENTHAELT AUCH DIE RUNDUNGSFEHLER) }
{ VI_RJ_DACH      : GLOBALER FEHLER RJ_DACH              SEITE 48 }
{ MI_AJ           : MATRIX AJ                            SEITE 48 }
{ MRB             : MATRIX MRBJ BZW MRBJ+1              SEITE 48 }
{ MIBINV          : MRB_JPLUS1_INV                      SEITE 48 }

VAR I , NR , ST , N , J      : INTEGER;
    T                        : REAL;
    HI , IX                  : INTERVAL;
    VIU_GROB , VIU_MID , VIU_SCHLANGE , VIU_RST : IVECTOR[1..2];
    VI_ZJPLUS1 , VI_ZJPLUS1_SCHLANGE , VI_RJ_DACH : IVECTOR[1..2];
    MRB                      : RMATRIX[1..2,1..2];
    MI_AJ , MI_HILFS , MIBINV : IMATRIX[1..2,1..2];
    TIP , TIP_RST            : ITKO[0..P];
    TI_U2                    : ITKO[0..2*P];
    TI_U2_IGRAL              : ITKO[0..2*P+1];
    TIVECU , TIVECUINCL      : ITVEC[1..2,0..P];
    TIVECQ                   : ITVEC[0..UB(VI_U2Q),0..P];
    TIVEC_U2Q_IGRAL          : ITVEC[0..UB(VI_U2Q),0..3*P+1];

BEGIN N := UB( VI_U2Q );
    HI := INTVAL( 0 , H.SUP );
    VIU[1] := 0;
    VIU[2] := 1;
    VI_U2 := 0;
    FOR I := 0 TO N DO
        VI_U2Q[I] := 0;
    VIU_MID := VIU;
    VI_RJ_DACH := VNULL( 2 );
    MRB := ID( 2 );
    FOR ST := 0 TO UPBD - 1 DO BEGIN
        TIP := TIVECP[ST];
        TIP[0] := TIP[0] - EIG;
        TIP_RST := TIVECP_RST[ST];
        TIP_RST[0] := TIP_RST[0] - EIG;
        FOR I := 0 TO N DO
            TIVECQ[I] := TIMATQ[I,ST];
        VIU_GROB := GROBEIN( HI , TIP_RST[0] , VIU );
        VI_ZJPLUS1 := RESTGLIED( H , TIP_RST , VIU_GROB );
        TIVECU[1,0] := VIU_MID[1];
        TIVECU[2,0] := VIU_MID[2];
        TIVECUINCL[1,0] := VIU[1];
        TIVECUINCL[2,0] := VIU[2];
        { TAYLOR-KOEFFIZIENTEN VON U SOWIE DIE MATRIX AJ BERECHNEN }
        PSI_ABLEITUNG( H , TIP , MI_AJ , TIVECU , TIVECUINCL );
        TIVECUINCL[1,P] := VI_ZJPLUS1[1] / ( H ** P );
        { TAYLOR-KOEFFIZIENTEN VON U^2 UND INTEGRAL DAVON BERECHNEN }
        TI_U2_IGRAL[0] := VI_U2;
        FOR NR := 0 TO P DO BEGIN
            IX := 0;
            FOR I := 0 TO NR DO
                IX := IX + TIVECUINCL[1,I] * TIVECUINCL[1,NR-I];
            TI_U2[NR] := IX;
            TI_U2_IGRAL[NR+1] := TI_U2[NR] / ( NR + 1 );
        END;
        FOR NR := 1 TO P DO BEGIN
            IX := 0;
            FOR I := NR TO P DO
                IX := IX + TIVECUINCL[1,I] * TIVECUINCL[1,P+NR-I];
            TI_U2[P+NR] := IX;
            TI_U2_IGRAL[P+NR+1] := TI_U2[P+NR] / ( P + NR + 1 );
        END;
    VI_U2 := TI_U2_IGRAL[2*P+1];

```

```

FOR I := 2 * P DOWNT0 0 DO
  VI_U2 := H * VI_U2 + TI_U2_IGRAL[I];
  { TAYLOR-KOEFFIZIENTEN VON U^2 * Q[I] (BASIS-POTENTIALE) }
  { UND INTEGRAL DAVON BERECHNEN }
  FOR I := 0 TO N DO BEGIN
    TIVEC_U2Q_IGRAL[I,0] := VI_U2Q[I];
    FOR NR := 0 TO P - 1 DO BEGIN
      IX := 0;
      FOR J := 0 TO NR DO
        IX := IX + TI_U2[J] * TIVECQ[I,NR-J];
      TIVEC_U2Q_IGRAL[I,NR+1] := IX / ( NR + 1 );
    END;
    FOR NR := 0 TO P DO BEGIN
      IX := 0;
      FOR J := 0 TO P DO
        IX := IX + TI_U2[NR+J] * TIVECQ[I,P-J];
      TIVEC_U2Q_IGRAL[I,P+NR+1] := IX / ( P + NR + 1 );
    END;
    FOR NR := 1 TO P DO BEGIN
      IX := 0;
      FOR J := NR TO P DO
        IX := IX + TI_U2[2*P+NR-J] * TIVECQ[I,J];
