



Interpolation und Quadratur

Interpolation

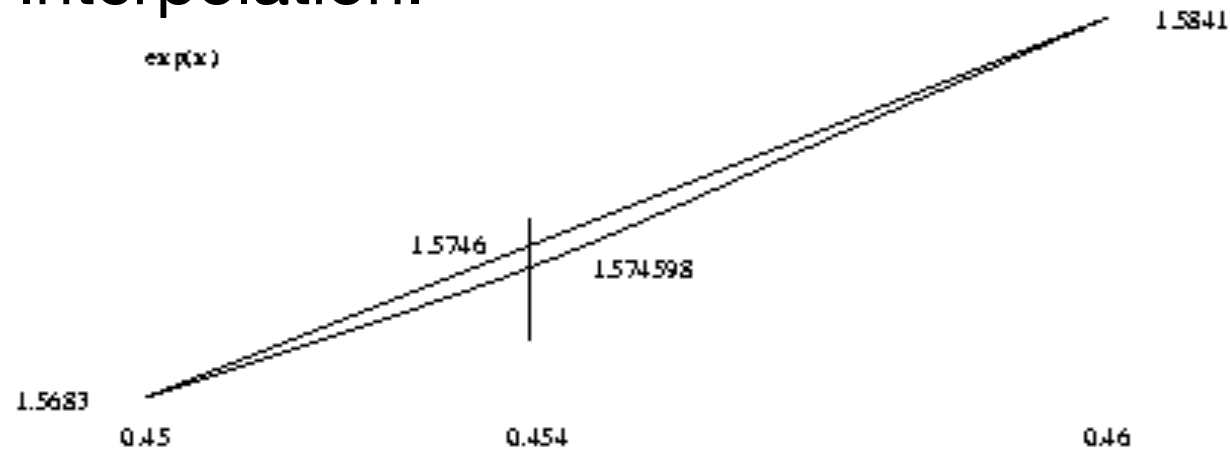
Interpolation

Beispiel: Interpolation mit Tafelwerken für exp, sin, cos, log

Gesucht: $\exp(0.454)$;

Tabelliert: $\exp(0.45)$ und $\exp(0.46)$

Lineare Interpolation.



x:	0.42	0.43	0.44	0.45	0.46
exp(x):	1.5683	1.5841



Allgemeine Problemstellung

Gegeben: Punktepaaire (x_j, y_j) , $j=0, 1, \dots, n$ (paarweise verschieden); und: linear unabhängige Funktionen $g_k(x)$, $k=0, 1, \dots, n$

Gesucht: Koeffizienten c_k , $k=0, 1, \dots, n$ so dass:

$$G(x_i) := \sum_{k=0}^n c_k g_k(x_i) = y_i \quad \text{für } j=0, 1, \dots, n$$

$$\begin{pmatrix} g_0(x_0) & \cdots & g_n(x_0) \\ \vdots & & \vdots \\ g_0(x_n) & \cdots & g_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

$$\mathbf{A}\mathbf{c} = \mathbf{y}$$

$(n+1) \times (n+1)$ lineares Gleichungssystem



Unterschied zu Approximation

Gegeben: Punktepaare (x_j, y_j) , $j=0, 1, \dots, n$ (paarweise verschieden); und: linear unabhängige Funktionen $g_k(x)$, $k=0, 1, \dots, m$ (im Allgemeinen: $m < n$)

Gesucht: Koeffizienten c_k , $k=0, 1, \dots, m$ so dass

$$\sum_{i=1}^n \left(\sum_{k=0}^m c_k g_k(x_i) - y_i \right)^2 \text{ minimal wird, d.h.:}$$

$$\min_{\mathbf{c}} \|\mathbf{A}\mathbf{c} - \mathbf{y}\|^2$$

In der Regel nimmt man hierfür die Euklidische Norm, da dann die Minimierung effizienter lösbar ist.



