

# **Remez-Algorithmus**

**Bachelorarbeit**

**Tarek Almahana**

**Fakultät für Mathematik  
Universität Passau**

**Februar 2019**

## **Inhaltsverzeichnis**

1. Einleitung	1
2. Charakterisierung der besten Polynomapproximation	4
3. Haar-Bedingung und Alternantensatz	8
4. Der Remez-Algorithmus	14
5. Konvergenz des Remez-Algorithmus	18
6. Matlab-Codes	26
7. Beispiele	30
8. Zusammenfassung	33

## 1 Einleitung

Diese Arbeit konzentriert sich darauf, der Remez -Algorithmus zu erklären.

Der Remez-Algorithmus ist ein iteratives Verfahren, das die beste Minimax-Approximation einer gegebenen stetigen Funktion auf dem Intervall  $[a,b]$  durch ein generalisiertes Polynom konvergiert.

Ein generalisiertes Polynom  $p$  ist eine Funktion der Form

$$p := \sum_{i=1}^n \alpha_i g_i$$

Wobei  $\alpha_1, \dots, \alpha_n$  Skalare sind und  $g_1, \dots, g_n$  sind stetige Funktionen.

Das System  $g_1, \dots, g_n$  ist Teil der Eingabe und muss der Haar-Bedingung unterliegen.

Heute hat der Remez-Algorithmus Applikationen im Filterdesign.

Die Hauptidee des Remez-Algorithmus basiert auf dem Alternantensatz.

Der Alternantensatz liefert uns eine Methode zur direkten Berechnung der besten Minimax-Approximation einer Referenz, die eine diskrete Teilmenge von  $[a,b]$ . Bei jeder Iteration berechnet der Remez-Algorithmus die beste Minimax-Approximation auf der Referenz, der in vorherigen Iteration erhalten hat, und passt dann die Referenz an. Die beste Minimax-Approximation auf diesen neuen Referenz, die in der nächsten Iteration berechnet wird, wird dann eine bessere Approximation auf gesamten Intervall  $[a,b]$ . Die Anfangsreferenz ist Teil der Eingabe und kann frei gewählt werden.

Das Berechnen der besten Minimax-Approximation auf der Referenz ist rechnerisch keine schwierige Aufgabe. Ein Korollar aus dem Alternantensatz zeigt uns, dass dies durch Lösen eines linearen Systems in  $n$  Gleichungen und  $n$  Unbekannten geschieht.

In anderen Schritten des Algorithmus müssen wir jedoch mehrere lokale Extrema der Fehlerfunktion berechnen

$$r(x) = f(x) - p(x),$$

Dabei ist  $f$  die approximant und  $p$  die Bestapproximation der Referenz

Es ist nicht garantiert, dass die Fehlerfunktion differenzierbar ist, und sie hat außerdem normalerweise viele Extreme. Für schnelle Berechnungen ist es erforderlich, die Positionen dieser Extrema auf effiziente Weise zu approximieren.

## 2 Charakterisierung der besten Polynomapproximation

Ein bekanntes Problem in der Approximationstheorie ist es, das beste Polynom des Grads  $n$  der Approximation einer Funktion  $f$  im Minimax Sinne zu finden. Das heißt, wir sind daran interessiert, ein Polynom  $p$  des Grades  $n$  zu finden, das die Quantität  $\max_{x \in [a,b]} |f(x) - p(x)|$  minimiert.

Das Hauptergebnis dieses Abschnitts, der Charakterisierungssatz, bespricht jedoch eine etwas allgemeinere Einstellung, in der unser Polynom vom Grad  $n$  durch ein generalisiertes Polynom  $p := \sum_{i=1}^n \alpha_i g_i$  ersetzt wird.

Hier  $g_1, \dots, g_n$  sind stetige Funktionen auf dem Intervall  $[a, b]$ .

Eine natürliche Frage ist, ob es immer eine beste Approximation durch ein generalisiertes Polynom gibt.

Da die Menge von Linearkombinationen der Funktionen  $g_1, \dots, g_n$  einen endlichen dimensionalen Unterraum von  $C[a, b]$  formt, ist die Existenz einer besten Approximation aus diesem Unterraum garantiert durch Charakterisierungssatz.

**Satz 2.1 (Charakterisierungssatz)** Seien  $f, g_1, \dots, g_n$  stetige Funktionen auf einem kompakten metrischen Raum  $X$  und eine Fehlerfunktion definieren

$$r(x) = \sum_{i=1}^n \alpha_i g_i(x) - f(x).$$

Die Koeffizienten  $\alpha_1, \dots, \alpha_n$  minimieren  $\|r\|_\infty = \max_{x \in X} \left| \sum_{i=1}^n \alpha_i g_i(x) - f(x) \right|$  genau

dann, wenn der Nullvektor in der konvexen Hülle der Menge

$$U = \left\{ r(x)x : |r(x)| = \|r\|_\infty \right\}$$

enthalten ist, wobei  $x = [g_1(x), \dots, g_n(x)]^T$ .

**Definition 2.2** Eine Teilmenge  $A$  eines linearen Raums wird als konvex bezeichnet, wenn  $f, g \in A$ , impliziert dass  $\lambda f + (1-\lambda)g \in A$  für alle  $\lambda \in [0,1]$ .

**Definition 2.3** Sei  $A$  eine Teilmenge eines linearen Raums. Die konvexe Hülle  $H(A)$  von  $A$  ist die Menge, die aus allen endlichen Summen der Form  $g = \sum \lambda_i f_i$  besteht, so dass  $f_i \in A$ ,  $\sum \lambda_i = 1$  und  $\lambda_i \geq 0$  ist. Summen in dieser Form werden Konvexkombination genannt.

**Satz 2.4 (Satz über lineare Ungleichungen)** Sei  $U$  kompakt. Für alle  $z \in \mathbb{R}^n$  gibt es mindestens ein  $u \in U$  so, dass  $\langle u, z \rangle \leq 0$  genau dann, wenn  $0 \in H(A)$ . (Referenz3, S. 19)

**Lemma 2.5** Sei  $X$  ein kompakter metrischer Raum und seien die Funktionen  $r, g_1, \dots, g_n : X \rightarrow \mathbb{R}$  stetig. Dann die Menge

$$U = \{r(x)\hat{x} : |r(x)| = \|r\|_\infty\}, \text{ mit } \hat{x} = [g_1(x), \dots, g_n(x)]^T \text{ und } x \in X$$

Kompakt ist.

**Beweis** Wir werden zeigen, dass  $U$  kompakt ist, indem es zeigt, dass es sich um eine sequentiell kompakte Teilmenge von  $\mathbb{R}^n$  handelt.

Sei  $(u_k)$  beliebige Sequenz in  $U$ . Dann für alle  $k \in \mathbb{N}$  gibt es  $x_k \in X$  so, dass

$$u_k = r(x_k)\hat{x}_k \text{ wobei } \hat{x}_k = [g_1(x_k), \dots, g_n(x_k)]^T.$$

Da  $(x_k)$  eine Sequenz in dem kompakten, also sequentiell kompakten, metrischen Raum  $X$  ist, hat  $(x_k)$  eine konvergente Untersequenz, beispielsweise  $(x_{k_j})$ , mit dem Limit  $x^* \in X$ . Dann durch stetigkeit von  $r$ ,

$$\lim_{j \rightarrow \infty} r(x_{k_j}) = r(x^*), \quad (*)$$

das heißt  $|r(x^*)| = \|r\|_\infty$ . Dann durch stetigkeit von  $g_1, \dots, g_n$ , haben wir

$$\lim_{j \rightarrow \infty} \hat{x}_{k_j} = [g_1(x^*), \dots, g_n(x^*)]^T := \hat{x}^*. \quad (**)$$

Kombinieren von (\*) und (\*\*) finden wir dass

$$\lim_{j \rightarrow \infty} r(x_{k_j})\hat{x}_{k_j} = r(x^*)\hat{x}^*$$

Mit  $|r(x^*)| = \|r\|_\infty$ , das beweisen  $r(x^*)\hat{x}^* \in U$ .

□

**Beweis von Satz 2.1** Beide Implikationen werden durch Widerspruch bewiesen.

( $\Leftarrow$ ) Angenommen  $\|r\|_\infty$  ist nicht minimum. Dann gibt es einen Vektor  $v = [v_1, \dots, v_n] \in \mathbb{R}^n$  so, dass

$$\left\| \sum_{i=1}^n (\alpha_i - v_i) g_i - f \right\|_\infty < \left\| \sum_{i=1}^n \alpha_i g_i - f \right\|_\infty,$$

das heißt,

$$\left\| r - \sum_{i=1}^n v_i g_i \right\|_\infty < \|r\|_\infty. \quad (1)$$

Definieren wir  $X_0 = \{x \in X : |r(x)| = \|r\|_\infty\}$ . Beachten wir, dass diese Definition gerechtfertigt ist, da  $r$  eine stetige Funktion auf einer kompakten Menge ist, und daher zu ihre Extrema in dieser Menge erreicht.

Durch Ungleichung (1) haben wir für  $x_0 \in X$  dass

$$(r(x) - \sum v_i g_i(x))^2 < r(x)^2.$$

durch erweitern wir die linke Seite dieser letzten Ungleichung, finden wir, dass

$$\begin{aligned} r(x)^2 - 2r(x) \sum v_i g_i(x) + (\sum v_i g_i(x))^2 &< r(x)^2 \\ \Rightarrow (\sum v_i g_i(x))^2 &< 2r(x) \sum v_i g_i(x) \\ \Rightarrow 0 < r(x) \sum v_i g_i(x) &= \langle v, r(x) \hat{x} \rangle. \end{aligned} \quad (2)$$

Die Ungleichung (2) sagt uns, gibt es kein Vektor  $u \in U$  so, dass  $\langle v, u \rangle \leq 0$ .