      TIVEC_U2Q_IGRAL[I,2*P+NR+1] := IX / ( 2 * P + NR + 1 );
    END;
  END;
  FOR I := 0 TO N DO BEGIN
    VI_U2Q[I] := TIVEC_U2Q_IGRAL[I,3*P+1];
    FOR NR := 3 * P DOWNT0 0 DO
      VI_U2Q[I] := H * VI_U2Q[I] + TIVEC_U2Q_IGRAL[I,NR];
    END;
    { FORMEL (2.4.1) VON LOHNER }
    VIU_SCHLANGE := PUNKTAUSWERTUNG( H , TIVECU ) + VI_ZJPLUS1;
    VIU_MID := MID( VIU_SCHLANGE );
    { FORMEL (2.4.2) VON LOHNER }
    VI_ZJPLUS1_SCHLANGE := VIU_SCHLANGE - VIU_MID;
    MI_HILFS := MI_AJ * MRB;
    { FORMEL (2.3.6) VON LOHNER }
    MRB := QR2X2( PERM2X2( MID( MI_HILFS ) , VI_RJ_DACH ) );
    MIBINV := MIINV2X2( MRB );
    { FORMEL (2.3.3) VON LOHNER }
    VI_RJ_DACH := (MIBINV * MI_HILFS) * VI_RJ_DACH;
    VI_RJ_DACH := VI_RJ_DACH + MIBINV * VI_ZJPLUS1_SCHLANGE;
    VIU_RST := MRB * VI_RJ_DACH;
    { FORMEL (2.3.1) VON LOHNER }
    VIU := VIU_MID + VIU_RST;
  END; { FOR , ST };
END;
END.

```

Literaturverzeichnis

- [1] Ahlberg, J. H., Nilson, E. N., und Walsh, J. L.: *The Theory of Splines and Their Applications*. Academic Press, New York, 1967.
- [2] Alefeld, G.: *Intervallanalytische Methoden bei nichtlinearen Gleichungssystemen*. In: Jahrbuch Überblicke der Mathematik 1979, 63–78, Bibliographisches Institut, Mannheim, 1979.
- [3] Alefeld, G.: *On the Convergence of some Interval-Arithmetic Modifications of Newton's Method*. SIAM J. Numer. Anal. **21**, 363–372, 1984.
- [4] Alefeld, G., und Herzberger, J.: *Einführung in die Intervallrechnung*. Bibliographisches Institut, Mannheim, 1974.
- [5] Alefeld, G., und Herzberger, J.: *Introduction to Interval Computations*. Academic Press, New York, 1983.
- [6] Alefeld, G., Gienger, A., und Potra, F.: *Efficient Numerical Validation of Solutions of Nonlinear Systems*. Reports on Computational Mathematics **35**, University of Iowa, 1992. *Erscheint in* SIAM J. Numer. Anal. **31**, 1994.
- [7] Ambarzumian, V.: *Über eine Frage der Eigenwerttheorie*. Z. Physik **53**, 690–695, 1929.
- [8] Barçilon, V.: *Iterative Solution of the Inverse Sturm-Liouville Equation*. J. Math. Phys. **15**, 429–436, 1974.
- [9] Barnes, D. C.: *The Inverse Eigenvalue Problem with Finite Data*. SIAM J. Math. Anal. **22**, 732–753, 1991.
- [10] Borg, G.: *Eine Umkehrung der Sturm-Liouvilleschen Eigenwertaufgabe*. Acta Math. **78**, 1–96, 1946.
- [11] Collatz, L.: *Eigenwertaufgaben mit technischen Anwendungen*. Akademie-Verlag, Leipzig, 1963.

- [12] Dieudonné, J.: *Grundzüge der modernen Analysis*. Vieweg, Braunschweig, 3. Auflage, 1985.
- [13] Fix, G.: *Asymptotic Eigenvalues of Sturm-Liouville Systems*. J. Math. Anal. Appl. **19**, 519–525, 1967.
- [14] Friedland, S., Nocedal, J., und Overton, M. L.: *The Formulation and Analysis of Numerical Methods for Inverse Eigenvalue Problems*. SIAM J. Numer. Anal. **24**, 634–667, 1987.
- [15] Gel'fand, I. M., und Levitan, B. M.: *On the Determination of a Differential Equation from its Spectral Function*. Amer. Math. Soc. Transl. Series 2 **1**, 253–304, 1951.
- [16] Gladwell, G. M. L.: *Inverse Problems in Vibration*. Martinus Nijhoff Publishers, Dordrecht, 1986.
- [17] Greenberg, L., und Babuška, I.: *A Continuous Analogue of Sturm Sequences in the Context of Sturm-Liouville Equations*. SIAM J. Numer. Anal. **26**, 920–945, 1989.
