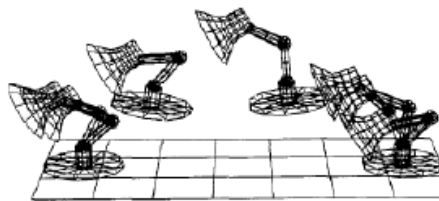


Angewandte Mathematik Interpolation

Univ.-Prof. Dr. Matthias Harders
Dr. Marcel Ritter
Sommersemester 2022



Einführungsfilm

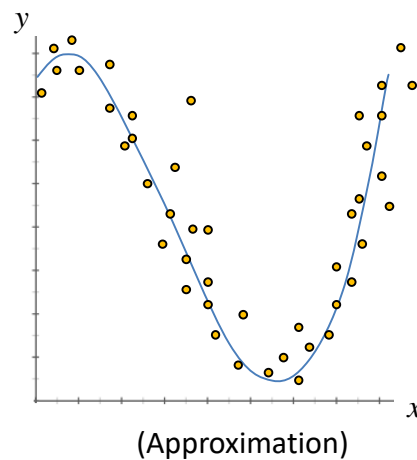
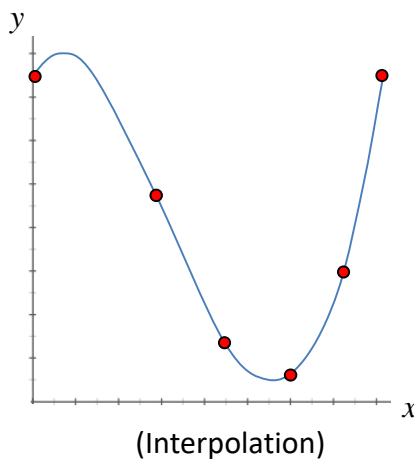


Inhalt

- Einführung
- Interpolation
- Spline-Interpolation
- Mehrdimensionale Interpolation
- Baryzentrische Interpolation
- Approximation

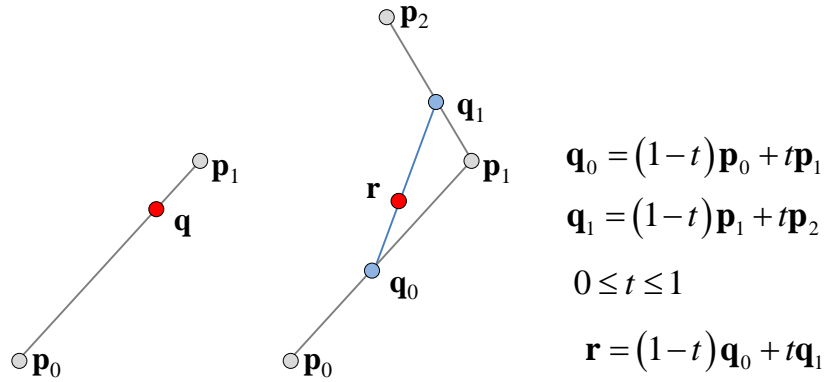


Motivation



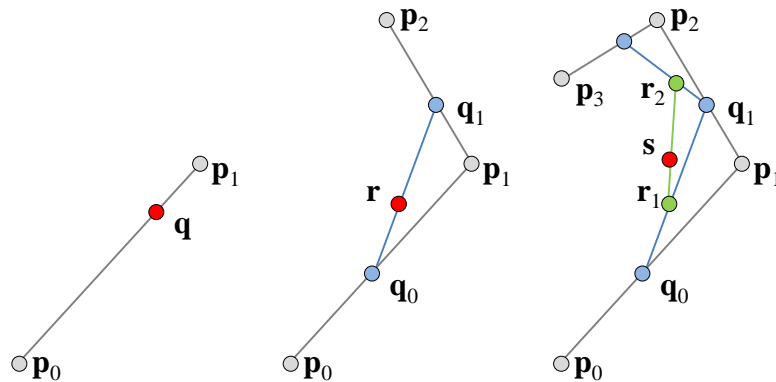
Motivation

▪ Bézierkurven



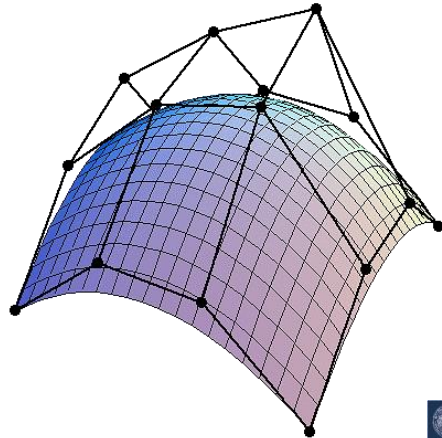
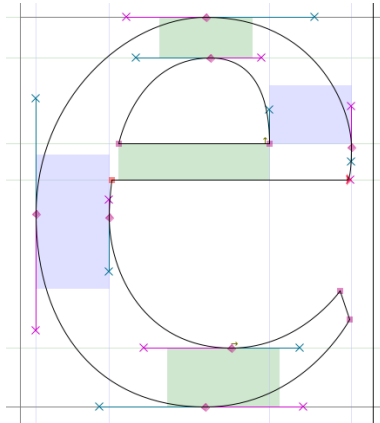
Motivation