Außerdem zeigt Lemma 2.5, dass das Set  $U$  kompakt ist. Deshalb sagt der Satz über lineare Ungleichungen, dass  $0 \notin H(U)$  ist.

( $\Rightarrow$ ) Angenommen  $0 \notin H(U)$ . Dann sagt uns der Satz über lineare Ungleichungen, dass es einen Vektor  $v = [v_1, \dots, v_n]$  gibt so, dass die Ungleichung (2) für  $x \in X_0$  gilt. Die Menge  $X_0$  ist kompakt (eine geschlossene Teilmenge von  $X$ ). Angenommen für alle  $n$   $(x_n) \rightarrow x^* \in X$  mit

$x_n \in X_0$ ). Dann  $\lim_{n \rightarrow \infty} r(x_n) = r(x^*)$  durch Stetigkeit von  $r$ . Also ist  $r(x^*) = \|r\|_\infty$  da  $r(x_n) = \|r\|_\infty$  für alle  $n$  ist, was  $x^* \in X_0$  impliziert. Wegen der Kompaktheit von  $X_0$  können wir die Zahl  $\varepsilon = \min_{x \in X_0} r(x) \langle v, \hat{x} \rangle$  definieren, die durch Ungleichung (2) positiv ist. Wir definieren

$$X_1 = \left\{ x \in X : r(x) \langle v, \hat{x} \rangle \leq \frac{\varepsilon}{2} \right\}.$$

Diese Menge ist das Urbild einer geschlossenen Menge unter einer stetigen Funktion, also geschlossen. Die Menge  $X_1$  ist kompakt (eine geschlossene Teilmenge von  $X$ ). Beachten wir außerdem, dass die Mengen  $X_0$  und  $X_1$  eine leere Schnittmenge haben. Durch Kompaktheit von  $X_1$  erreicht  $|r(x)|$  sein Supremum  $E < \|r\|_\infty$  auf  $X_1$ . Wir möchten beweisen, dass es  $\lambda > 0$  gibt, so dass  $\|r - \lambda \sum v_i g_i\|_\infty < \|r\|_\infty$ , was bedeutet, dass die Koeffizienten  $\alpha_1, \dots, \alpha_n$  nicht  $\|r\|_\infty$  minimieren. Wir nehmen  $x \in X_1$  und Sei  $0 < \lambda < (\|r\|_\infty - E) / \|\sum v_i g_i\|_\infty$ .

Wir wenden die Dreieck-Ungleichung an, um zu sehen, dass

$$\begin{aligned} |r(x) - \lambda \sum v_i g_i(x)| &\leq |r(x)| + \lambda \left| \sum v_i g_i(x) \right| \\ &\leq E + \lambda \|\sum v_i g_i\|_\infty \\ &\leq \|r\|_\infty \end{aligned} \quad (3)$$

Ist, für alle  $x \in X_1$ . Nun nehmen wir  $x \notin X_1$  und wir wählen  $\lambda$  so dass  $0 < \lambda < \varepsilon / \|\sum v_i g_i\|_\infty^2$ . Dann

$$\begin{aligned} (r(x) - \lambda \sum v_i g_i(x))^2 &= r(x)^2 - 2\lambda r(x) \langle v, \hat{x} \rangle + \lambda^2 (\sum v_i g_i(x))^2 \\ &\leq r(x)^2 - 2\lambda \varepsilon + \lambda^2 (\sum v_i g_i(x))^2 \\ &< \|r\|_\infty^2 + \lambda(-\varepsilon + \lambda \|\sum v_i g_i\|_\infty^2) \\ &< \|r\|_\infty^2 \end{aligned} \quad (4)$$

Ungleichungen (3) und (4) beweisen, dass die Koeffizienten  $\alpha_1, \dots, \alpha_n$  nicht  $\|r\|_\infty$  minimieren. □

### 3 Haar-Bedingung und Alternantensatz

Der Alternantensatz ist der Schlüssel zum Verständnis des Mechanismus des Remez-Algorithmus. außerdem gibt es am Ende dieses Abschnitts ein Korollar des Alternantensatzes, dem explizit in jeder Iteration des Remez-Algorithmus verwendet.

#### Haar-Bedingung 3.1

**Definition 3.1.1** Seien  $g_1, \dots, g_n$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und seien  $x_i \in [a, b]$  für alle  $1 \leq i \leq n$ . Man sagt, dass das System  $\{g_1, \dots, g_n\}$  die Haar-Bedingung erfüllt, wenn die Determinante

$$D[x_1, \dots, x_n] = \begin{vmatrix} g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_n) & \cdots & g_n(x_n) \end{vmatrix} \quad (5)$$

nicht Null ist, wenn  $x_1, \dots, x_n$  alle verschieden sind.

**Beispiel 3.1.2** Das System  $\{1, x, \dots, x^n\}$  erfüllt die Haar-Bedingung. Dafür dieses System, haben wir

$$D[x_1, \dots, x_{n+1}] = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix},$$

die berühmte Vandermonde-Determinante hat den Wert

$$D = \prod_{0 \leq j < i \leq n} (x_i - x_j),$$

**Beispiel 3.1.3** Das System  $\{\sin x, \cos x\}$  erfüllt die Haar-Bedingung auf jedem Intervall  $[a, b] \subset (k\pi, (k+1)\pi); k \in \mathbb{Z}$ : Angenommen  $x_1, x_2 \in [a, b]$ .

Dann

$$\begin{aligned} D[x_1, x_2] &= \begin{vmatrix} \sin x_1 & \cos x_1 \\ \sin x_2 & \cos x_2 \end{vmatrix} \\ &= \sin x_1 \cos x_2 - \cos x_1 \sin x_2 \\ &= \sin(x_1 - x_2) \neq 0 \text{ für } x_1 - x_2 \neq k\pi, k \in \mathbb{Z}. \end{aligned}$$



**Lemma 3.1.4** Sei  $g_1, \dots, g_n$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und nehmen wir an, die Haar-Bedingung ist erfüllt.

Nehmen wir an, dass  $a \leq x_1 < \dots < x_n \leq b$  und  $a \leq y_1 < \dots < y_n \leq b$ .

Dann sind die Determinanten  $D[x_1, \dots, x_n]$  und  $D[y_1, \dots, y_n]$ , die durch (5) definieren, haben dasselbe Vorzeichen.

**Beweis** Durch Widerspruch Nehmen wir an, dass die Bedingungen des Lemmas nicht erfüllt sind. außerdem Nehmen wir ohne Verlust der Allgemeinheit an, dass

$$D[x_1, \dots, x_n] < 0 < D[y_1, \dots, y_n] \quad (6)$$

Andernfalls tauschen wir die Rollen von  $x_1, \dots, x_n$  und  $y_1, \dots, y_n$  aus.

Da die Funktionen  $g_i$  stetig sind, hängt der Wert von  $D[x_1, \dots, x_n]$  kontinuierlich von  $x_i$  ab. Wir können daher die stetige Funktion

$$f : [0, 1] \rightarrow \mathbb{R}, f(\lambda) = D[\lambda x_1 + (1-\lambda)y_1, \dots, \lambda x_n + (1-\lambda)y_n] \quad (7)$$

Definieren. Durch den Zwischenwertsatz und die Annahme (6) gibt es  $\lambda^* \in (0, 1)$  so, dass  $f(\lambda^*) = 0$  ist. Aus der Haar-Bedingung folgt dann, dass nicht alle Einträge in der Determinante

$$D[\lambda^* x_1 + (1-\lambda^*)y_1, \dots, \lambda^* x_n + (1-\lambda^*)y_n]$$

verschieden sind; ansonsten ist diese Determinante ungleich Null.

Mit anderen Worten, es gibt ein  $i \neq j$  so dass

$$\lambda x_i + (1-\lambda)y_i = \lambda x_j + (1-\lambda)y_j,$$

dass heißt

$$\lambda(x_i - x_j) = (1-\lambda)(y_j - y_i),$$

so dass die Quantities  $(x_i - x_j)$  und  $(y_j - y_i)$ , entgegengesetzte Vorzeichen haben, das heißt, nicht beide Mengen  $\{x_1, \dots, x_n\}$  und  $\{y_1, \dots, y_n\}$  sind in aufsteigender Reihenfolge. □

**Lemma 3.1.5** Sei  $g_1, \dots, g_n$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und nehmen wir an, die Haar-Bedingung ist erfüllt.

Nehmen wir an, dass  $a \leq x_1 < \dots < x_n \leq b$ ,  $a \leq y_1 < \dots < y_n \leq b$  und die Konstanten  $\lambda_0, \dots, \lambda_n$  ungleich Null sind. Zusätzlich lassen

$$A = \{\lambda_i \hat{x}_i : \hat{x}_i = [g_1(x_i), \dots, g_n(x_i)], 0 \leq i \leq n\}.$$

Dann  $0 \in H(A)$  genau dann, wenn  $\lambda_i \lambda_{i-1} < 0$  für  $1 \leq i \leq n$ .

