

Numerik WS2018/19

Dozent: Prof. Dr. Andreas Fischer

14. Oktober 2018

Inhaltsverzeichnis

I	Interpolation	2
1	Grundlagen	2
2	Interpolation durch Polynome	4
2.1	Existenz und Eindeutigkeit	4
2.2	NEWTON-Form des Interpolationspolynoms	5
2.3	Interpolationsfehler	6
3	Interpolation durch Polynomsplines	8
II	numerische Quadratur und Integration	9
1	Integration von Interpolationspolynomen	9
2	NEWTON-COTES-Formeln	10
3	spezielle NEWTON-COTES-Formeln	11
4	Zusammengesetzte NEWTON-COTES-Formeln	12
5	GAUSS'sche Quadraturformeln	13
III	direkte Verfahren für lineare Gleichungssysteme	14
1	GAUSS'scher Algorithmus für quadratische Systeme	14
2	Lineare Quadratmittelprobleme	15
3	Kondition linearer Gleichungssysteme	16
IV	Kondition von Aufgaben und Stabilität von Algorithmen	17
1	Maschinenzahlen und Rundungsfehler	17
2	Fehleranalyse	18
V	Newton-Verfahren zur Lösung nichtlinearer Gleichungssysteme	19
1	Das NEWTON-Verfahren	19
2	Gedämpftes NEWTON-Verfahren	20
VI	lineare Optimierung	21
1	Ecken und ihre Charakterisierung	21
2	Simplex-Verfahren	22
3	Die Tableauform des Simplex-Verfahrens	23
4	Revidiertes Simplex-Verfahren	24
5	Bestimmung einer ersten zulässigen Basislösung	25
	Anhang	27
A	Listen	27
A.1	Liste der Theoreme	27
A.2	Liste der benannten Sätze, Lemmata und Folgerungen	28

Vorwort

Kapitel I

Interpolation

1. Grundlagen

Aufgabe:

Gegeben sind $n + 1$ Datenpaare $(x_0, f_0), \dots, (x_n, f_n)$, alles reelle Zahlen und paarweise verschieden.

Gesucht ist eine Funktion $F : \mathbb{R} \rightarrow \mathbb{R}$, die die Interpolationsbedingungen

$$F(x_0) = f_0, \dots, F(x_n) = f_n \quad (1)$$

genügt.

Definition (Stützstellen, Stützwerte)

Die x_0 bis x_n werden Stützstellen genannt.

Die f_0 bis f_n werden Stützwerte genannt.

Die oben gestellte Aufgabe wird zum Beispiel durch

$$F(x) = \begin{cases} 0 & x \notin \{x_0, \dots, x_n\} \\ f_i & x = x_i \end{cases}$$

gelöst. Weitere Möglichkeiten sind: Polygonzug, Treppenfunktion, Polynom, ...

- In welcher Menge von Funktionen soll F liegen?
- Gibt es im gewählten Funktionenraum für beliebige Datenpaare eine Funktion F , die den Interpolationsbedingungen genügt (eine solche Funktion heißt Interpolierende)?
- Ist die Interpolierende in diesem Raum eindeutig bestimmt?
- Welche weiteren Eigenschaften besitzt die Interpolierende, zum Beispiel hinsichtlich ihrer Krümmung oder der Approximation einer Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $f_k = f(x_k)$ für $k = 0, \dots, n$?
- Wie sollte man die Stützstellen wählen, falls nicht vorgegeben?
- Wie lässt sich die Interpolierende effizient bestimmen, gegebenenfalls auch unter der Berücksichtigung, dass neue Datenpaare hinzukommen oder dass sich nur die Stützwerte ändern?

■ Beispiel 1.1

k	0	1	2	3	4	5
x_k in s	0	1	2	3	4	5
f_k in °C	80	85,8	86,4	93,6	98,3	99,1

Interpolation im

- Raum der stetigen stückweise affinen Funktionen
- Raum der Polynome höchstens 5. Grades
- Raum der Polynome höchstens 4. Grades (Interpolation im Allgemeinen nicht lösbar, Regression nötig)

2. Interpolation durch Polynome

Π_n bezeichne den Vektorraum der Polynome von Höchstgrad n mit der üblichen Addition und Skalarmultiplikation. Für jedes $p \in \Pi_n$ gibt es $a_0, \dots, a_n \in \mathbb{R}$, sodass

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (2)$$

und umgekehrt.