▪ Bézierkurven



Motivation

- Bézierkurven



Problemstellung

- Gegeben seien $n+1$ voneinander verschiedene reelle Stützstellen x_0, x_1, \dots, x_n , sowie zugehörige Stützwerte y_0, y_1, \dots, y_n
- Die Wertepaare (x_i, y_i) können z.B. Messdaten sein
- Gesucht ist eine (interpolierende) Funktion f , welche die Interpolationsbedingungen erfüllt

$$f(x_i) = y_i \quad i = 0, \dots, n$$
- Als Funktionen werden z.B. Polynome, rationale Funktionen, trigonometrische Funktionen verwendet
- Vorerst Fokus auf skalare Funktionen $f: \mathbb{R} \rightarrow \mathbb{R}$

Polynominterpolation

- P_n ist die Menge aller Polynome über \mathbb{R} , mit einem Grad kleiner oder gleich $n \in \mathbb{N}$

$$P_n = \left\{ p(x) = \sum_{i=0}^n a_i x^i, a_i \in \mathbb{R} \right\}$$
- P_n bildet einen n -dimensionalen Vektorraum, mit den Monomen $1, x, x^2, \dots, x^n$ als Basis
- In der Polynominterpolation ist das Ziel ein Polynom $p \in P_n$ zu finden, welches $p(x_i) = y_i$ für alle i erfüllt
- Die Stützstellen seien geordnet: $x_0 < x_1 < \dots < x_n$
- Gesucht sind die Koeffizienten a_i



Polynominterpolation

- Wir erhalten ein lineares Gleichungssystem

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

(Vandermonde-Matrix)

- Für paarweise verschiedene x_i ist dieses Problem eindeutig lösbar
- Die Matrix ist allgemein aber schlecht konditioniert, so dass alternative Ansätze vorgezogen werden



Lagrange-Interpolation

- Für die paarweise verschiedenen x_i erhalten wir $n+1$ Lagrange-Polynome vom Grad n , für $i = 0, \dots, n$

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

- Für diese gilt insbesondere:

$$l_i^{[n]}(x_i) = 1 \quad l_i^{[n]}(x_j) = 0, \quad j \neq i$$

- Das Interpolations-Polynom vom Grad n ist eindeutig und gegeben als:

$$p(x) = \sum_{i=0}^n y_i \cdot l_i^{[n]}(x)$$



Lagrange-Interpolation

- Beispiel: lineare Lagrange-Interpolation ($n = 1$)

$$y = y_0 + t(y_1 - y_0)$$

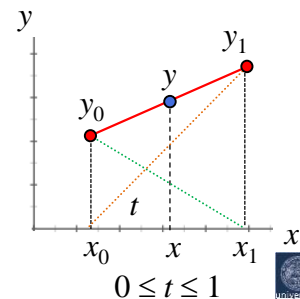
$$= y_0 + \frac{x - x_0}{x_1 - x_0}(y_1 - y_0)$$

$$= \left(\frac{x_1 - x_0}{x_1 - x_0} - \frac{x - x_0}{x_1 - x_0} \right) y_0 + \frac{x - x_0}{x_1 - x_0} y_1$$

$$= \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1$$

$$= l_0^{[1]} \cdot y_0 + l_1^{[1]} \cdot y_1$$

$$t = \frac{x - x_0}{x_1 - x_0}$$



Lagrange-Interpolation

- Beispiel: Interpolations-Polynom für die (bekannte) Funktion $g(x) = x^2$, mit Wertepaaren $(1,1)$, $(2,4)$, $(3,9)$

$$\begin{aligned}
 p(x) &= l_0^{[2]} \cdot y_0 + l_1^{[2]} \cdot y_1 + l_2^{[2]} \cdot y_2 \\
 &= \frac{(x-2)(x-3)}{(1-2)(1-3)} \cdot 1 + \frac{(x-1)(x-3)}{(2-1)(2-3)} \cdot 4 + \frac{(x-1)(x-2)}{(3-1)(3-2)} \cdot 9 \\
 &= \frac{1}{2}(x^2 - 5x + 6) - 4(x^2 - 4x + 3) + \frac{9}{2}(x^2 - 3x + 2) \\
 &= x^2
 \end{aligned}$$

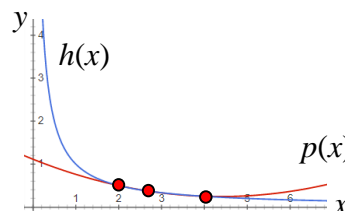


Lagrange-Interpolation

- Beispiel: Interpolations-Polynom für die (bekannte) Funktion $h(x) = \frac{1}{x}$, mit Wertepaaren $(2, \frac{1}{2})$, $(\frac{11}{4}, \frac{4}{11})$, $(4, \frac{1}{4})$

$$\begin{aligned}
 p(x) &= l_0^{[2]} \cdot y_0 + l_1^{[2]} \cdot y_1 + l_2^{[2]} \cdot y_2 \\
 &= \frac{(x-2.75)(x-4)}{(2-2.75)(2-4)} \cdot \frac{1}{2} + \frac{(x-2)(x-4)}{(2.75-2)(2.75-4)} \cdot \frac{4}{11} + \frac{(x-2)(x-2.75)}{(4-2)(4-2.75)} \cdot \frac{1}{4} \\
 &= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}
 \end{aligned}$$

$$p(3) \approx 0.32955 \approx 0.3333... = h(3)$$



Interpolationsfehler

- Es sei $g(x)$ auf $[a,b]$ $n+1$ -mal stetig differenzierbar mit $a \leq x_0 < x_1 < \dots < x_n \leq b$, dann gibt es zu jedem $x \in [a,b]$ ein $\xi \in (a,b)$ mit entsprechendem Fehler

$$e(x) = g(x) - p(x) = \frac{g^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\cdots(x-x_n)$$