Spezialfall Polynom-Interpolation

Ansatzfunktionen $g_k(x)$ sind Polynome x^k

Gesucht: Koeffizienten $c_k, k=0,1,\dots,n$ mit

$$p(x_j) = \sum_{k=0}^n c_k x_j^k = y_j \quad \text{für } j=0,1,\dots,n$$

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$

$n+1$ Gleichungen für $n+1$ Unbekannte: Teuer! $O(n^3)$



Lösung mit Lagrange-Polynomen

Idee: Definiere geschickt Basis-Polynome:

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

das sind also $n+1$ Polynome vom Grad n

Eigenschaften der Lagrange-Polynome:

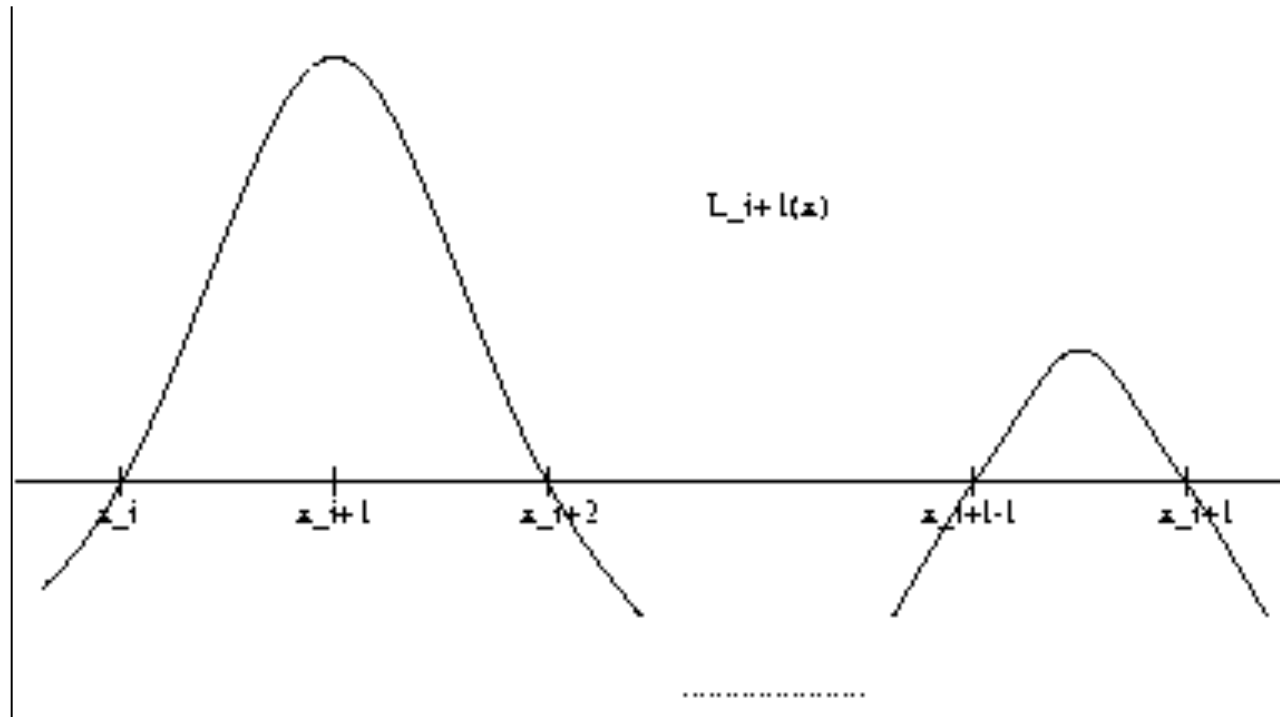
$$L_j(x_i) = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$$

Gesucht: $p(x) = \sum_{j=0}^n c_j L_j(x)$

das die Interpolationsbedingungen erfüllt.



Beispiel



Lagrange-Polynome

Aus diesen Eigenschaften ergibt sich zur Lösung des Interpolationsproblems ein lineares Gleichungssystem für die Koeffizienten:

$$\begin{pmatrix} L_0(x_0) & \cdots & L_n(x_0) \\ \vdots & & \vdots \\ L_0(x_n) & \cdots & L_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}$$



Existenz und Eindeutigkeit der Lösung

Die Lösung ist $c_j = y_j$; daher ist ein Interpolationspolynom:

$$p(x) = \sum_{j=0}^n y_j L_j(x)$$

denn es ist $p(x_i) = \sum_{j=0}^n y_j L_j(x_i) = y_i L_i(x_i) = y_i$.

Damit ist die Existenz eines interpol. Polynoms gezeigt.
Eindeutigkeit?



