Analysis II für Studierende der Ingenieurwissenschaften

Reiner Lauterbach
Fachbereich Mathematik
Universität Hamburg

Technische Universität Hamburg–Harburg
Sommersemester 2005
Basierend auf der Vorlesung von
Jens Struckmeier (SS 2002)

Kapitel 7: Interpolation

7.1 Problemstellung

Gegeben: Diskrete Werte einer Funktion $f : \mathbb{R} \to \mathbb{R}$ (Stützstellen)

$$(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n) \qquad (x_0 < x_1 < \ldots < x_n)$$

Gesucht: **Einfache** Funktion $p : \mathbb{R} \to \mathbb{R}$, die die Daten **interpoliert**:

$$p(x_i) = f_i \quad i = 0, 1, \ldots, n$$

(p = Polynom, trigonometrisches Polynom, rationale Funktion).
Zunächst: Klassische Polynom–Interpolation
Bestimme ein Polynom höchstens n–ten Grades

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n,$$

das die gegebenen Daten interpoliert.

Erster Lösungsansatz: Interpolationsbedingungen ergeben ein LGS

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n &= f_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n &= f_1 \\ \vdots \qquad\quad \vdots \qquad\quad \vdots \; &= \; \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n &= f_n \end{aligned}$$

Vandermonde–Matrix

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}$$

Es gilt (Beweis mittels vollständiger Induktion)

$$\det V(x_0, \ldots, x_n) = \prod_{0 \le i < j \le n} (x_j - x_i).$$

Satz: Sind die Knoten x_j des Interpolationsproblems paarweise verschieden, d.h.

$$x_i \neq x_j \qquad (i \neq j)$$

so **existiert genau ein** Polynom p_n höchstens n–ten Grades, das die Interpolationsbedingungen

$$p_n(x_j) = f_j \qquad j = 0, 1, \ldots, n$$

erfüllt.

Aber:
LGS mit Vandermonde–Matrix ist **numerisch** zu kompliziert!

7.2 Interpolationsformeln nach Lagrange und Newton

Interpolation nach Lagrange:

Wir definieren

$$\begin{aligned} l_j(x) &= \frac{(x-x_0)\cdot\ldots\cdot(x-x_{j-1})\cdot(x-x_{j+1})\cdot\ldots\cdot(x-x_n)}{(x_j-x_0)\cdot\ldots\cdot(x_j-x_{j-1})\cdot(x_j-x_{j+1})\cdot\ldots\cdot(x_j-x_n)} \\ &= \prod_{i=0,i\neq j}^{n}\frac{(x-x_i)}{(x_j-x_i)}. \end{aligned}$$

Dann gilt für alle $j = 0, \ldots, n$:

$$l_j(x_i) = \begin{cases} 1 & : \quad i = j \\ 0 & : \quad i \neq j \end{cases} .$$

Also wird die Interpolationsaufgabe gelöst durch das Polynom $p_n(x)$

$$p_n(x) := f_0 l_0(x) + \ldots + f_n l_n(x) = \sum_{i=0}^{n} f_i l_i(x)$$

Beispiel: Wir betrachten die Daten

x_j	0	1	2	3
f_j	0	0	4	18

Dann gilt:

$$\begin{aligned} l_0(x) &= \frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} \quad l_1(x) = \frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} \\ l_2(x) &= \frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} \quad l_3(x) = \frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)} \end{aligned}$$

Das Polynom ist dann:

$$\begin{aligned} p_n(x) &= 4\cdot l_2(x) + 18\cdot l_3(x) \\ &= -4\,\frac{x(x-1)(x-3)}{2} + 18\,\frac{x(x-1)(x-2)}{6} \\ &= x^3 - x^2 \end{aligned}$$

Interpolation in der Newton–Darstellung:

Das Interpolationsproblem wird auch gelöst durch das Newton–Polynom

$$\begin{aligned} p_n(x) &= \sum_{i=0}^{n} c_i \prod_{j=0}^{i-1}(x - x_j) \\ &= c_0 + c_1(x - x_0) + \ldots + c_n(x - x_0)\ldots(x - x_{n-1}) \end{aligned}$$

mit geeigneten Koeffizienten $c_0, c_1, \ldots, c_n$.