- Hieraus wird ersichtlich, dass der Fehler Null beträgt bei der Interpolation eines Polynoms vom Grad n
- Für die Berechnung des Fehlers wird die Ableitung der (eigentlich unbekannten) Funktion g benötigt



Hermite-Interpolation

- Bei der Hermite-Interpolation werden neben den Funktions- auch die Ableitungswerte herangezogen
- Es sei $g(x)$ auf $[a,b]$ n -mal stetig differenzierbar mit $a \leq x_0 \leq x_1 \leq \dots \leq x_n \leq b$; wobei Stützstellen x_i m_i -mal in den Interpolationsbedingungen (in Ableitungen) auftreten können

- Für ein Interpolations-Polynom muss gelten:

$$g^{(k)}(x_i) = p^{(k)}(x_i), \quad 0 \leq i \leq n, \quad 0 \leq k \leq m_i - 1$$

- Bestimmung des Polynoms z.B. über das Schema der dividierten Differenzen



Hermite-Interpolation – Beispiel

- In der Computer Grafik wird häufig die *smoothstep*-Funktion $s(x)$ im Intervall $[0,1]$ verwendet; für sie gilt

$$s(0)=0, s(1)=1, s'(0)=0, s'(1)=0,$$

- Annahme: es sei $s(x) = a_3x^3 + a_2x^2 + a_1x + a_0$, somit $s'(x) = 3a_3x^2 + 2a_2x + a_1$

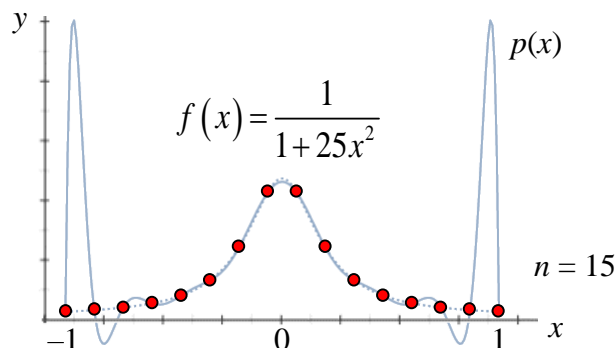
- Bestimmung der Koeffizienten (Einsetzen in s und s'):

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \\ 0 \\ 0 \end{bmatrix} \Rightarrow s(x) = 3x^2 - 2x^3$$



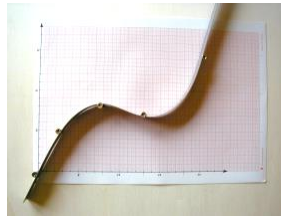
Runges Phänomen

- Bei äquidistanten Stützstellen können bei der Polynominterpolation an den Rändern große Fehler auftreten
- Mögliche Alternative: Spline-Interpolation



Hintergrund

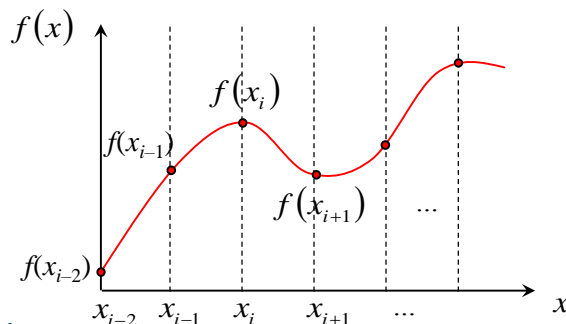
- Bisher: globale Interpolation mit einem einzelnen Polynom, mit Grad n gleich Anzahl Stützstellen $- 1$
- Idee: stückweise Interpolation auf Intervallen, mit mehreren Polynomen niedrigen Grades (Splines)
- Namensgebend sind biegsame Straklatten (engl.: *splines*), die z.B. im Schiffbau verwendet werden



Stückweise Interpolation

- Näherung der Funktion $f(x)$ durch Linearkombination von stückweisen Basisfunktionen $N(x)$

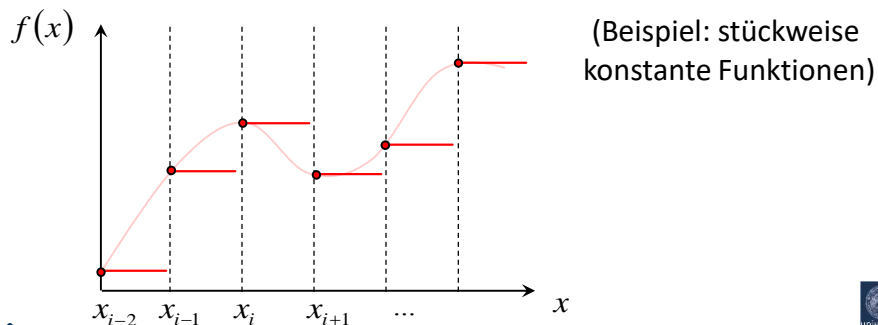
$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