Existenz und Eindeutigkeit der Lösung

Die Lösung ist $c_j = y_j$; daher ist ein Interpolationspolynom:

$$p(x) = \sum_{j=0}^n y_j L_j(x)$$

denn es ist $p(x_i) = \sum_{j=0}^n y_j L_j(x_i) = y_i L_i(x_i) = y_i$.

Damit ist die Existenz eines interpol. Polynoms gezeigt.
Eindeutigkeit?

Hauptsatz der Algebra:

Jedes Polynom $p(x)$ vom Grad n kann als Produkt von n linearen Faktoren (den ev. komplexen Nullstellen z_k) geschrieben werden in der Form:

$$p(x) = \alpha(x - z_1)(x - z_2) \cdots (x - z_n)$$



Beweis der Eindeutigkeit

Ann.: Es gibt zwei Polynome p und q vom Grad $\leq n$, die beide die Interpolationsbedingungen erfüllen.

Definiere neues Polynom $h(x) := p(x) - q(x)$. Dann gilt

$$h(x) = \alpha(x - z_1)(x - z_2) \cdots (x - z_n)$$

und $h(x_i) := p(x_i) - q(x_i) = 0$ für $i=0,1,\dots,n$

Daher hat das Polynom $h(x)$ den Grad n und $n+1$ Nullstellen.

Aus dem Hauptsatz der Algebra folgt daher, dass

$\alpha = 0$ sein muss, und daher ist $h(x) \equiv 0$, oder

$$p(x) \equiv q(x).$$

Also existiert genau ein Interpolationspolynom!

Problem: Lagrange zur Lösung der Interpolation nicht geeignet, da numerisch problematisch und teuer.



Berechnung des Interpolationspolynoms

Löse nicht das lineare Gleichungssystem, da dies zu teuer ist!

Außerdem wird oft nur der Wert des Polynoms an einer einzigen Stelle gesucht!

Idee: Berechne induktiv interpolierende Polynome für immer mehr Stützstellen.

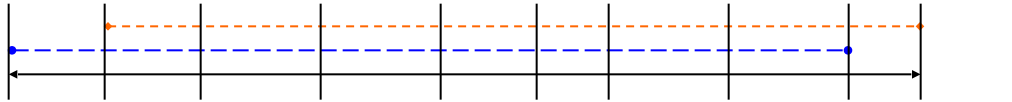
Setze dazu $p_{i,\dots,i+l}(x)$ als das interpolierende Polynom vom Grade l , das genau an den Stellen x_i, \dots, x_{i+l} die Interpolations-Bedingungen erfüllt.

Zur Bestimmung von $p_{i,\dots,i+l}(x)$ verwende die interpolierenden Polynome $p_{i+1,\dots,i+l}(x)$ und $p_{i,\dots,i+l-1}(x)$ vom Grade $l-1$, die zu den Stützstellen x_{i+1}, \dots, x_{i+l} bzw. x_i, \dots, x_{i+l-1} gehören.



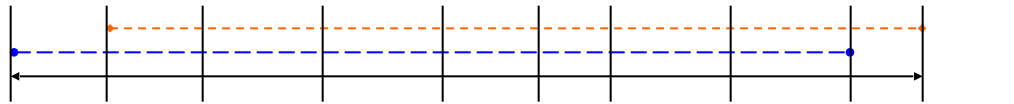
Rekursive Berechnungsformel

$$p_{i,\dots,i+l}(x) = \frac{(x - x_i)p_{i+1,\dots,i+l}(x) - (x - x_{i+l})p_{i,\dots,i+l-1}(x)}{x_{i+l} - x_i} \quad (*)$$



Rekursive Berechnungsformel

$$p_{i,\dots,i+l}(x) = \frac{(x - x_i)p_{i+1,\dots,i+l}(x) - (x - x_{i+l})p_{i,\dots,i+l-1}(x)}{x_{i+l} - x_i} \quad (*)$$



Nachprüfen der Interpolationsbedingung:

$$p_{i,\dots,i+l}(x_i) = \frac{0 - (x_i - x_{i+l})y_i}{x_{i+l} - x_i} = y_i$$

$$p_{i,\dots,i+l}(x_{i+l}) = \frac{(x_{i+l} - x_i)y_{i+l} - 0}{x_{i+l} - x_i} = y_{i+l}$$

und für alle anderen j mit $i < j < i+l$:

$$p_{i,\dots,i+l}(x_j) = \frac{(x_j - x_i)y_j - (x_j - x_{i+l})y_j}{x_{i+l} - x_i} = y_j$$



Anwendung der Rekursion

Wegen der Eindeutigkeit des interpolierenden Polynoms ist jedes der so definierten Polynome genau die eindeutige Lösung des jeweiligen Interpolationsproblems!