**Beweis** Sei die Menge  $A$  wie in der Aussage des Lemmas definiert.

Wir haben  $0 \in H(A)$  genau dann, wenn es Konstanten  $\theta_i > 0, i = 0, \dots, n$  (Wenn einer von ihnen gleich null wäre, würde die Haar-Bedingung verletzt werden) so dass

$$\sum_{i=0}^n \theta_i \lambda_i \hat{x}_i = 0. \quad (8)$$

Hier beachten wir, dass wir den Vektor  $[\theta_0, \dots, \theta_n]$  normieren könnten, so dass  $\sum \theta_i = 1$  ist. Aus Gleichung (8) folgt, dass wir

$$\hat{x}_0 = -\sum_{i=1}^n \frac{\theta_i \lambda_i}{\theta_0 \lambda_0} \hat{x}_i$$

schreiben können, die wir als Matrix-Vektor-Gleichung schreiben:

$$\begin{bmatrix} g_1(x_1) & \cdots & g_1(x_n) \\ \vdots & \ddots & \vdots \\ g_n(x_1) & \cdots & g_n(x_n) \end{bmatrix} \begin{bmatrix} -\theta_1 \lambda_1 / \theta_0 \lambda_0 \\ \vdots \\ -\theta_n \lambda_n / \theta_0 \lambda_0 \end{bmatrix} = \begin{bmatrix} g_1(x_0) \\ \vdots \\ g_n(x_0) \end{bmatrix}.$$

Wir wenden die Cramer-Regel an, um zu finden

$$\frac{-\theta_i \lambda_i}{\theta_0 \lambda_0} = \frac{D[x_1, \dots, x_{i-1}, x_0, x_{i+1}, \dots, x_n]}{D[x_1, \dots, x_n]} \quad (9)$$

Wir ordnen  $x_i$  in der Determinante im Zähler durch Verschieben von  $x_0$   $i-1$  Stellen nach links, dh die Determinante ändert das Vorzeichen  $i-1$  mal.

Nach Lemma 3.1.4 haben der Zähler und Nenner in (9) das gleiche Vorzeichen, wenn wir  $x_0$   $i-1$ -Stellen links platziert haben. Daher

$$\operatorname{sgn}\left(\frac{\theta_i \lambda_i}{\theta_0 \lambda_0}\right) = (-1)^{i-1}. \text{ da } \theta_0 \lambda_0 > 0 \text{ und } \theta_i > 0 \text{ für alle } i, \text{ erhalten wir}$$

$\operatorname{sgn}(\lambda_i) = (-1)^i$  und schließen daraus, dass die  $\lambda_i$  alternieren im Vorzeichen.

Um die umgekehrte Richtung zu beweisen, sei  $\operatorname{sgn}(\lambda_i) = (-1)^i$ . Dann können wir in der Lösung (9) für die Gleichung (8) alle  $\theta_i$  streng positiv wählen, dass heißt  $0 \in H(A)$ . □

## Der Alternantensatz 3.2

### Satz 3.2.1 (Alternantensatz)

Sei  $g_1, \dots, g_n$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und wir nehmen an, die Haar-Bedingung ist erfüllt. Sei  $X$  eine geschlossene Teilmenge von  $[a, b]$  mit mindestens  $n + 1$  Punkten. Sei  $f$  eine stetige Funktion, die auf  $X$  definiert, und  $r$  definiert eine Fehlerfunktion als folgt

$$r(x) = \sum_{i=1}^n \alpha_i g_i(x) - f(x).$$

Die Koeffizienten  $\alpha_1, \dots, \alpha_n$  minimieren  $\|r\|_\infty = \max_{x \in X} \left| \sum_{i=1}^n \alpha_i g_i(x) - f(x) \right|$  genau dann, wenn

$$r(x_i) = -r(x_{i-1}) = \pm \|r\|_\infty$$

für  $a \leq x_1 < \dots < x_n \leq b$  mit  $x_0, \dots, x_n \in X$ .

**Satz 3.2.2 (Carathéodory)** Sei  $A$  eine Teilmenge eines  $n$ -dimensionalen linearen Raums. Jeder Punkt in der konvexen Hülle von  $A$  kann als konvexe Linearkombination von nicht mehr als  $n + 1$  Elementen von  $A$  ausgedrückt werden. (Referenz 3, S.17)

### Beweis von Satz 3.2.1 (Alternantensatz)

( $\Rightarrow$ ) wir nehmen an, Die Koeffizienten  $\alpha_1, \dots, \alpha_n$  minimieren  $\|r\|_\infty$ .

wir Bezeichnen den Vektor  $[g_1(x), \dots, g_n(x)]$  mit  $\hat{x}$  und definieren die Menge

$$U = \{r(x)\hat{x} : |r(x)| = \|r\|_\infty, x \in X\}.$$

Nach Charakterisierungssatz  $0 \in H(U)$ . Nach dem Satz von Carathéodory kann man jedes Element in  $H(U)$  als eine konvexe Linearkombination von nicht mehr als  $n + 1$  Elementen aus  $U \subset \mathbb{R}^n$  geschrieben werden, das heißt, es gibt eine ganze Zahl  $k \leq n + 1$  und Skalare  $\lambda_0, \dots, \lambda_k$ , die alle streng positiv sind, so dass

$$0 = \sum_{i=0}^k \lambda_i r(x_i) \hat{x}_i, \quad r(x_i) \hat{x}_i \in U. \quad (10)$$

Hier ist  $\hat{x}_i = [g_1(x), \dots, g_n(x)]$ . Nach der Gleichung (10),  $0 \in H(A)$  wobei  $A = \{\lambda_i r(x_i) \hat{x}_i : i = 0, \dots, n\}$ . Lemma 3.1.5 sagt uns, Dies ist nur möglich wenn  $\lambda_i r(x_i) \lambda_{i-1} r(x_{i-1}) < 0; i = 1, \dots, n$ . Da die  $\lambda_i$  alle streng positiv sind,  $r(x_i)$  alterniert im Zeichen. Da  $r(x_i) \hat{x}_i \in U$  für alle  $i$ , beachten wir, dass  $|r(x_i)| = \|r\|_\infty; i = 0, \dots, n$ .

( $\Leftarrow$ ) Angenommen  $a \leq x_0 < \dots < x_n \leq b$  für  $x_0, \dots, x_n \in X$ , und  $r(x_i) = -r(x_{i-1}) = \pm \|r\|_\infty$  ist. Dann, da  $r(x_i)$  alterniert im Zeichen, Lemma 3.1.5 sagt uns, dass  $0 \in H(B)$  wobei

$$\begin{aligned} B &= \{r(x_i) \hat{x}_i : i = 0, \dots, n\} \\ &= \{r(x_i) \hat{x}_i : |r(x_i)| = \|r\|_\infty, i = 0, \dots, n\} \\ &\subset U, \end{aligned}$$

Da  $0 \in H(B)$ , dann ist  $0 \in H(U)$ . Nach Charakterisierungssatz wurden die Koeffizienten  $\alpha_1, \dots, \alpha_n$  so gewählt, dass die Supremumsnorm der Fehlerfunktion

$$r(x) = \sum_{i=1}^n \alpha_i g_i(x) - f(x).$$

auf  $[a, b]$  minimiert wird. □

In einigen einfachen Fällen kann die beste Approximation an eine Funktion durch direkte Anwendung des Alternantensatzes gefunden werden. Nachfolgend wird ein Beispiel dafür beschrieben.

**Beispiel 3.2.2** lässt uns die beste lineare Approximation  $P(x) = a_0 + a_1 x$  für die Funktion  $f(x) = e^x$  auf  $[0, 1]$  finden, d.h. Unser Haar-System ist  $\{1, x\}$ . nach dem Alternantensatz muss die Fehlerfunktion  $r = f - P$  mindestens dreimal alternieren. Die alternante Punkte sind 0, 1 und ein Punkt  $\varepsilon$  zwischen den beiden, an die alternante Punkte haben wir  $|f(x) - P(x)| = \|r\|_\infty := \varepsilon$ . Wir erhalten die folgenden Gleichungen:

$$\begin{aligned} \varepsilon &= f(0) - P(0) = 1 - a_0 \\ -\varepsilon &= f(\gamma) - P(\gamma) = e^\gamma - c_0 - c_1 \gamma \\ \varepsilon &= f(1) - P(1) = e - c_0 - c_1 \\ 0 &= f'(\gamma) - P'(\gamma) = e^\gamma - c_1 \end{aligned} \text{ Daf } -P \text{ einen extremen Wert bei } \gamma \text{ hat}$$

Durch die Lösung der vorherigen Gleichungen erhalten wir:

$$a_0 = 1 - \varepsilon, a_1 = e^\gamma \text{ und } \gamma = \log(e - 1), \text{ und dann } \varepsilon = \frac{2 - e + (e - 1) \log(e - 1)}{2} \approx 0,106.$$

Die beste lineare Approximation an  $e^x$  auf  $[0,1]$  ist

$$P(x) = (e - 1)x + 1 - \varepsilon. \quad \square$$

Das nächste Korollar wird explizit im ersten Schritt jeder Iteration des Remez-Algorithmus verwendet.

**Korollar 3.2.3** Sei  $g_1, \dots, g_n$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und die die Haar-Bedingung ist erfüllt. und  $r$  definiert eine Fehlerfunktion als folgt

$$r(x) = \sum_{i=1}^n \alpha_i g_i(x) - f(x).$$

Wir nehmen an, dass  $a \leq x_1 < \dots < x_n \leq b$ . Die Koeffizienten  $\alpha_1, \dots, \alpha_n$  minimieren  $\max_{i=0, \dots, n} |r(x_i)|$  werden durch Lösen des folgenden linearen Systems aus  $n$  Gleichungen und  $n$  Unbekannten

$$\sum_{j=1}^n \alpha_j [g_j(x_i) - (-1)^i g_j(x_0)] = f(x_i) - (-1)^i f(x_0), \quad i = 1, \dots, n \quad (11)$$

Erhalten.