2.1. Existenz und Eindeutigkeit

Satz 2.1

Zu $n+1$ Datenpaaren $(x_0, f_0), \dots, (x_n, f_n)$ mit paarweise verschiedenen Stützstellen existiert genau ein Polynom $p \in \Pi_n$, dass die Interpolationsbedingung Gleichung (1) erfüllt.

Beweis. • Existenz: Sei $j \in \{0, \dots, n\}$ und $L_j : \mathbb{R} \rightarrow \mathbb{R}$ mit

$$L_j(x) := \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i} = \frac{(x - x_0) \cdots (x - x_{j-1})(x - x_{j+1}) \cdots (x - x_n)}{(x_j - x_0) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)}$$

das LAGRANGE-Basispolynom vom Grad n . Offenbar gilt $L_j \in \Pi_n$ und

$$L_j(x_k) = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} = \delta_{jk} \quad (3)$$

Definiert man $p : \mathbb{R} \rightarrow \mathbb{R}$ durch

$$p(x) := \sum_{j=0}^n f_j \cdot L_j(x) \quad (4)$$

so ist $p \in \Pi_n$ und außerdem erfüllt p wegen Gleichung (3) die Interpolationsbedingung Gleichung (1)

- Eindeutigkeit: Angenommen es gibt Interpolierende $p, \tilde{p} \in \Pi_n$ mit $p \neq \tilde{p}$. Dann folgt $p - \tilde{p} \in \Pi_n$ und $(p - \tilde{p})(x_k) = p(x_k) - \tilde{p}(x_k) = 0$ für $k = 0, \dots, n$. Also hat $(p - \tilde{p})$ mindestens $n + 1$ Nullstellen, hat aber Grad n . Das heißt, dass $(p - \tilde{p})$ das Nullpolynom sein muss. \square

Definition (Interpolationspolynom)

Das Polynom, dass die Interpolationsbedingung erfüllt, heißt Interpolationspolynom zu $(x_0, f_0), \dots, (x_n, f_n)$.

► Bemerkung 2.2

- Die Darstellung Gleichung (4) heißt LAGRANGE-Form des Interpolationspolynoms.
- Um mittels Gleichung (4) einen Funktionswert $p(x)$ zu berechnen, werden $\mathcal{O}(n^2)$ Operationen benötigt; bei gleichabständigen Stützstellen kann man diesen Aufwand auf $\mathcal{O}(n)$ verringern. Ändern sich die Stützwerte, kann man durch Wiederverwendung von den $L_j(x)$ das $p(x)$ in $\mathcal{O}(n)$ Operationen berechnen.
- Man kann zeigen, dass L_0 bis L_n eine Basis von Π_n bilden.

2.2. Newton-Form des Interpolationspolynoms

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0) \cdots (x - x_{n-1}) \quad (5)$$

mit Koeffizienten $c_0, \dots, c_n \in \mathbb{R}$. Die Berechnung des Koeffizienten c_j kann rekursiv durch Ausnutzen der Interpolationsbedingung Gleichung (1) erfolgen. Für c_0 erhält man

$$f_0 \stackrel{!}{=} p(x_0) = c_0$$

Seien c_0 bis c_{j-1} bereits ermittelt. Dann folgt:

$$f_j \stackrel{!}{=} p(x_j) = c_0 + \underbrace{\sum_{k=1}^{j-1} c_k (x_j - x_0) \cdots (x_j - x_{k-1})}_{\text{bekannt}} + c_j \underbrace{(x_j - x_0) \cdots (x_j - x_{j-1})}_{\text{unbekannt}}$$

► Bemerkung 2.3

- Der Aufwand um die Koeffizienten c_0, \dots, c_n zu ermitteln ist $\mathcal{O}(n^2)$. Kommt ein Datenpaar hinzu, kann man Gleichung (5) um einen Summanden erweitern und mit $\mathcal{O}(n)$ Operationen c_{n+1} bestimmen.
- Sind die Koeffizienten c_0, \dots, c_n in Gleichung (5) bekannt, dann benötigt man zur Berechnung von $p(x)$ $\mathcal{O}(n)$ Operationen.
- Die Polynome $N_0, \dots, N_n : \mathbb{R} \rightarrow \mathbb{R}$ mit

$$N_0 = 1 \quad \text{und} \quad N_j = (x - x_0) \cdots (x - x_{j-1})$$

heißen NEWTON-Basispolynome und bilden eine Basis von Π_n .