Stückweise Interpolation

- Näherung der Funktion $f(x)$ durch Linearkombination von stückweisen Basisfunktionen $N(x)$

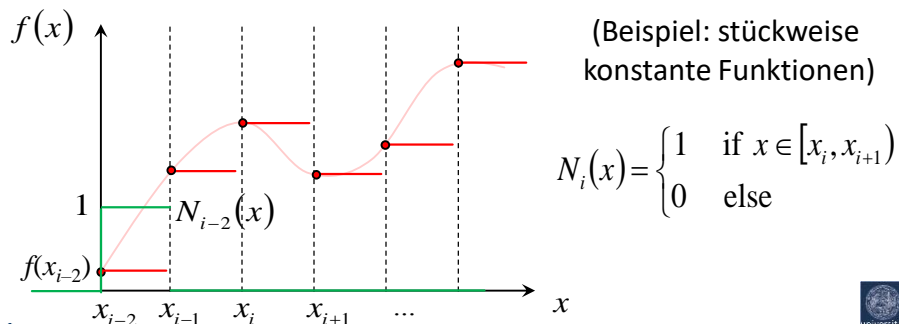
$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



Stückweise Interpolation

- Näherung der Funktion $f(x)$ durch Linearkombination von stückweisen Basisfunktionen $N(x)$

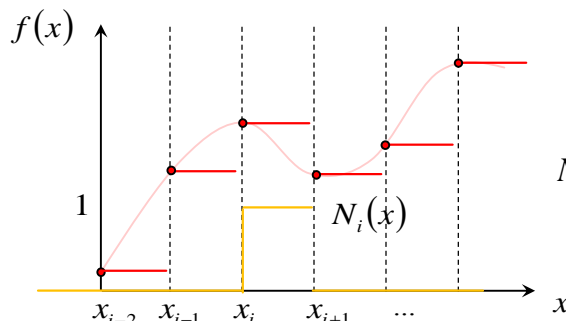
$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



Stückweise Interpolation

- Näherung der Funktion $f(x)$ durch Linearkombination von stückweisen Basisfunktionen $N(x)$

$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



(Beispiel: stückweise konstante Funktionen)

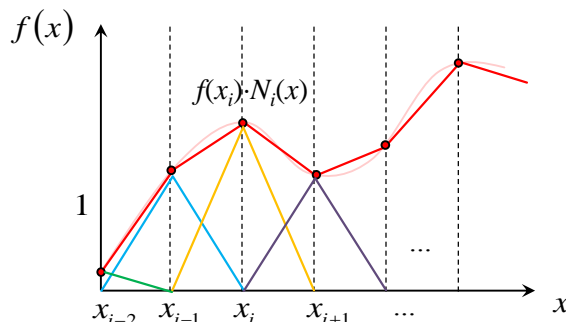
$$N_i(x) = \begin{cases} 1 & \text{if } x \in [x_i, x_{i+1}) \\ 0 & \text{else} \end{cases}$$



Stückweise Interpolation

- Näherung der Funktion $f(x)$ durch Linearkombination von stückweisen Basisfunktionen $N(x)$

$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



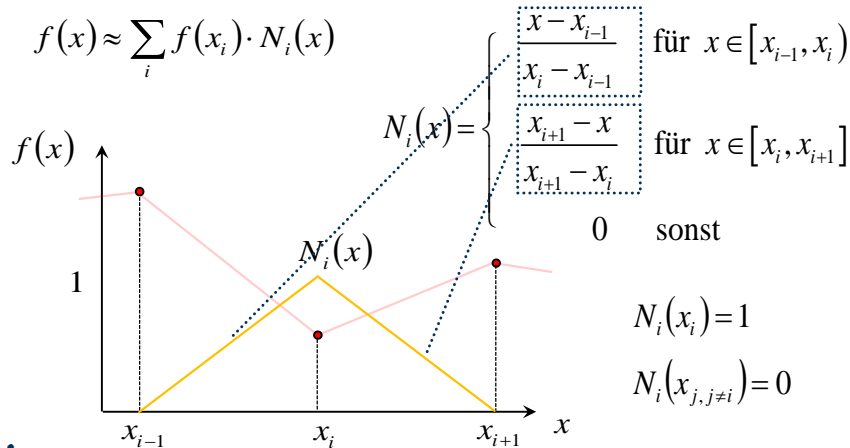
(Beispiel: stückweise lineare Funktionen)



Lineare Basisfunktionen

■ Stückweise lineare Basisfunktionen in 1D

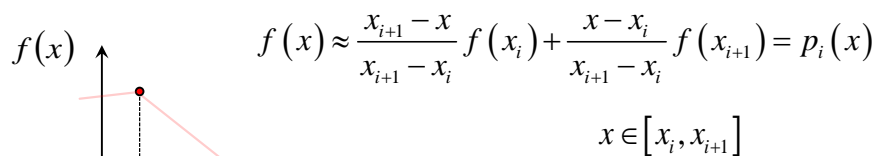
$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



Lineare Basisfunktionen

■ Stückweise lineare Basisfunktionen in 1D

$$f(x) \approx \sum_i f(x_i) \cdot N_i(x)$$



Kubische Splines

- Bei Polynomen höheren Grades (z.B. $k = 3$) werden an den Stützstellen zusätzliche Bedingungen eingeführt