Daher ist $p_{i,\dots,i+l}(x)$ die gesuchte Lösung!

Anwendung der Formel (*) zur punktweisen Auswertung des Interpolationspolynoms an einer Stelle \bar{x} : $p(\bar{x}) = ?$

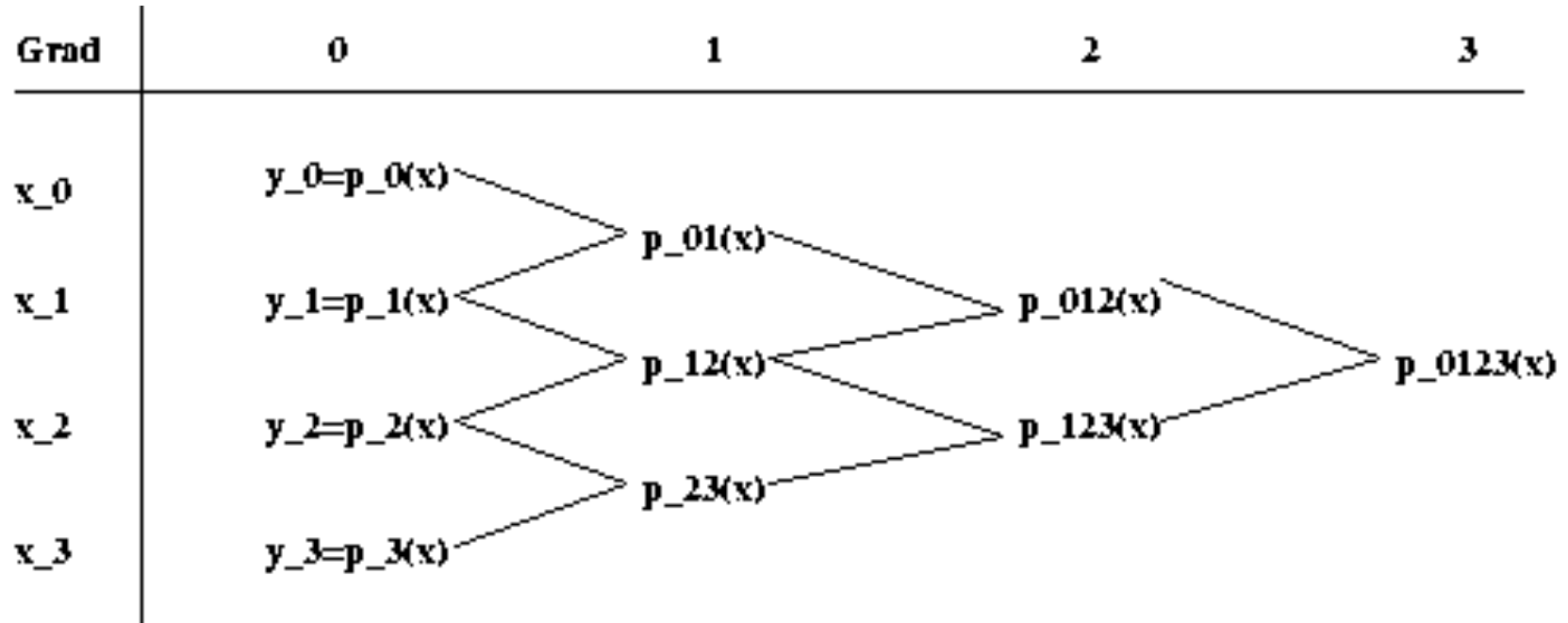
Eingabe: Stützwerte $(x_j, y_j), j=0,\dots,n$ und Stelle \bar{x} ;

Ausgabe: $p_{i,\dots,i+l}(\bar{x})$.

Tableau-artige Berechnung der Interpolationspolynome aufsteigenden Grades, aber nur an der Stelle \bar{x} .



Neville-Tableau



Erste Spalte sind konstante interpolierende Polynome, also genau die jeweils vorgegebenen Werte y_i .

Zweite Spalte sind interpolierende lineare Polynome zu jeweils zwei benachbarten Stützstellen.