Die Koeffizienten c_0, \dots, c_n ergeben sich wegen Gleichung (2) auch als Lösung des folgenden linearen Gleichungssystems:

$$\begin{pmatrix} 1 & & & & \\ 1 & (x_1 - x_0) & & & \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & & \\ \vdots & \vdots & \vdots & \ddots & \\ 1 & (x_n - x_0) & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{i=0}^{n-1} (x_n - x_i) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

Die Systemmatrix dieses linearen Gleichungssystems ist eine reguläre untere Dreiecksmatrix.

Zu effizienten Berechnung eines Funktionswertes $p(x)$ nach Gleichung (5) mit gegebenen Koeffizienten

c_0, \dots, c_n kann man das HORNER-Schema anwenden. Überlegung für $n = 3$.

$$\begin{aligned} p(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2) \\ &= c_0 + (x - x_0) \left[c_1 + (x - x_1) [c_2 + (x - x_2)c_3] \right] \end{aligned}$$

Für beliebiges n liefert das den folgenden Algorithmus:

■ **Algorithmus 2.4 (Horner-Schema für Newton-Form)**

Input: $n, x, c_0, \dots, c_n, x_0, \dots, x_n$

```

1  p = c_n
2  do j = n-1, 0, -1
3    p = c_j + (x - x_j)p
4  end do
```

2.3. Interpolationsfehler

Definition (Maximum-Norm)

Die Norm

$$\|g\|_\infty := \max_{x \in [a, b]} |g(x)| \quad \text{für } g \in C[a, b]$$

definiert die Maximum-Norm in $C[a, b]$.

Satz 2.5

Sei $f \in C[a, b]$. Dann existiert zu jedem $\varepsilon > 0$ ein Polynom p_ε mit $\|f - p_\varepsilon\| \leq \varepsilon$.

Also liegt die Menge aller Polynome (beliebig hohen Grades) dicht in $C[a, b]$.

Definition 2.6 (Stützstellensystem)

Stützstellensystem : $a \leq x_0^{(n)} < \dots < x_n^{(n)} \leq b$. Weiterhin bezeichne $p_n \in \Pi_n$ das zu den Datenpaaren $(x_k^{(n)}, f(x_k^{(n)}))$ gehörende eindeutig bestimmte Interpolationspolynom.

Satz 2.7 (Satz von Faber 1914)

Zu jedem Stützstellensystem gibt es $f \in C[a, b]$, sodass (p_n) nicht gleichmäßig gegen f konvergiert. $\|p_n - f\|_\infty \rightarrow 0$ bedeutet, dass (p_n) gleichmäßig gegen f konvergiert.