Insbesondere gilt dann:

$$\begin{aligned} p_n(x_0) &= c_0 \\ p_n(x_1) &= c_0 + c_1(x_1 - x_0) \\ p_n(x_2) &= c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) \end{aligned}$$

Was sind die Koeffizienten $c_0, c_1, \ldots, c_n$?

$$\begin{aligned} p_n(x_0) &= c_0 \stackrel{!}{=} f_0 \quad \Rightarrow \quad c_0 = f_0 \\ p_n(x_1) &= c_0 + c_1(x_1 - x_0) \stackrel{!}{=} f_1 \quad \Rightarrow \quad c_1 = \frac{f_0}{x_1 - x_0} \\ &\vdots = \vdots \\ p_n(x_n) &= \sum_{i=0}^{n} c_i \prod_{j=0}^{i-1}(x_n - x_j) \stackrel{!}{=} f_n \\ &= c_0 + c_1(x_n - x_0) + \ldots + c_n(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1}) \\ \Rightarrow \quad c_n &= \frac{1}{\prod_{j=0}^{n-1}(x_n - x_j)} \left(f_n - \sum_{j=0}^{n-1} c_j \prod_{i=0}^{j-1}(x_n - x_i) \right) \end{aligned}$$

Folgerungen:

1) Zur Berechnung von c_j benötigt man nur die ersten $(j+1)$ Punkte $(x_0, f_0), \ldots, (x_j, f_j)$.
   Notation:

$$c_j = f[x_0, x_1, \ldots, x_{j-1}, x_j] \qquad (j = 0, 1, \ldots, n)$$

2) Nimmt man eine Stützstelle (x_{n+1}, f_{n+1}) hinzu, so gilt:

$$p_{n+1}(x) = p_n(x) + c_{n+1} \prod_{i=0}^{n} (x - x_i)$$

   mit

$$c_{n+1} = \frac{f_{n+1} - p_n(x_{n+1})}{\prod_{j=0}^{n} (x_{n+1} - x_j)}$$

Methoden zur Polynomwertberechnung:

Auswertung eines gegebenen Polynoms über das **Horner–Schema:**

Sei $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$.

Man berechnet $p(\xi)$ an einer festen Stelle ξ mittels

$$p(\xi) = a_0 + \xi(a_1 + \xi(a_2 + \xi(\ldots(a_{n-1} + a_n \xi)\ldots).$$

Rekursive Darstellung:

$$\begin{aligned} b_{n-1} &= a_n \\ b_k &= a_{k+1} + \xi b_{k+1}, \quad k = n-2, \ldots, 1, 0 \\ b_{-1} &= a_0 + \xi b_0 \end{aligned}$$

Dann gilt $p(\xi) = b_{-1}$.

(Effiziente) Berechnung des Newton–Polynoms über

Methode der dividierten Differenzen.

Satz: Definiert man die sogenannten dividierten Differenzen mittels

$$\begin{aligned} f[x_j] &= f_j \qquad j = i, i+1, \ldots, i+k \\ f[x_i, x_{i+1}, \ldots, x_{i+k}] &= \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i} \end{aligned}$$

so gilt

$$c_j = f[x_0, x_1, \ldots, x_j],$$

d.h. die dividierten Differenzen ergeben gerade die Koeffizienten des Newton–Polynoms.

Berechnungsschema zu den dividierten Differenzen (für $n = 3$):

x	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	f_0			
x_1	f_1	$f[x_0, x_1]$		
x_2	f_2	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
x_3	f_3	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$

wobei

$$\begin{aligned} f[x_j] &= f_j \qquad j = i, i+1, \ldots, i+k \\ f[x_i, x_{i+1}, \ldots, x_{i+k}] &= \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i} \end{aligned}$$

Der Interpolationsfehler:

$$\begin{aligned}\varepsilon(x) &= f(x) - p_n(x) \\ &= f(x) - \left(p_{n+1}(x) - c_{n+1} \prod_{i=0}^{n} (x - x_i) \right) \\ &= f[x_0, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i)\end{aligned}$$