- Es gibt $2n$ Interpolationsbedingungen

$$p_i(x_i) = f(x_i) \quad p_i(x_{i+1}) = f(x_{i+1}) \quad i = 0, \dots, n-1$$

- Zusätzlich $2(n-1)$ Bedingungen für die Stetigkeit der 1. und 2. Ableitungen an den inneren Stützstellen

$$p'_{i-1}(x_i) = p'_i(x_i) \quad p''_{i-1}(x_i) = p''_i(x_i) \quad i = 1, \dots, n-1$$

- Bei kubischen Splines gibt es je Polynom vier Freiheitsgrade (d.h. die Koeffizienten $a_{j,i}$)

$$p_i(x) = a_{3,i}(x-x_i)^3 + a_{2,i}(x-x_i)^2 + a_{1,i}(x-x_i) + a_{0,i}$$



Kubische Splines

- Für $4n$ Freiheitsgrade bisher nur $4n-2$ Bedingungen

- Verschiedene Optionen für die fehlenden zwei Bedingungen, z.B.:

– Freier Rand: $p''_0(x_0) = 0 \quad p''_{n-1}(x_n) = 0$
(keine Krümmung)

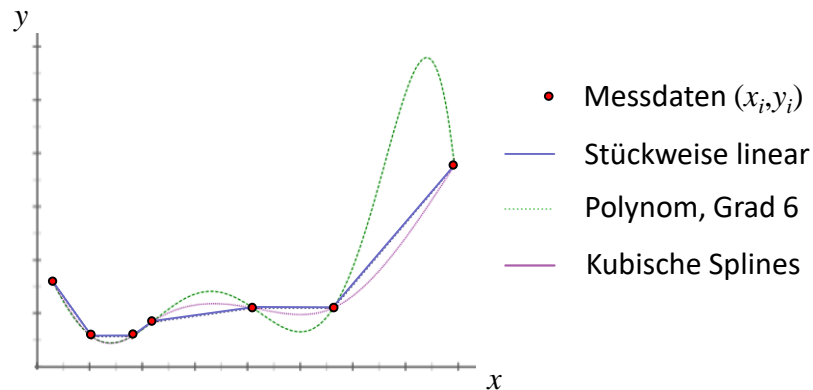
– Hermite-Bedingung: $p'_0(x_0) = f'(x_0) \quad p'_{n-1}(x_n) = f'(x_n)$

- Letztendlich ergibt sich ein lineares Gleichungssystem, mit einer Tridiagonalmatrix \mathbf{A} , welches für die unbekannten Koeffizienten gelöst werden kann



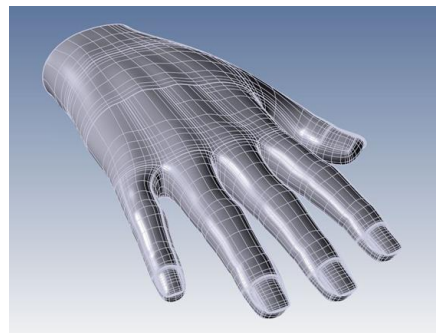
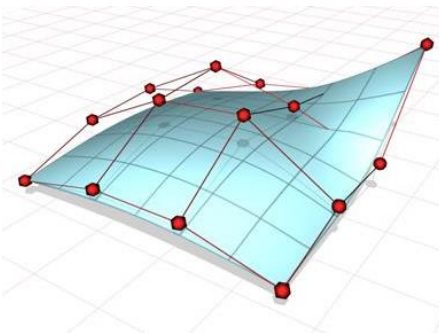
Vergleich

- Kubische Splines liefern möglichst „glatte“ Interpolation



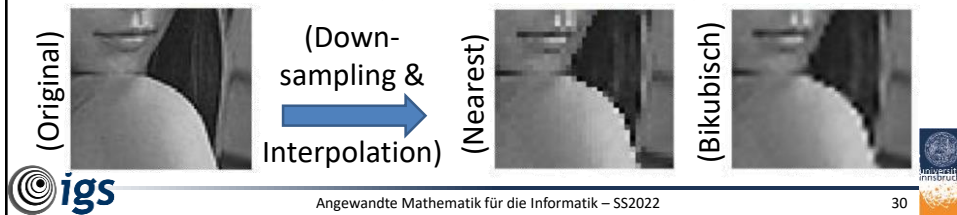
Anwendungsbeispiel

- 3D-Formbeschreibung in der Computer Grafik über Non-Uniform Rational B-Splines (NURBS)



Mehrdimensionale Interpolation

- Die vorgestellten Konzepte der Interpolation lassen sich auf multivariate, skalare Funktionen in höheren Dimensionen erweitern
- Im \mathbb{R}^2 findet z.B. die bilineare und die bikubische, sowie im \mathbb{R}^3 die trilineare Interpolation häufig Anwendung
- Bei diesen wird häufig von einem regulärem Gitter ausgegangen (Beispiel: Bildverarbeitung)