Letzte Spalte enthält das interpolierende Polynom zu allen vorgegebenen Stützstellen.



Neville-Tableau

Neue Stützstelle x_4 mit Wert y_4 kann in das Tableau eingefügt werden und führt zu einer neuen ‚Zeile‘ und einer neuen Endspalte $p_{01234}(x)$.

Auswertung des Tableaus jeweils nur an einer festen Stelle x möglich.

Beispiel: $x_0 = 0, y_0 = 1$,
 $x_1 = 1, y_1 = 3$ ’ $x_2 = 3, y_2 = 2$

Auswertung des interpolierenden Polynoms an der Stelle $x=2$ mit Lagrange, bzw. Neville-Tableau:



Vergleich mit Lagrange

$$L_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)}, \quad L_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)} \quad \text{und}$$

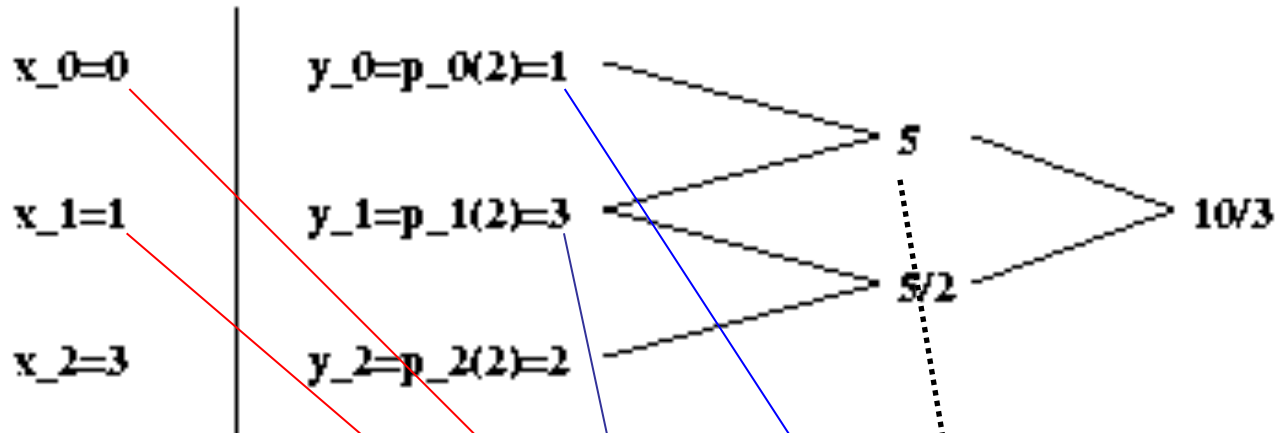
$$L_2(x) = \frac{(x-0)(x-1)}{(3-0)(3-1)}$$

→

$$p_{012}(2) = 1 \cdot L_0(2) + 3 \cdot L_1(2) + 2 \cdot L_2(2) = -\frac{1}{3} + 3 + 2 \cdot \frac{1}{3} = \frac{10}{3}$$



Mit Neville-Tableau



Da nach (*)