**Satz 3.2.3 (Existenztheorem)** Sei  $X$  ein normierter linearer Raum und  $Y$  ein endlicher dimensionaler Unterraum von  $X$ . Dann gibt es für alle  $x \in X$  ein Element  $y^* \in Y$  so, dass  $\|x - y^*\| = \inf_{y \in Y} \|x - y\|$ . (Referenz 3, S. 20).

**Beweis von Korollar 3.2.2** Wir beginnen mit  $p(x) = \sum_{j=1}^n \alpha_j g_j(x)$ .

Die Anwendung des Alternationsatzes mit  $X = \{x_i : a \leq x_0 < \dots < x_n \leq b\}$  gibt uns

$$f(x_{i+1}) - p(x_{i+1}) = -[f(x_i) - p(x_i)], \quad i = 0, \dots, n. \quad (12)$$

Sei  $h = f(x_0) - p(x_0)$ . nach Alternantensatz,  $|r(x_i)| = |h|$  für alle  $i$ . Aus Gleichung (12) folgt nun dass

$$f(x_i) - p(x_i) = (-1)^i h \quad (13)$$

für alle  $i$ . Die letzte Gleichung können wir als

$$\begin{aligned}
f(x_i) - \sum_{j=1}^n \alpha_j g_j(x_i) &= (-1)^i \left[ f(x_0) - \sum_{j=1}^n \alpha_j g_j(x_0) \right] \\
\Rightarrow \sum_{j=1}^n \alpha_j [g_j(x_i) - (-1)^i g_j(x_0)] &= f(x_i) - (-1)^i f(x_0), \quad i = 1, \dots, n \quad (14)
\end{aligned}$$

Schreiben. Die Matrix, die zum System gehört, ist nichtsingulär. Existenztheorem und Alternantensatz garantieren die Existenz des Lösungsvektors  $\alpha = [\alpha_1, \dots, \alpha_n]$  für alle  $f \in C[a, b]$

□

#### 4 Der Remez-Algorithmus

Der Remez-Algorithmus ist ein iteratives Verfahren, der basiert auf dem Alternantensatz, welcher die beste Approximation an eine stetige Funktion im Sinne der Minimax findet. Genauer wir sagen, Sei  $f \in C[a, b]$  und Sei  $g_1, \dots, g_n$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und wir nehmen an, die Haar-Bedingung ist erfüllt. Der Algorithmus wird einen Koeffizientenvektor  $\alpha = [\alpha_1, \dots, \alpha_n]$  finden, der die Supremumsnorm der Funktion  $r(x) = f(x) - \sum_{j=1}^n \alpha_j^* g_j(x)$  auf dem Intervall  $[a, b]$  minimiert.

In der Praxis beenden wir das Verfahren, sobald unsere Approximation nahe genug an der besten Approximation liegt.

Bevor wir den Algorithmus starten, machen wir eine Bemerkung, die hilfreich sein kann, um die Schritte des Remez-Algorithmus zu verstehen.

**Bemerkung 4.1** Sei  $\{g_1, \dots, g_n\}$  stetige Funktionen, die auf dem Intervall  $[a, b]$  definiert sind, und wir nehmen an, die Haar-Bedingung ist erfüllt. Sei  $f \in C[a, b]$  und seien  $a \leq x_0 < \dots < x_n \leq b$ . Korollar 3.2.2 sagt uns, dass die Lösung des linearen Systems von (11) gegeben, gibt uns die Koeffizienten, die

$$\max_{i=0, \dots, n} \left| f(x_i) - \sum_{j=1}^n \alpha_j^* g_j(x_i) \right|$$

minimieren. Definieren wir

$$r(x) = f(x) - \sum_{j=1}^n \alpha_j g_j(x).$$

Nach alternantensatz können wir schreiben

$$r(x_i) = -r(x_{i-1}) = \pm \|r\|_\infty, \quad i = 1, \dots, n,$$

dass heißt, dass die Fehlerfunktion  $r$  in jedem Intervall  $(x_{i-1}, x_i)$  eine Nullstelle hat.

□

### Remez-Algorithmus mit einfachem Austauschschritt 4.3

**Input:** Das Intervall  $[a, b]$ . Eine Funktion  $f \in C[a, b]$ . Funktionen  $g_1, \dots, g_n \in C[a, b]$ , die Haar-Bedingung erfüllt. Eine Anfangsreferenz  $\{x_0, \dots, x_n\}; a \leq x_0 < \dots < x_n \leq b$ . Eine Konstante  $\delta$  für das Abbruchkriterium.

**Wir beschreiben die Schritte einer Iteration  $k$ .**

**Schritt 1:** Wenn  $k = 1$  ist, wird die Referenz  $\{x_0, \dots, x_n\}$  von Input bestimmt, andernfalls wird sie in der vorherigen Iteration definiert. die Lösung des linearen Systems (11), gibt uns die Koeffizienten  $\alpha_1, \dots, \alpha_n$ , die

$$\max_{i=0, \dots, n} \left| f(x_i) - \sum_{j=1}^n \alpha_j^* g_j(x_i) \right|$$

minimieren. definiert

$$r(x) = f(x) - \sum_{j=1}^n \alpha_j g_j(x).$$

mit den gerade berechneten Koeffizienten.

**Abbruchkriterium** wir finden  $y \in [a, b]$  so dass,

$$|r(y)| = \max_{x \in [a, b]} |r(x)|.$$

Hör auf zu iterieren, wenn

$$|r(y)| - |r(x_0)| < \delta.$$

sonst fahren wir mit Schritt 2 fort.

**Schritt 2** man definiert  $z_0 = a, z_{n+1} = b$  und man findet eine Nullstelle  $z_i$  in  $(x_{i-1}, x_i)$  für alle  $i = 1, \dots, n$ . Die Existenz dieser Nullstellen wurde in der Bemerkung 4.1 erwähnt

**Schritt 3** Sei  $\sigma_i = \text{sgn } r(x_i)$ . für alle  $i = 0, \dots, n$ , man findet  $y_i \in [z_i, z_{i+1}]$  wobei ist  $\sigma_i r(y_i)$  maximal mit der Eigenschaft, dass  $\sigma_i r(y_i) \geq \sigma_i r(x_i)$ . man ersetzt die Referenz  $\{x_0, \dots, x_n\}$  durch  $\{y_0, \dots, y_n\}$ .

**Schritt 4** Im Abbruchkriterium haben wir  $y \in [a, b]$  wie folgt berechnet

$$|r(y)| = \max_{x \in [a, b]} |r(x)|.$$

Wenn

$$|r(y)| = \max_{i=0, \dots, n} |r(y_i)|,$$

fahr mit Schritt 1 der Iteration  $k+1$  mit der Referenz, die in schritt 3 definiert, fort. Wenn

$$|r(y)| > \max_{i=0, \dots, n} |r(y_i)|,$$

Füg den Punkt  $y$  in die Menge  $\{y_0, \dots, y_n\}$  ein und setze es in die richtige Position, so dass sich die Elemente noch in aufsteigender Reihenfolge befinden. Entferne schließlich eines der  $y_i$  so, dass sich  $r(x)$  immer noch im Vorzeichen der resultierenden Menge alterniert.

**4.2.1 Eine Visualisierung der Referenzanpassung** In Abbildung (1) ist zu sehen, wie die Referenz in einer Iteration des Remez-Algorithmus angepasst wird. Die Abbildung zeigt die Fehlerfunktion in der ersten Iteration zum Bestimmen einer Polynomapproximation zweiten Grades für die Funktion

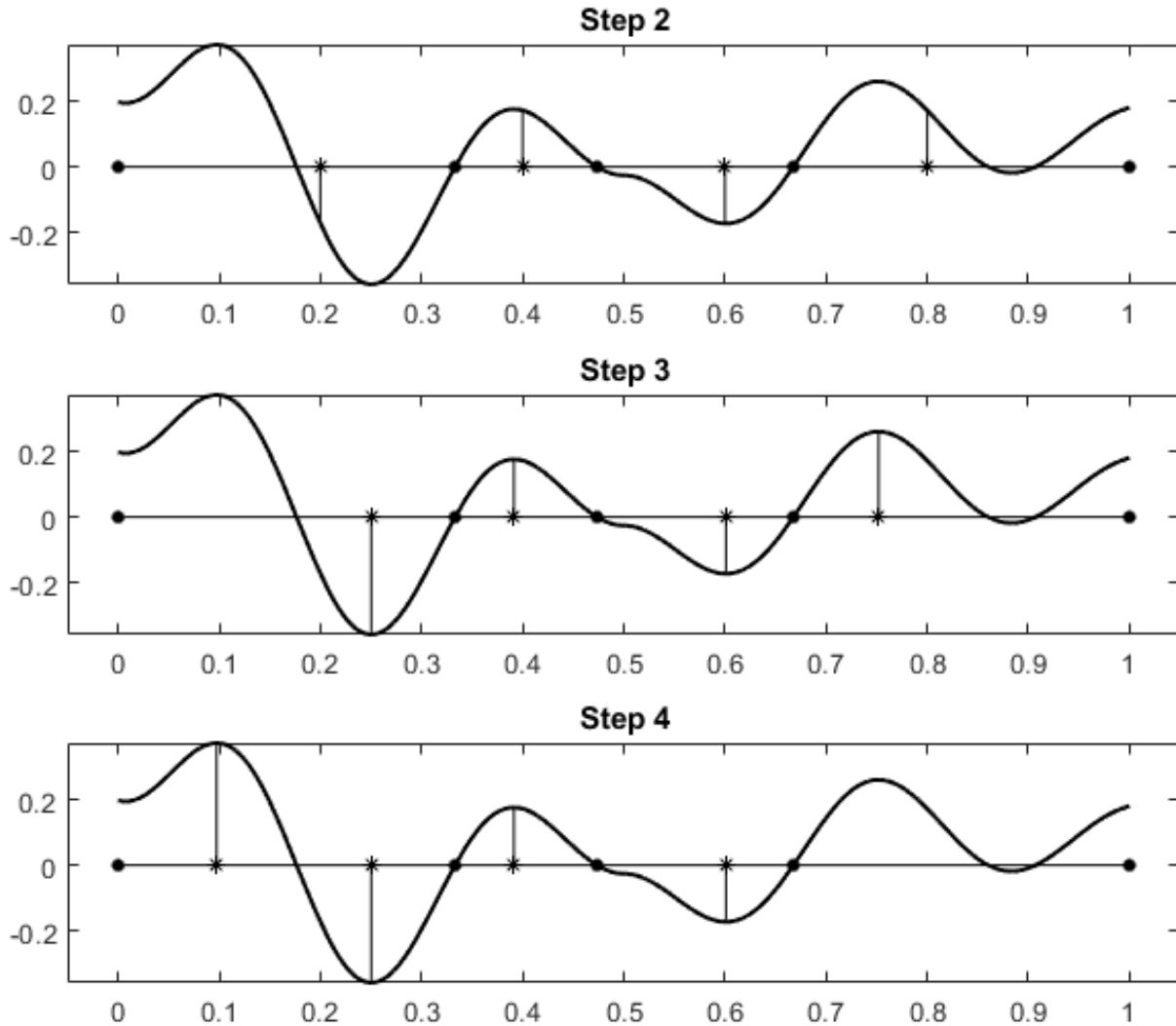
$$f(x) = e^{-x} \sin(3\pi x) \cos(3\pi x) |\sin(2\pi x)|$$