- [18] Hald, O. E.: *The Inverse Sturm-Liouville Problem with Symmetric Potentials*. Acta Math. **141**, 263–291, 1978.
- [19] Hald, O. E.: *The Inverse Sturm-Liouville Problem and the Rayleigh-Ritz Method*. Math. Comp. **32**, 687–705, 1978.
- [20] Heuser, H.: *Lehrbuch der Analysis, Teil 2*. Teubner, Stuttgart, 6. Auflage, 1991.
- [21] Hochstadt, H.: *Asymptotic Estimates for the Sturm-Liouville Spectrum*. Comm. Pure Appl. Math. **XIV**, 749–764, 1961.
- [22] Hochstadt, H.: *The Inverse Sturm-Liouville Problem*. Comm. Pure Appl. Math. **XXVI**, 715–729, 1973.
- [23] Klein, W.: *Zur Einschließung der Lösung von linearen und nicht-linearen Fredholmschen Integralgleichungssystemen zweiter Art*. Dissertation. Universität Karlsruhe, 1990.
- [24] Kulisch, U.: *Grundlagen des Numerischen Rechnens – Mathematische Begründung der Rechnerarithmetik*. Bibliographisches Institut, Mannheim, 1976.
- [25] Kulisch, U., und Miranker, W. L.: *Computer Arithmetic in Theory and Practice*. Academic Press, New York, 1981.
- [26] Klatte, R. et al.: *PASCAL-XSC. Sprachbeschreibung mit Beispielen*. Springer-Verlag, Berlin, 1991.

- [27] Levitan, B. M.: *On the Determination of a Sturm-Liouville Equation by Spectra*. Amer. Math. Soc. Transl. Series 2 **68**, 1–20, 1964.
- [28] Liouville, J.: *Mémoire sur le développement des fonctions ou parités de fonctions en séries dont le divers termes sont assujettis á une même équation différentielle du second ordre contenant un paramètre variable*. J. de. Math. **1**, 253–265, 1836; **2**, 16–35 und 418–436, 1837.
- [29] Lohner, R.: *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*. Dissertation. Universität Karlsruhe, 1988.
- [30] Lohner, R.: *Enclosing all eigenvalues of symmetric matrices*. In: Ulrich, Ch., Wolff v. Gudenberg, J. (eds): *Accurate Numerical Algorithms*. Springer-Verlag, Berlin, 1989.
- [31] Lowe, B. D., Pilant, M., und Rundell, W.: *The Recovery of Potentials from Finite Spectral Data*. SIAM J. Math. Anal. **23**, 482–504, 1992.
- [32] Marti, J. T.: *Small Potential Corrections for the Discrete Eigenvalues of the Sturm-Liouville Problem*. Numer. Math. **57**, 51–62, 1990.
- [33] McLaughlin, J. R.: *Analytical Methods for Recovering Coefficients in Differential Equations from Spectral Data*. SIAM Rev. **28**, 53–72, 1986.
- [34] Moore, R. E.: *Interval Analysis*. Prentice Hall, Englewood Cliffs, N.J., 1966.
- [35] Neher, M.: *Inclusion of Eigenvalues and Eigenfunctions of the Sturm-Liouville Problem*. In: Atanassova, L., Herzberger, J. (eds): *Computer Arithmetic and Enclosure Methods*. Elsevier, Amsterdam, 1992.
- [36] Neumaier, A.: *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.
- [37] Ortega, J. M.: *Numerical Analysis. A Second Course*. SIAM, Philadelphia, 1990.
- [38] Ortega, J. M., und Rheinboldt, W. C.: *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.
- [39] Pöschel, J., und Trubowitz, E.: *Inverse Spectral Theory*. Academic Press, Orlando, 1987.
- [40] Ratschek, H., und Rokne, J.: *Computer Methods for the Range of Functions*. Ellis Horwood, Chichester, 1984.
- [41] Rundell, W., und Sacks, P. E.: *The Reconstruction of Sturm-Liouville Operators*. Inverse Problems **8**, 457–482, 1992.

- [42] Sacks, P. E.: *An Iterative Method for the Inverse Dirichlet Problem*. Inverse Problems **4**, 1055–1069, 1988.
- [43] Sauer, R., und Szabó, I.: *Mathematische Hilfsmittel des Ingenieurs. Teil II*. Springer-Verlag, Berlin, 1969.
- [44] Stoer, J.: *Numerische Mathematik 1*. Springer-Verlag, Berlin, 4. Auflage, 1983.
- [45] Stoer, J., und Bulirsch, R.: *Numerische Mathematik 2*. Springer-Verlag, Berlin, 3. Auflage, 1990.
- [46] Sturm, C.: *Sur les équations différentielle linéaires du second ordre*. J. de Math. **1**, 106–186, 1836.
- [47] Walter, W.: *Gewöhnliche Differentialgleichungen*. Springer-Verlag, Berlin, 4. Auflage, 1990.
- [48] Walter, W.: *Rand- und Eigenwertprobleme*. Universität Karlsruhe, Vorlesung im Sommer-Semester 1992.