Satz: Sei $f \in C^{n+1}([a,b])$. Dann gibt es ein $\xi \in [a,b]$ mit

$$f[x_0, \ldots, x_{n+1}] = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \; .$$

Abschätzung für den Interpolationsfehler:

$$|f(x) - p_n(x)| \leq \frac{1}{(n+1)!} \max_{\xi \in [a,b]} |f^{(n+1)}(\xi)| \cdot \left| \prod_{i=0}^{n} (x - x_i) \right|$$

Das Knotenpolynom:

Ein Term des Interpolationsfehlers ist das **Knotenpolynom**:

$$\omega(x) = \prod_{i=0}^{n} (x - x_i).$$

Optimierungsproblem:

Bestimme die Knoten $x_0, x_1, \ldots, x_n$, so dass

$$\max_{x_0, \ldots, x_n \in [a,b]} \left| \prod_{i=0}^{n} (x - x_i) \right|$$

für $x \in [a,b]$ minimal ist.

Lösung: Tschebyscheff–Knoten auf $[-1,1]$:

$$x_j = \cos\left(\frac{2j+1}{2n+2} \, \pi \right) \quad j = 0, 1, \ldots, n.$$

6.3 Spline–Interpolation

Sei Δ_n eine Unterteilung des Intervalls $[a, b]$:

$$\Delta_n \quad : \quad a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b$$

mit Teilintervallen $[x_{j-1}, x_j]$, $j = 1, \ldots, n$.

Definition: (Kubischer Spline)
Eine Funktion $S : [a, b] \to \mathbb{R}$ heißt **kubischer Spline**, falls

1) $S \in C^2([a, b])$, d.h. S ist zweimal stetig differenzierbar,

2) S ist auf jedem Teilintervall ein Polynom dritten Grades:

   $$S(x) = s_j(x) = a_j + b_j(x - x_{j-1}) + c_j(x - x_{j-1})^2 + d_j(x - x_{j-1})^3$$

   für $x \in [x_{j-1}, x_j]$ und $j = 1, \ldots, n$.

Interpolation der Daten $(x_0, f_0), \ldots, (x_n, f_n)$ mit kubischen Splines.

Ein kubischer Spline besitzt also $4n$ Parameter, die folgendermaßen bestimmt werden:

1) Interpolationseigenschaft:

   $$s_j(x_{j-1}) = f_{j-1}, \quad s_j(x_j) = f_j \qquad \forall\, j = 1, \ldots, n$$

2) Stetigkeit der Ableitung:

   $$s_j'(x_j) = s_{j+1}'(x_j) \qquad \forall\, j = 1, \ldots, n-1$$

3) Stetigkeit der zweiten Ableitung:

   $$s_j''(x_j) = s_{j+1}''(x_j) \qquad \forall\, j = 1, \ldots, n-1.$$

Dies ergibt $(4n - 2)$ Gleichungen, es fehlen also noch zwei Bedingungen (um $4n$ Paramter zu berechnen).

Definition: Ein kubischer Spline heißt

1) **natürlicher Spline**, falls gilt: $S''(a) = S''(b) = 0$,
2) **periodischer Spline**, falls gilt: $S^{(i)}(a) = S^{(i)}(b),\ i = 0, 1, 2$,
3) **allgemeiner Spline**, falls (z.B.) gilt: $S'(a) = f'_0,\ S'(b) = f'_n$.

Alle drei Bedingungen ergeben die zwei zusätzliche Gleichungen.

Besondere Eigenschaft des natürlichen Splines:

Satz: Unter allen interpolierenden C^2–Funktionen minimiert der **natürliche kubische Spline** das Funktional

$$I[y] := \int_a^b (y''(x))^2 dx \quad .$$

Dieses Funktional ist ein Maß für die Krümmung der Interpolationsfunktion.