auf dem Intervall  $[0, 1]$ . Beachten wir, dass unser Haar-System für eine Polynomapproximation zweiten Grades

$$\{1, x, x^2\}$$

ist, d. H. Wir haben 3 Basiselemente. Daher muss unsere Anfangsreferenz aus 4 Punkten bestehen. Als Anfangsreferenz wählten wir 4 Knoten in  $[0.2, 0.8]$  Diese Wahl für  $f$  und für die Anfangsreferenz ist das Ergebnis von Versuch und Irrtum. Ziel war es, Plots zu erhalten, in denen sichtbar ist, wie die Referenz in jedem Schritt einer Iteration angepasst wird.





**Abbildung (l)** : Fehlerfunktion und Referenzpunkte in den Schritten 2,3,4 der ersten Iteration des Remez-Algorithmus zum Bestimmen einer Polynomapproximation zweiten Grades an die Funktion  $f(x)$  .

In jedem Schritt werden die Positionen der Referenzpunkte und die Zahlen  $z_0, \dots, z_5$  durch Sterne und Punkte beziehungsweise angezeigt. In Schritt 2 sind Äquioskillationen auf der Anfangsreferenz sichtbar. In Schritt 3 werden die Referenzpunkte auf die Position der lokalen Maxima umgesiedelt. Im letzten Schritt wird die Position des globalen Maximums von  $|r|$  in die Referenz aufgenommen und ein Punkt entfernt.

## 4.2.2 Anpassung der Referenz

**Fall 1:** wir Nehmen an, dass  $y_0 < y < y_n$ . In diesem Fall hat der neue Menge zwei benachbarte Punkte, an denen  $r(x)$  dasselbe Vorzeichen hat. Wir werden die eine der beiden behalten, auf der  $|r(x)|$  den größten Wert hat. Dieser Fall kann wie folgt aussehen. Nach dem Einfügen von  $y$  in  $\{y_0, \dots, y_n\}$  haben wir die folgende Menge:

$$\{y_0, \dots, y_{i-1}^+, y^+, y_i^-, \dots, y_n\},$$

Wir haben  $|r(y)| > |r(y_{i-1})|$  und werden daher  $y_{i-1}$  entfernen. auf der neuen Menge  $\{y_0, \dots, y_{i-2}^-, y^+, y_i^-, \dots, y_n\}$ ,  $r(x)$  alternierend im Vorzeichen. Diese Menge ist die Referenz, mit der wir die nächste Iteration starten.

**Fall 2** wir Nehmen an, dass  $a < y < y_0$ . Zwei Situationen können auftreten; Wenn die neue Menge zwei benachbarte Punkte hat, an denen  $r(x)$  dasselbe Vorzeichen hat, behalten wir den Punkt, an dem  $|r(x)|$  den größten Wert hat, wie im ersten Fall. Dies gibt die gewünschte Referenz.

Andernfalls alternierend  $r$  bereits auf der neuen Menge und wir setzen

$$y_n = y_{n-1}, y_{n-1} = y_{n-2}, \dots, y_0 = y,$$

was zu einer geordneten Menge von  $n+1$  Punkten führt, an denen  $r(x)$  alternierend ist. Das Ergebnis ist, dass  $y$  in der Menge enthalten ist und der alte Wert  $y_n$  entfernt wird.

**Fall 3** wir Nehmen an, dass  $y_n < y < b$ . Ein zu entfernender Punkt  $y_i$  kann auf analoge Weise wie im zweiten Fall beschrieben gewählt werden.

## 5 Konvergenz des Remez-Algorithmus

Es ist für viele Approximanten bekannt, dass die Konvergenz des Remez-Algorithmus quadratisch ist. Wir werden diese Tatsache hier nicht beweisen, sondern abschließend einen Beweis für die lineare Konvergenz vorlegen, der für jede stetige Approximation gilt. Bevor wir dies tun, müssen wir den Satz von de La Vallée-Poussin und den Strong unicityssatz beweisen. Zuerst beweisen wir ein kurzes Lemma über die Haar-Bedingung.

**Lemma 5.1** Das System der stetige Funktionen  $\{g_1, \dots, g_n\}$  erfüllt die Haar-Bedingung genau dann, wenn kein nicht-triviales generalisiertes Polynom  $\sum_{i=1}^n \alpha_i g_i$  mehr als  $n-1$  verschiedene nullstellen hat.

**Beweis:**

( $\Rightarrow$ ) Angenommen, die Haar-Bedingung ist erfüllt. Dann ist die Matrix in der Gleichung

$$\begin{bmatrix} g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_n) & \cdots & g_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad (15)$$

nicht singulär, wenn  $x_1, \dots, x_n$  alle verschieden sind. In diesem Fall ist die einzige Lösung die triviale Lösung  $\alpha_1 = \dots = \alpha_n = 0$ . Ein nicht triviales generalisiertes Polynom kann  $n$  oder mehr nullstellen haben.

( $\Leftarrow$ ) Durch widerspruch. Angenommen, die Haar-Bedingung ist nicht erfüllt. Dann gibt es  $x_1, \dots, x_n$  alle verschiedene, so dass die Matrix in Gleichung (15) den Rang  $< n$  hat. In diesem Fall existiert ein nichttrivialer Lösungsvektor  $[\alpha_1, \dots, \alpha_n]^T$  daher hat das nichttrivial generalisierte Polynom  $\sum_{i=1}^n \alpha_i g_i$  die Nullstellen  $x_1, \dots, x_n$ . □

Der nächste Satz gibt eine untere Grenze für die größte Abweichung zwischen Approximant und der besten Minimax-Approximation an.

**Satz (de La Vallée-Poussin) 5.2** Sei das System der stetige Funktionen  $\{g_1, \dots, g_n\}$  die Haar-Bedingung erfüllt. Definieren wir  $E(f) = \inf \|P - f\|_\infty$ . Sei  $P$  ein generalisiertes Polynom, so dass  $f - P$  abwechselnd positiv und negativ bei  $n+1$  aufeinander folgende Punkte  $x_i \in [a, b]$  ist. Dann

$$E(f) \geq \min_i |f(x_i) - P(x_i)|.$$

**Beweis:** Durch Widerspruch. Angenommen dass,

$$E(f) < \min_i |f(x_i) - P(x_i)|.$$

Dann existiert ein generalisiertes Polynom  $P_0$  so, dass

$$\max_{x \in [a,b]} |f(x) - P_0(x)| < \min_i |f(x_i) - P(x_i)|.$$

Jetzt schreiben wir  $P_0 - P = (f - P) - (f - P_0)$ . Aus der obigen Ungleichung folgt, dass das generalisierte Polynom  $P_0 - P$  an den  $n+1$  aufeinander folgenden Punkten  $x_i$  im Vorzeichen wechselt ist. Aber dann  $P_0 - P$  hat  $n$  Nullstellen, was dem Lemma widerspricht wir haben es gerade bewiesen.  $\square$

**Satz (Strong unicity) 5.3** Sei das System der stetige Funktionen  $\{g_1, \dots, g_n\}$  die Haar-Bedingung erfüllt, sei  $P^*$  die beste generalisierte Polynomapproximation an  $f \in C[a, b]$ . Dann existiert eine Konstante  $\mu(f) > 0$  so, dass für alle anderen generalisierten Polynome, haben wir

$$\|f - P\|_\infty \geq \|f - P^*\|_\infty + \mu \|P^* - P\|_\infty.$$

**Beweis:** Zuerst betrachten wir  $\|f - P^*\|_\infty = 0$ . Dann wenden wir die Dreiecksungleichung an, um zu sehen

$$\begin{aligned} \|P - P^*\|_\infty &= \|(f - P) - (f - P^*)\|_\infty \\ &\leq \|f - P\|_\infty + \|f - P^*\|_\infty \\ &= \|f - P\|_\infty. \end{aligned}$$

In diesem Fall wählen wir  $\mu = 1$ .