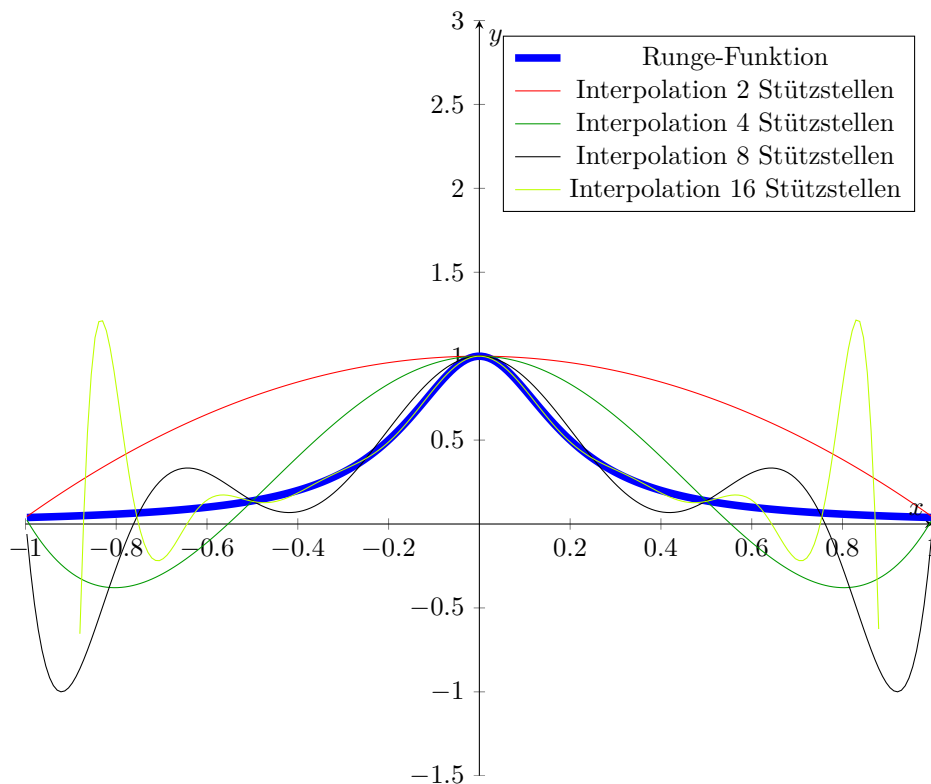
Nach einem Resultat von ERDÖS/VERTESI (1980) gilt sogar, dass $(p_n(x))$ fast überall divergiert.

■ **Beispiel 2.8 (Runge)**

$$f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = \frac{1}{1+25x^2}$$

äquidistante Stützstellen $x_0, \dots, x_n, p \in \Pi_n$ als Interpolationspolynom

Stützstellen	interpoliertes Polynom
2	$1 - \frac{25x^2}{26}$
4	$3,31565x^4 - 4,27719x^2 + 1$
8	$53,6893x^8 - 102,815x^6 + 61,3672x^4 - 13,203x^2 + 1$
16	$15403,1x^{16} - 49713,5x^{14} + 63743,8x^{12} - 41870x^{10} + 15206x^8 - 3100,35x^6 + 351,984x^4 - 22,7759x^2 + 1$

**Anmerkung**

Wer mit Mathematica selber diese Polynome berechnen will, muss folgende Befehle benutzen:

- Funktion definieren: `f[x_]:=1/(1+25x^2)`
- Interpolationspolynome ausrechnen: `Expand[InterpolatingPolynomial[Table[{i,f[i]}, {i,-1,-1,Schrittweite}],{x}]]`
- plotten: `Plot[f[x],InterpolatingPolynomial[Table[{i,f[i]}, {i,-1,-1,Schrittweite}],{x}],{x,-1,1}]`

3. Interpolation durch Polynomsplines

Kapitel II

numerische Quadratur und Integration

1. Integration von Interpolationspolynomen

2. Newton-Cotes-Formeln

3. spezielle Newton-Cotes-Formeln

4. Zusammengesetzte Newton-Cotes-Formeln

5. Gauss'sche Quadraturformeln

Kapitel III

direkte Verfahren für lineare Gleichungssysteme

1. Gauss'scher Algorithmus für quadratische Systeme

2. Lineare Quadratmittelprobleme

3. Kondition linearer Gleichungssysteme

Kapitel IV

Kondition von Aufgaben und Stabilität von Algorithmen

1. Maschinenzahlen und Rundungsfehler

2. Fehleranalyse

Kapitel V

Newton- *Verfahren zur Lösung nichtlinearer Gleichungssysteme*

1. Das Newton-Verfahren

2. Gedämpftes Newton-Verfahren

Kapitel VI

lineare Optimierung

1. Ecken und ihre Charakterisierung

2. Simplex-Verfahren

3. Die Tableauform des Simplex-Verfahrens

4. Revidiertes Simplex-Verfahren

5. Bestimmung einer ersten zulässigen Basislösung

Anhang

Anhang A: Listen

A.1. Liste der Theoreme

A.2. Liste der benannten Sätze, Lemmata und Folgerungen

Satz I.2.7:	Satz von FABER 1914	6
-------------	-------------------------------	-------------------