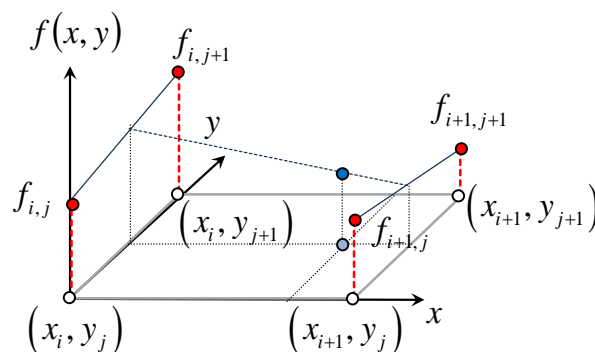


Angewandte Mathematik für die Informatik – SS2022

30

Bilineare Interpolation

- Interpolation anhand bekannter Werte $f(x_i, y_j) = f_{ij}$ an vier Stützstellen (x_i, y_j) im \mathbb{R}^2



Bilineare Interpolation

- Vereinfachung: Interpolation auf $[0,1]^2$

$$f(x, y) = ((1-y) \cdot f_{00} + y \cdot f_{01}) \cdot (1-x) + ((1-y) \cdot f_{10} + y \cdot f_{11}) \cdot x$$

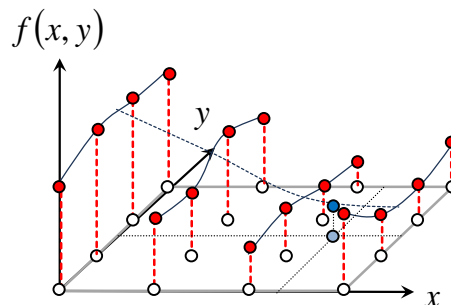
$$= \begin{bmatrix} 1-x \\ x \end{bmatrix}^T \begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

- Beachte: Interpolant nicht mehr linear (als Produkt von zwei linearen Funktionen)
- Interpolierte Werte liegen in $[\min(f_{i,j}), \max(f_{i,j})]$
- Verwendung z.B. für 2D Textur-Filterung
- Analog in 3D: trilineare Interpolation



Bikubische Interpolation

- Interpolation anhand bekannter Werte $f(x_i, y_j) = f_{ij}$ an sechzehn (4×4) Stützstellen (x_i, y_j) im \mathbb{R}^2
- Bei bikubischer Interpolation kann ein neuer Wert evtl. außerhalb des Intervalls $[\min(f_{i,j}), \max(f_{i,j})]$ liegen



Baryzentrische Koordinaten

- Ursprünglich entwickelt von A. Möbius (1827)
- Für nicht kollineare Punkte $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}$, $\mathbf{p}_i \neq \mathbf{p}_j$, sowie zu diesen zugehörige Gewichte $\lambda_0, \lambda_1, \dots, \lambda_{n-1}$, ist ein Punkt \mathbf{q} deren Baryzentrum falls

$$\mathbf{q} = \frac{\lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + \dots + \lambda_{n-1} \mathbf{p}_{n-1}}{\lambda_0 + \lambda_1 + \dots + \lambda_{n-1}}$$

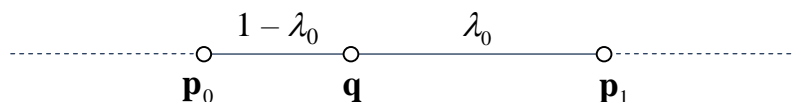
- Die Gewichte λ_i sind die baryzentrischen Koordinaten von \mathbf{q} , in Bezug auf die gegebenen $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}$
- Diese sind vorerst nicht eindeutig: $k\lambda_0, k\lambda_1, \dots, k\lambda_{n-1}$, $k \neq 0$ sind ebenso baryzentrische Koordinaten von \mathbf{q}



Baryzentrische Koordinaten

- Üblicherweise werden baryzentrische Koordinaten zusätzlich normalisiert: $\lambda_0 + \lambda_1 + \dots + \lambda_{n-1} = 1$
- Insbesondere gilt dann, dass $\mathbf{q} = \mathbf{p}_i$ für $\lambda_i = 1$, $\lambda_{j \neq i} = 0$
- Beispiel: normalisierte baryzentrische Koordinaten für ein Liniensegment ($n = 2$)

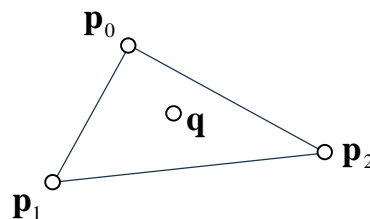
$$\mathbf{q} = \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 = \lambda_0 \mathbf{p}_0 + (1 - \lambda_0) \mathbf{p}_1 \quad \lambda_0 + \lambda_1 = 1$$



Baryzentrische Koordinaten in Dreiecken

- Normalisierte baryzentrische Koordinaten eines Punktes \mathbf{q} in einem durch Punkte $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ gegebenen Dreieck ($n = 3$)

$$\begin{aligned}\mathbf{q} &= \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + \lambda_2 \mathbf{p}_2 \\ &= \lambda_0 \mathbf{p}_0 + \lambda_1 \mathbf{p}_1 + (1 - \lambda_0 - \lambda_1) \mathbf{p}_2\end{aligned}$$