Angenommen, dass  $\|f - P^*\|_\infty > 0$ . Sei  $r(x) = f(x) - P^*(x)$ . Da angenommen wird, dass  $P^*$  die beste Minimaxapproximation für  $f$  ist, sagt Charakterisierungssatz uns, dass  $0$  in der konvexen Hülle der Gruppe

$$U = \left\{ r(x) [g_1(x), \dots, g_n(x)]^T : |r(x)| = \|r\|_\infty \right\}$$

Enthalten ist. und dann können wir schreiben

$$0 = \sum_{i=0}^n \theta_i r(x_i) [g_1(x_i), \dots, g_n(x_i)]^T ; \theta_i \geq 0 \text{ und } \sum \theta_i = 1.$$

Sei  $\sigma_i = \text{sgn}(r(x_i))$  für  $i = 0, \dots, n$ , dann

$$0 = \sum_{i=0}^n \theta_i \sigma_i [g_1(x_i), \dots, g_n(x_i)]^T ; \theta_i \geq 0.$$

Das ist eine Gleichung  $0 = \sum_{i=0}^n \theta_i \sigma_i g_j(x_i)$  gilt für  $j = 1, \dots, n$ .

Nach der Haar-Bedingung  $k \geq n$ . Charakterisierungssatz sagt uns, dass  $k \leq n$ .

und wir schließen  $k = n$ . Sei  $Q(x) = \sum_{j=1}^n \alpha_j g_j(x)$  ein generalisiertes Polynom mit Norm 1. Dann

$$\begin{aligned} \sum_{i=0}^n \theta_i \sigma_i Q(x_i) &= \sum_{i=0}^n \theta_i \sigma_i \sum_{j=1}^n \alpha_j g_j(x_i) \\ &= \sum_{j=1}^n \alpha_j \sum_{i=0}^n \theta_i \sigma_i g_j(x_i) \\ &= 0. \end{aligned}$$

Nach der Haar-Bedingung, sagt Lemma 5.1 uns, dass  $\sigma_i Q(x_i)$  nicht alle Null sein kann. Daher muss mindestens einer der  $\sigma_i Q(x_i)$  streng positiv sein.

Dies impliziert, dass  $\max_i \sigma_i Q(x_i)$  eine streng positive Funktion von  $Q$  ist.

Die Menge

$$\left\{ Q(x) = \sum_{j=1}^n \alpha_j g_j(x) : \|Q\|_{\infty} = 1 \right\}$$

ist kompakt wie es ist eine geschlossene und begrenzte Teilmenge des endlichen dimensionalen linearen Raums. Also die Nummer

$$\mu = \min_{\|Q\|_{\infty}=1} \max_i \sigma_i Q(x_i) \quad (16)$$

ist streng positive wie es ist das Minimum einer streng positiven stetigen Funktion auf einer kompakten Menge. Nun Sei  $P$  ein generalisiertes Polynom. Es gibt zwei Fälle. Wenn  $P = P^*$ , folgt die Ungleichheit, die wir

beweisen wollen, trivial, da wir in diesem Fall willkürlich wählen können. Ansonsten hat das generalisierte Polynom

$$Q = \frac{P^* - P}{\|P^* - P\|_\infty}$$

die Norm 1. Nach der Definition von  $\mu$  in (16) haben wir  $\sigma_i Q(x_i) \geq \mu$  für einen Index  $i$ . Folglich für diesen Index  $\sigma_i (P^* - P)(x_i) \geq \mu \|P^* - P\|_\infty$  und dann

$$\begin{aligned} \|f - P\|_\infty &\geq \sigma_i (f - P)(x_i) \\ &= \sigma_i (f - P^*)(x_i) + \sigma_i (P^* - P)(x_i) \\ &\geq \|f - P^*\|_\infty + \mu \|P^* - P\|_\infty. \end{aligned}$$

Im letzten Schritt haben wir die Tatsache ausgenutzt, dass

$$f(x_i) - P^*(x_i) = r(x_i) = \pm \|r\|_\infty \text{ und dann } \sigma_i r(x_i) = \|r\|_\infty.$$

□

**Satz (Konvergenz des Remez-Algorithmus) 5.4** Sei  $P^k$  das generalisierte Polynom, das in der Iteration  $k$  im Remez-Algorithmus erhalten wird, und sei  $P^*$  die beste Minimax-Approximation, die vom entsprechenden Haar-System aufgespannt wird. Dann eine Ungleichheit der Form

$$\|P^k - P^*\|_\infty \leq A\theta^k, \quad 0 < \theta < 1$$

Das heißt  $P^k \rightarrow P^*$  gleichförmig.

**Beweis:** Wir verwenden die Notation aus der Beschreibung des Remez-Algorithmus im vorherigen Abschnitt. Am Ende der Iteration  $k$ , man kann definieren

$$\begin{aligned} \alpha &= \min_i |r(x_i)| = \max_i |r(x_i)|, \\ \beta &= \max_i |r(y_i)| = \|r\|_\infty, \\ \gamma &= \min_i |r(y_i)|. \end{aligned}$$

Wir beachten, dass die Definition von  $\alpha$  gerechtfertigt ist, da im ersten Schritt der Iteration die beste Approximation für die Referenz  $\{x_0, \dots, x_n\}$  errechnet wird. Nach dem Alternantensatz sind absolute Werte der  $r(x_i)$

alle gleich. man beachte außerdem, dass im letzten Schritt der Iteration ein Element  $y \in [a, b]$ , das  $r(y) = \|r\|_\infty$  erfüllt, in die neue Referenz von  $\{y_0, \dots, y_n\}$  aufgenommen wurde, was die Definition von  $\beta$  rechtfertigt. Die entsprechenden Mengen, die bei der nächsten Iteration erhalten werden, sind mit  $\alpha', \beta'$  und  $\gamma'$  bezeichnet. Wir definieren  $\beta^* = \|f - P^*\|_\infty$ . Nach dem Satz von de La Vallée-Poussin, ist es klar, dass  $\beta \geq \beta^*$  ist. Aus der Definition der neuen Referenz, ist es auch klar, dass  $\gamma \geq \alpha$  ist. Dies gibt uns folgendes:

$$\alpha \leq \gamma \leq \beta^* \leq \beta. \quad (17)$$

Zu Beginn der nächsten Iteration wird der Vektor  $c' = [c'_1, \dots, c'_n]^T$ , der

$$\max_i \left| f(y_i) - \sum_{j=1}^n c'_j g_j(y_i) \right|$$

minimiert, berechnet. Nach (13) im Beweis von Korollar 3.2.2 ist der Koeffizientenvektor  $c' = [c'_1, \dots, c'_n]^T$  durch Lösen des linearen Systems

$$(-1)^i h + \sum_{j=1}^n c'_j g_j(y_i) = f(y_i), \quad i = 0, \dots, n$$

,für die Unbekannten  $h$  und  $c'_1, \dots, c'_n$ , wobei  $h = r(y_0)$ , erhalten. Die Matrix dieses Systems ist nicht singulär; Im Beweis von Korollar 3.2.3 haben wir festgestellt, dass der Lösungsvektor  $c'$  existiert und eindeutig ist. Die Konstante  $h$  wird dann eindeutig durch  $c'$  bestimmt. Wir schreiben dieses System wie folgt

$$\begin{bmatrix} 1 & g_1(y_0) & \cdots & g_n(y_0) \\ -1 & g_1(y_1) & \cdots & g_n(y_1) \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^n & g_1(y_n) & \cdots & g_n(y_n) \end{bmatrix} \begin{bmatrix} h \\ c'_1 \\ \vdots \\ c'_n \end{bmatrix} = \begin{bmatrix} f(y_0) \\ f(y_1) \\ \vdots \\ f(y_n) \end{bmatrix}.$$

Als Nächstes verwenden wir die Cramer-Regel, und erhalten

$$h = \frac{\begin{bmatrix} f(y_0) & g_1(y_0) & \cdots & g_n(y_0) \\ f(y_1) & g_1(y_1) & \cdots & g_n(y_1) \\ \vdots & \vdots & \ddots & \vdots \\ f(y_n) & g_1(y_n) & \cdots & g_n(y_n) \end{bmatrix}}{\begin{bmatrix} 1 & g_1(y_0) & \cdots & g_n(y_0) \\ -1 & g_1(y_1) & \cdots & g_n(y_1) \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^n & g_1(y_n) & \cdots & g_n(y_n) \end{bmatrix}}.$$

Bezeichne die Spalte mit  $\pm 1$  durch  $M_i$ . Die Lösung für  $h$  kann dann als

$$h = \frac{\sum_{i=0}^n f(y_i) M_i (-1)^i}{\sum_{j=0}^n M_j} \quad (18)$$

geschrieben werden. Wenn  $f$  selbst in (18) durch ein generalisiertes Polynom  $P = \sum_{j=1}^n a_j g_j(x)$  ersetzt wird, ist die Approximaion an der Referenz

$\{y_0, \dots, y_n\}$  genau und daher  $h = 0$ , das heißt  $\sum_{i=0}^n f(y_i) M_i (-1)^i = 0$  ist.