Berechnung des natürlichen kubischen Splines:

Sei S auf dem Teilintervall $[x_{j-1}, x_j]$ gegeben durch

$$S(x) = a_j + b_j(x - x_{j-1}) + c_j(x - x_{j-1})^2 + d_j(x - x_{j-1})^3 \ .$$

Dann gilt:

$$\begin{aligned} a_j &= f_{j-1} \\ b_j &= \frac{f_j - f_{j-1}}{h_j} - \frac{2M_{j-1} + M_j}{6} h_j \\ c_j &= \frac{M_{j-1}}{2} \\ d_j &= \frac{M_j - M_{j-1}}{6h_j} \end{aligned}$$

und die **Momente** M_j lösen ein LGS mit Tridiagonalmatrix.

Herleitung des Splines über die Momente M_j:

$$M_j := S''(x_j), \quad j = 0, \ldots, n$$

nennt man **Momentenmethode:** $s_j''(x)$ ist eine Gerade mit

$$s_j''(x) = M_{j-1} + \frac{M_j - M_{j-1}}{h_j}(x - x_{j-1}) \qquad (h_j = x_j - x_{j-1})$$

Integration liefert

$$s_j'(x) = B_j + M_{j-1}(x - x_{j-1}) + \frac{M_j - M_{j-1}}{2h_j}(x - x_{j-1})^2$$

$$s_j(x) = A_j + B_j(x - x_{j-1}) + \frac{M_{j-1}}{2}(x - x_{j-1})^2 + \frac{M_j - M_{j-1}}{6h_j}(x - x_{j-1})^3$$

mit den Integrationskonstanten A_j, B_j.

Aus

$$s_j(x_{j-1}) = f_{j-1} \qquad s_j(x_j) = f_j$$

folgt direkt

$$A_j = f_{j-1} \qquad B_j = \frac{f_j - f_{j-1}}{h_j} - \frac{h_j}{6}(M_j + 2M_{j-1})\,. \qquad (*)$$

Stetigkeit von S' an den Punkten x_j, $j = 1, \ldots, n-1$ ergibt

$$s_j'(x_j) = s_{j+1}'(x_j) \quad \Rightarrow \quad B_j + \frac{M_j + M_{j-1}}{2}h_j = B_{j+1}\,.$$

Einsetzen von $(*)$ ergibt dann:

$$\frac{h_j}{6}M_{j-1} + \frac{h_j + h_{j+1}}{3}M_j + \frac{h_{j+1}}{6}M_{j+1} = \frac{f_{j+1} - f_j}{h_{j+1}} - \frac{f_j - f_{j-1}}{h_j}$$

für $j = 1, \ldots, n-1$.

Tridiagonalsystem für $M_0, \ldots, M_n$ mit $S''(a) = S''(b) = 0$ lautet

$$\begin{pmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}$$

mit $h_j = x_j - x_{j-1}$ und

$$\lambda_j = \frac{h_{j+1}}{h_j + h_{j+1}}, \quad \mu_j = 1 - \lambda_j$$

$$d_j = \frac{6}{h_j + h_{j+1}} \left(\frac{f_{j+1} - f_j}{h_{j+1}} - \frac{f_j - f_{j-1}}{h_j} \right)$$

für $j = 1, \ldots, n-1$ sowie den Randwerten

$$\lambda_0 = 0, \quad d_0 = 0, \quad \mu_n = 0, \quad d_n = 0 \; .$$

Abschließende Bemerkungen:

1) Die Berechnung des (natürlichen) kubischen Splines ist relativ leicht, da das Tridiagonalsystem direkt gelöst werden kann.
2) Ein Spline oszilliert weit weniger als klassische Interpolationspolynome, denn er minimiert die Krümmung.
3) Nimmt man immer mehr Stützstellen hinzu, so konvergiert der Spline mit der Ordnung h^4 gegen die Ausgangsfunktion.
4) Schreibt man als Randbedingungen die ersten Ableitungen vor, d.h.

   $$S'(a) = f'_0, \qquad S'(b) = f'_n$$

   so erhält man ebenfalls ein Tridiagonalsystem, allerdings mit anderen Werten für λ_0, d_0, μ_n und d_n.
5) Beim periodischen Spline erhält man bei der Lösung **kein** Tridiagonalsystem.