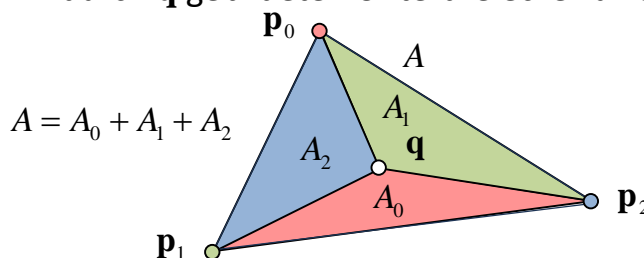


$$\lambda_0 + \lambda_1 + \lambda_2 = 1$$



Baryzentrische Koordinaten in Dreiecken

- Positionen in der Dreiecksebene sind durch normalisierte baryzentrische Koordinaten eindeutig bestimmt
- In Dreiecken sind diese Koordinaten äquivalent zum Verhältnis der (vorzeichenbehafteten) Flächen der durch \mathbf{q} gebildeten Unterdreiecke zur Gesamtfläche



$$\lambda_0 = \frac{A_0}{A}$$

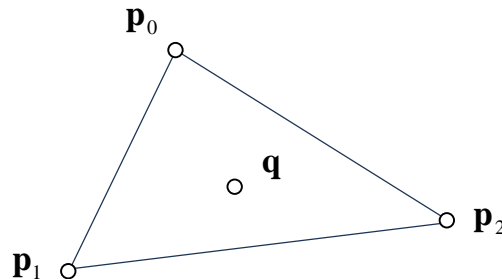
$$\lambda_1 = \frac{A_1}{A}$$

$$\lambda_2 = \frac{A_2}{A}$$



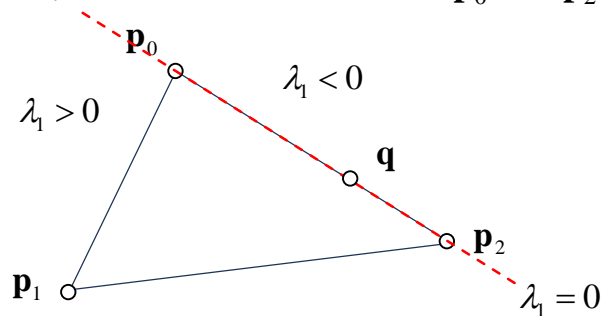
Baryzentrische Koordinaten in Dreiecken

- Punkt **q** liegt innerhalb eines Dreiecks, falls alle seine normalisierten baryzentrische Koordinaten $\lambda_{0,1,2} > 0$
- Punkt **q** liegt im Flächenschwerpunkt des Dreiecks falls $\lambda_0 = \lambda_1 = \lambda_2 = 1/3$



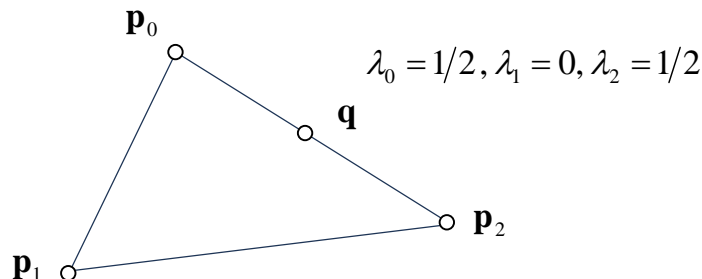
Baryzentrische Koordinaten in Dreiecken

- Punkt **q** liegt außerhalb eines Dreiecks, falls eine seiner normalisierten baryzentrischen Koordinaten $\lambda_i < 0$
- Punkte **q**₁ und **q**₂ liegen auf Linie, falls eines ihrer λ_i identisch, z.B. auf einer Linie durch **p**₀ und **p**₂ falls $\lambda_1 = 0$



Baryzentrische Koordinaten in Dreiecken

- Punkt \mathbf{q} liegt außerhalb eines Dreiecks, falls eine seiner normalisierten baryzentrischen Koordinaten $\lambda_i < 0$
- Punkte \mathbf{q}_1 und \mathbf{q}_2 liegen auf Linie, falls eines ihrer λ_i identisch, z.B. auf einer Linie durch \mathbf{p}_0 und \mathbf{p}_2 falls $\lambda_1 = 0$



Baryzentrische Interpolation

- Baryzentrische Koordinaten können für lineare Interpolation (vektorwertiger) Funktionen f über eine Dreiecksfläche verwendet werden
- Gegeben seien Funktionswerte an Dreieckspunkten

$$f(\mathbf{p}_i) = \mathbf{f}_i, \quad i = 0, 1, 2 \quad f : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \quad (\text{z.B. RGB})$$
- Interpolation der Werte an einem Punkt \mathbf{x} im Dreieck, über dessen baryzentrische Koordinaten $\lambda_{x,0}, \lambda_{x,1}, \lambda_{x,2}$

$$g(\mathbf{x}) = \lambda_{x,0} \mathbf{f}_0 + \lambda_{x,1} \mathbf{f}_1 + \lambda_{x,2} \mathbf{f}_2$$
- Hinweis: Interpolation auch auf außerhalb des Dreiecks erweiterbar (über negative Koordinaten)



Problemstellung

- Gegeben sei eine große Anzahl von Wertepaaren (x_i, y_i) , $i = 0, \dots, m$, z.B. Messdaten; vorerst $\mathbb{R} \rightarrow \mathbb{R}$
- Gesucht ist eine (approximierende) Funktion f , welche von n Parametern c_j , $j = 0, \dots, n-1$, abhängt, und z.B. eine Fehlermetrik hinsichtlich der Daten minimiert
- Als Funktionen werden z.B. Polynome (u.a. Geraden) sowie weitere nichtlineare Abbildungen verwendet
- Im Bereich der Optimierung wird dies Problem im Rahmen der Ausgleichsrechnung (z.B. via Regression) behandelt