Dies zeigt, dass in (18) die Approximaion  $f(x)$  durch

$$r(x) = f(x) - \sum_{j=1}^n c'_j g_j(x)$$

ersetzt werden kann, wobei  $h$  unverändert bleibt. Da  $r(y_i)$  alternierend im Vorzeichen ist, dann

$$\sum_{i=0}^n r(y_i) (-1)^i M_i = \pm \sum_{i=0}^n |r(y_i)| M_i \quad (19)$$

da  $y_0 < \dots < y_n$  und die Haar-Bedingung durch  $\{g_1, \dots, g_n\}$  erfüllt ist, sagt

Lemma 3.1.4 darüber hinaus, dass alle minors  $M_i$  dasselbe Zeichen haben.

Indem wir (18) mit (19) kombinieren, finden wir dass

$$\alpha' = |h| = \frac{\sum_{i=0}^n |M_i| |r(y_i)|}{\sum_{j=0}^n |M_j|} . \quad (20)$$

Nun lass

$$\theta_i = \frac{|M_i|}{\sum_{j=0}^n |M_j|} . \quad (21)$$

Wir beachten hier, dass  $\theta_i \in [0,1]$ . Kombiniere (20) und (21), um zu erhalten



$$\begin{aligned}
\alpha' &= \sum_{i=0}^n \theta_i |r(y_i)| \\
&\geq \sum_{i=0}^n \theta_i \min_j |r(y_i)| \\
&= \sum_{i=0}^n \theta_i \gamma \\
&= \gamma
\end{aligned} \tag{22}$$

Nehmen wir jetzt an, dass während der gesamten Iterationen des Algorithmus die Zahlen  $\theta_i$  größer als eine feste Konstante  $1-\theta > 0$  bleiben. Wir stellen die folgenden Ungleichungen fest,

$$\begin{aligned}
\gamma' - \gamma &\geq \alpha' - \gamma & \gamma' &\geq \alpha' \text{ von (22)} \\
&= \sum_{i=0}^n \theta_i (|r(y_i)| - \gamma) & \sum \theta_i &= 1 \\
&\geq \min_i \theta_i (\beta - \gamma) & \text{wobei } \beta &= \max_i |r(y_i)| \tag{23} \\
&\geq (1-\theta)(\beta - \gamma) & \text{Da } \theta_i &\geq 1-\theta \text{ für alle } i \\
&\geq (1-\theta)(\beta^* - \gamma) & \text{nach (19) } \beta &\geq \beta^*.
\end{aligned}$$

Außerdem,

$$\begin{aligned}
\beta^* - \gamma' &= (\beta^* - \gamma) - (\gamma' - \gamma) \\
&\leq (\beta^* - \gamma) - (1-\theta)(\beta^* - \gamma), \quad \gamma' - \gamma \geq (1-\theta)(\beta^* - \gamma) \\
&= \theta(\beta^* - \gamma).
\end{aligned}$$

Wir bezeichnen die Werte von  $\gamma$  und  $\beta$  in der Iteration  $k$  als  $\gamma^{(k)}$  bzw.  $\beta^{(k)}$ .

Wenn wir die Ungleichung  $k$ -mal anwenden, finden wir, dass

$$\begin{aligned}
\beta^* - \gamma^{(k)} &\leq \theta^{(k)} (\beta^* - \gamma^{(0)}) \\
&= B \theta^{(k)},
\end{aligned}$$

Wobei  $B$  ist eine nicht negative Konstante. Wir schaffen eine andere Ungleichheit.

$$\begin{aligned}
\beta^{(k)} \beta^* &\leq \beta^{(k)} \gamma^{(k)} && \text{beachten wir dass, } \gamma^{(k)} \leq \beta^* \\
&\leq \frac{1}{1-\theta} (\gamma^{(k+1)} - \gamma^{(k)}) && \text{nach (23)} \\
&\leq \frac{1}{1-\theta} (\beta^* - \gamma^{(k)}) && \text{Da } \gamma^{(k)} \leq \beta^* \\
&\leq \frac{1}{1-\theta} B \theta^k \\
&= C \theta^k && \text{mit } C \text{ nicht negative}
\end{aligned}$$

Nach dem Satz (Strong unicity) existiert eine konstante  $\gamma > 0$  so, dass

$$\|f - P^*\|_\infty + \gamma \|P^* - P^k\|_\infty \leq \|f - P^k\|_\infty.$$

Wir vervollständigen den Beweis durch die folgende Ungleichung:

$$\begin{aligned}
\|P^* - P^k\|_\infty &\leq \frac{\|f - P^k\|_\infty - \|f - P^*\|_\infty}{\gamma} \\
&= \frac{\beta^{(k)} - \beta^*}{\gamma} \\
&\leq \frac{C}{\gamma} \theta^k \\
&= A \theta^k, \quad \text{wobei } A \text{ ist nicht negative.}
\end{aligned}$$

□

## 6 Matlab-Codes

Das folgende zeigt, wie die Codes verwendet werden können, um eine lineare Approximation für  $e^x$  auf dem Intervall  $[0,1]$ .

```

f=@(x)exp(x);
a=0;
b=1;
ref=linspace(a,b,3);
basis=@(x)[1 x];
eps=0,0001;
[approx,normr]=RemezExample(f,ref,basis,a,b,eps)

```

### Ausführen des Algorithmus RemezExample.m 6.1

Berechnet die Approximation (approx) für Funktion (f) und gibt die unendliche Norm von der Fehlerfunktion (normr).

Algorithmus stoppt, wenn  $\text{delta} \leq \text{epsilon}$ .

**Input:** Approximant (f),  
Anfangsreferenz (ref),  
Basis für Haar System (basis),  
Endpunkte a, b des Intervalls [a,b],  
Epsilon für das Stoppkriterium (eps).

**Output:** Approximation für f (approx),  
die unendliche Norm von der Fehlerfunktion (normr).

```
function[approx,normr]=RemezExample(f,ref,basis,a,b,eps)
delta=eps+1
While delta>epsilon
[ymaxr,r,newref,approx,delta]=Remez2(f,ref,basis,a,b);
ref=newref;
end
normr=abs(r(ymaxr));
```

### Einzelne Iteration des Remez-Algorithmus(Remez2.m) 6.2

Führt eine Iteration des Remez-Algorithmus aus.

**Input:** Approximant (f),  
Anfangsreferenz (ref),  
Basis für Haar System (basis),  
Endpunkte a, b des Intervalls [a,b],  
**output:** Lage von max von  $|r|$  ,  
Fehlerfunktion (r),

Referenz für nächste Iteration (refnew),  
 Approximation für f (approx),  
 delta für das Stoppkriterium.

```
function[ymaxr,r,newref,approx,delta]=Remez2(f,ref,basis,a,b
)
[lamda]=minimaxsol2(f,ref,basis)
r=@(x)f(x)-dot(basis(x),lamda);
h=r(ref(1));
[ymaxr]=findmax2(r,a,b)
normr=abs(r(ymaxr));
delta=abs(normr-abs(h));
s=size(ref);
n=s(2)-1;
z(1)=a;
for i=2:n+1
    z(i)=fzero(r,[ref(i-1) ref(i)]);
end
    z(n+2)=b;
for i=1:n+1
    sigma=sign(r(ref(i)));
    newref(i)=fmaxsigmar(sigma,r,z(i),z(i+1),ref(i),a,b);
end
if a<=ymaxr && ymaxr<newref(1)
    if sign(r(ymaxr))==sign(r(newref(1)))
        newref(1)=ymaxr;
    else
        for i=n+1:-1:2
            newref(i)=newref(i-1);
        end
        newref(1)=ymaxr;
    end
elseif newref(n+1)<=ymaxr && ymaxr<=b
    if sign(r(ymaxr))==sign(r(newref(n+1)))
        newref(n+1)=ymaxr;
    else
        for i=1:n
            newref(i)=newref(i+1)
        end
        newref(n+1)=ymaxr
    end
else
    for i=2:n+1
        if newref(i-1)<=ymaxr && ymaxr<newref(i)
            if sign(r(ymaxr))==sign(r(newref(i-1)))
```

```

        newref(i-1)=ymaxr;
    elseif
        sign(r(ymaxr))==sign(r(newref(i)))
        newref(i)=ymaxr;
    end
end
end
end
end
end
end
end

```

### Lösen eines linearen Systems, um die beste Approximation auf einer Referenz zu erhalten 6.3

```

function[lamda]=minimaxsol2(f,ref,basis)
s=size(ref);
n=s(2)-1;
for i=2:(n+1)
    b(i-1)=f(ref(i))-power(-1,i-1)*f(ref(1));
end
vektor2=basis(ref(1));
for i=1:n
    vektor1=basis(ref(i+1));
    for j=1:n
        A(i,j)=vektor1(j)-power(-1,i)*vektor2(j);
    end
end
b=transpose(b);
lamda=A\b
end

```

### Suche nach x-Koordinate des Maximums von r findmaxr2.m 6.4

```

function[ymaxr]=findmax2(r,a,b)
stepsize=(b-a)/10000
% Vektor mit Funktionswerten erstellen
for i=1:10001
    valvector(i)=abs(r(a+stepsize*(i-1)));
end
maxval=max(valvector);
% Finden wir den Ort mit dem groessten Wert
for i=1:10001
    if valvector(i)==maxval
        ymaxr=a+stepsize*(i-1);
    end
end
end
end

```

## Lokale Maxima finden fmaxsigmar.m 6.5

```
function[maxsol]=fmaxsigmar(sigma,r,lb,ub,xi,a,b)
f=@(x) -1*sigma*r(x);
maxsol=fminband(f,lb,ub)
if sigma*r(xi)>sigma*r(maxsol)
    stepsize=(ub-lb)/10000;
    for i=1:10001
        valvector(i)=sigma*r(lb)+stepsize*(i-1);
    end
    maxval=max(valvector);
    for i=1:10001
        if valvector(i)==maxval
            maxsol=lb+stepsize*(i-1);
        end
    end
end
end
% Falls Sigma*r(ref(i))bereits maximal war
if sigma*r(xi)>sigma*r(maxsol)
    maxsol=xi;
end
% Testwerte an den Endpunkten von [a,b]
if lb==a && sigma*r(a)>sigma*r(maxsol)
    maxsol=a;
end
if ub== b && sigma*r(b)>sigma*r(maxsol)
    maxsol=b
end
end
end
```

## 7 Beispiele

**7.1** In diesen beiden Funktionen  $f_1 = e^x$  und  $f_2 = \sin(\frac{1}{2}\pi x)$  treten die Extrema der Fehlerfunktion nahe der Mitte und an den Endpunkten von  $[0,1]$ , und die Anzahl der verwendeten Iterationen beträgt 2.

die unendliche Norm von der Fehlerfunktion (normr).

$$\|f_1 - P_1\|_\infty = 0.105933415992$$

$$\|f_2 - P_2\|_\infty = 0.105256830566$$

Eine Darstellung der beiden Fehlerfunktionen ist in Abbildung 2 dargestellt.