Methode der Kleinsten Fehlerquadrate

- Übliche Fehlermetrik: Summe der quadratischen Differenzen zwischen den gegebenen Daten und den Funktionswerten

$$\arg \min_{c_j} \|f_{c_j}(x_i) - y_i\|_2^2 \quad (\text{euklidische Norm})$$

- Als Spezialfall kann ein lineares Modell angenommen werden

$$f(x) = c_1 x + c_0$$

- Ziel ist somit die Bestimmung der Parameter c_0 und c_1 , welche die Fehlerquadratsumme minimieren



Methode der Kleinsten Fehlerquadrate

- Erinnerung: mit $n = 2$ Unbekannten, bei üblicherweise $m \gg n$ Bedingungen, ergibt sich ein überbestimmtes Gleichungssystem; dies meistens ohne exakte Lösung

$$\begin{bmatrix} x_0 & 1 \\ x_1 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix} \quad \mathbf{Ac} = \mathbf{y}$$

- Statt dessen wird ein Minimierungsproblem gelöst

$$(\mathbf{Ac} - \mathbf{y})^T (\mathbf{Ac} - \mathbf{y}) = (\mathbf{Ac})^T (\mathbf{Ac}) - 2(\mathbf{Ac})^T \mathbf{y} + \mathbf{y}^T \mathbf{y} = \mathbf{r}^T \mathbf{r} \rightarrow \min$$



Methode der Kleinsten Fehlerquadrate

- Minimieren der Fehlerquadratsumme

$$\nabla \left((\mathbf{Ac})^T (\mathbf{Ac}) - 2(\mathbf{Ac})^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \right) = \mathbf{0}$$

$$2\mathbf{A}^T \mathbf{Ac} - 2\mathbf{A}^T \mathbf{y} = \mathbf{0}$$

$$\mathbf{A}^T \mathbf{Ac} = \mathbf{A}^T \mathbf{y} \quad (\text{Normalengleichungen})$$

$$\mathbf{c} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \quad (\text{Pseudo-Inverse})$$

- Systemmatrix $\mathbf{A}^T \mathbf{A}$ ist symmetrisch sowie positiv-definit (für $m > n$), und von Größe $(n+1) \times (n+1)$
- Lösung LGS ggbs. mittels numerischer Verfahren



Methode der Kleinsten Fehlerquadrate

- Im Falle einer einfachen, linearen Modellfunktion ist die Lösung gegeben durch:

$$c_1 = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^m (x_i - \bar{x})^2} \quad c_0 = \bar{y} - c_1 \bar{x}$$

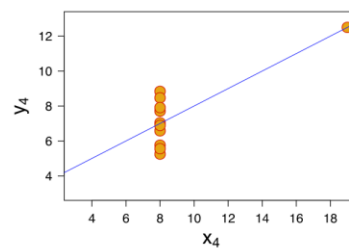
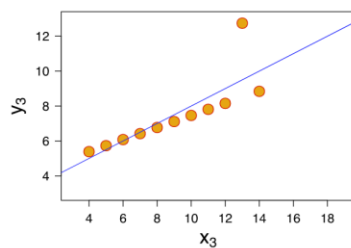
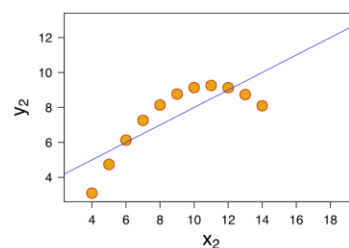
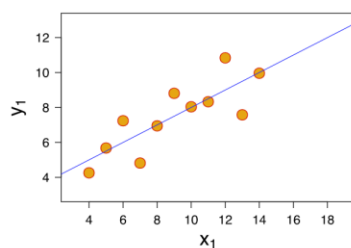
mit

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i \quad \bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$$

- Es bleibt zu prüfen, ob die lineare Modellannahme berechtigt gewesen ist



Visualisierung



Weitere Hilfreiche Literatur

- E. Weitz, **Konkrete Mathematik (nicht nur) für Informatiker**, 1. Auflage, eBook ISBN 978-3-658-21565-1, Springer, 2018.
- H.-R. Schwarz, N. Köckler, **Numerische Mathematik**, 8. Auflage, eBook ISBN 978-3-8348-8166-3, Springer, 2011.
- R. L. Burden, J. D. Faires, A. M. Burden, **Numerical Analysis**, 10th edition, ISBN 978-1305-25366-7, Cengage Learning, 2015.



Vorlesungsplan

Datum	Thema	Proseminar
11.03.22	Einführung, Grundlagen, Funktionen	<i>(Beginn zuvor am 8.3.)</i>
18.03.22	Differentialrechnung	
25.03.22	Integralrechnung	
01.04.22	Differentialgleichungen	
08.04.22	Weitere Funktionen	
<i>Osterferien</i>		
29.04.22	Reihen und Folgen	
06.05.22	Numerische Auswertung von Funktionen	
13.05.22	Lösung von Gleichungssystemen	
20.05.22	Interpolation	
27.05.22	Zufallszahlen	
03.06.22	Komplexe Zahlen	
10.06.22	Klausurvorbereitung	
01.07.22	Klausur	