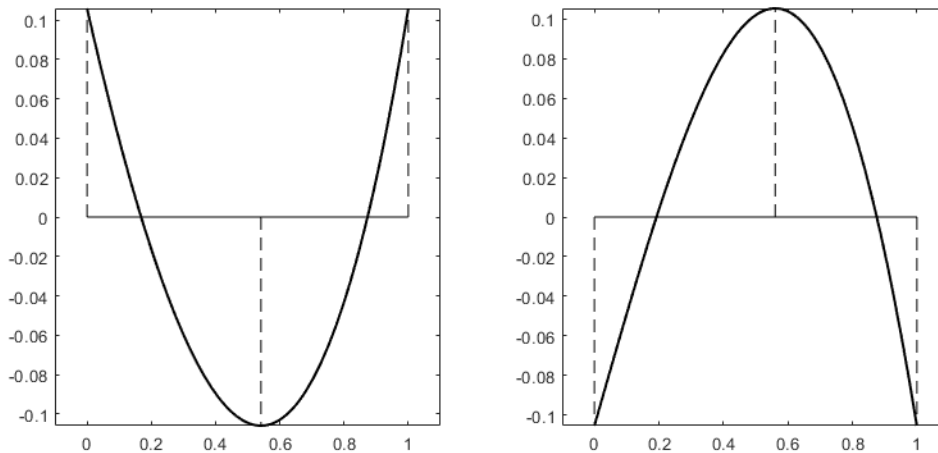


Abbildung 2: Fehlerfunktion für  $f_1 = e^x$  links und  $f_2 = \sin(\frac{1}{2}\pi x)$  rechts. Die x-Koordinaten der vertikalen Linien sind die Referenzpunkte, an denen die Approximation berechnet wurde.

**7.2** Mehrere Polynomapproximationen für  $f(x) = e^x \cos(2\pi x) \sin(2\pi x)$  diese Funktion hat mehrere Extremwerte auf dem Intervall  $[0,1]$ , deshalb wählen wir sie aus.

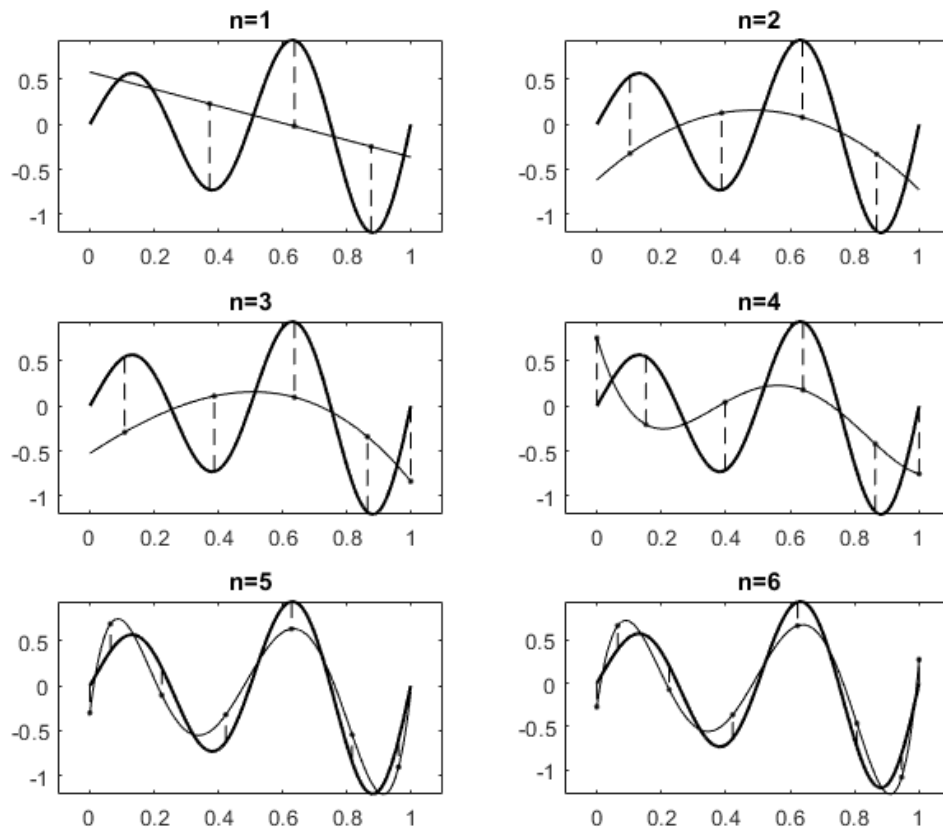


Abbildung 3: Der beste Grad  $n$  Polynomapproximation auf  $[0,1]$  für  $f(x) = e^x \cos(2\pi x) \sin(2\pi x)$ , bestimmt durch unsere Implementierung des Remez-Algorithmus. Der Algorithmus wurde angehalten, wenn  $|\delta| < 10^{-5}$ .

In der folgenden Tabelle haben Wir den Grad der Polynomapproximation bis zum Scheitern erhöht. Wir beachten, dass für  $n=18$  die berechnete Fehlerfunktion die Alterationseigenschaft möglicherweise nicht mehr erfüllt, Ferner beobachten wir aus der Tabelle, dass Polynome höheren Grades, wie zu erwarten, bessere Approximationen ergeben.



$n$	$\ f - P\ _\infty$	Iterations	$n$	$\ f - P\ _\infty$	Iterations
1	0.95484123	5	10	$4.3157657 \times 10^{-3}$	4
2	0.85490254	4	11	$1.1728234 \times 10^{-3}$	4
3	0.83717665	5	12	$2.6148995 \times 10^{-4}$	4
4	0.75385311	3	13	$7.9211438 \times 10^{-5}$	5
5	0.30308145	4	14	$1.0753948 \times 10^{-5}$	5
6	0.27180459	4	15	$3.8528423 \times 10^{-6}$	5
7	$7.6258611 \times 10^{-2}$	4	16	$3.6481047 \times 10^{-7}$	4
8	$4.5322638 \times 10^{-2}$	4	17	$1.6218161 \times 10^{-7}$	4
9	$1.1750510 \times 10^{-2}$	4	18	N/A	N/A

Tabelle 1: Normen der Fehlerfunktionen und Anzahl der Iterationen, die für Grad  $n = 1, \dots, 17$  Approximationen durch den Remez-Algorithmus an  $f(x) = e^x \cos(2\pi x) \sin(2\pi x)$  verwendet werden. Für  $n = 1, \dots, 11$  verwendeten wir  $\delta = 10^{-5}$ . Wir haben  $\delta$  nach  $n = 11$  auf  $10^{-6}$  reduziert und nach  $n = 13$  auf  $10^{-7}$  reduziert. Bei  $n = 18$  schlägt unsere Implementierung zum ersten Mal fehl.

## 8 Zusammenfassung

Wir haben zunächst den Charakterisierungssatz und dann den Alternantensatz bewiesen. Letzterer Satz spielt eine entscheidende Rolle im Mechanismus des Remez-Algorithmus, und sein Korollar wird im ersten Schritt jeder Iteration explizit verwendet, um die beste Minimax-Approximation für eine Referenz zu berechnen. Wir haben gesehen, dass in den verbleibenden Schritten einer Iteration die Referenz angepasst wird, um eine bessere Approximation bei der nächsten Iteration zu erhalten, und dann haben wir die lineare Konvergenz für den Remez-Algorithmus bewiesen. Am Ende der Fokus lag auf der Erstellung von Beispielen mit unserer Implementierung, um zu zeigen, was die Implementierung kann und was nicht. Wir haben gesehen, dass der Algorithmus für hochgradige Polynomapproximationen letztendlich aufgrund einer steigende Bedingungsnummer in der am Programm beteiligten Matrix versagt.

## Literatur

1. R.A. DeVore and G.G. Lorentz. Constructive Approximation. Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, United States of America, 1993.
2. Powell, M. J. D. (1981) Approximation theory and methods. Cambridge University Press, Cambridge-NewYork-New Rochelle-Melbourne-Sydney.
3. E.W. Cheney. Approximation Theory. Chelsea Publishing Company, United States of America, second edition, 1982.
4. N.L. Carothers. A Short Course on Approximation Theory.
5. R. Pachon and L.N. Trefethen. Barycentric-Remez algorithms for best polynomial approximation in the chebfun system.
6. H. Van de Vel. Haar intervals for trigonometric polynomials. Journal of Computational and Applied Mathematics