$$p_{01}(2) = \frac{(2-0) \cdot 3 - (2-1) \cdot 1}{1-0} = 5$$

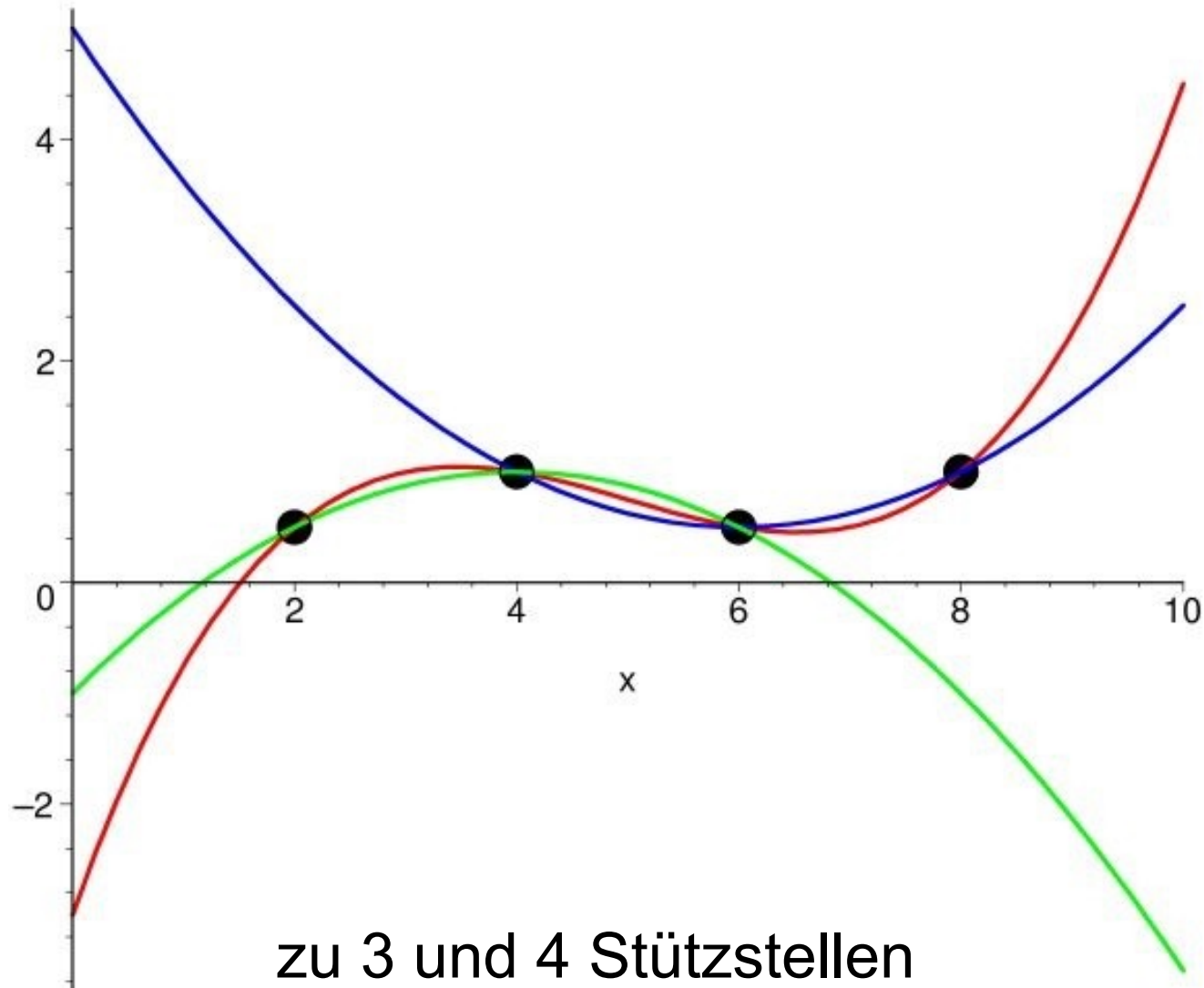
$$p_{01}(2) = \frac{(2-x_0)p_1(2) - (2-x_1)p_0(2)}{x_1 - x_0}$$

$$p_{12}(2) = \frac{(2-1) \cdot 2 - (2-3) \cdot 3}{3-1} = \frac{5}{2}$$

und daher auch
$$p_{012}(2) = \frac{(2-0) \cdot \frac{5}{2} - (2-3) \cdot 5}{3-0} = \frac{10}{3}$$



Folge von Interpolationspolynomen



Fehler bei der Polynominterpolation

Satz: Gegeben Stützstellen $x_0 < \dots < x_n$ und genügend oft diff'bare Funktion $f(x)$. $p(x)$ sei das interpolierende Polynom vom Grad n mit $p(x_i) = f(x_i)$, $i = 0, \dots, n$.

An einer beliebigen Stelle \bar{x} gilt dann für die Abweichung zwischen p und f :

$$f(\bar{x}) - p(\bar{x}) = \frac{f^{(n+1)}(\chi)}{(n+1)!} (\bar{x} - x_0) \dots (\bar{x} - x_n)$$

Dabei ist $f^{(n+1)}(\chi)$ die $(n+1)$ -te Ableitung von f an einer Zwischenstelle χ aus dem Intervall

$$I = [\min\{x_0, x_n, \bar{x}\}, \max\{x_0, x_n, \bar{x}\}]$$

Frage: Wie gut wird f durch p dargestellt?



Beweis des Satzes (1)

Definiere Hilfsfunktion

$$g(x) = f(x) - p(x) - K(x - x_0) \dots (x - x_n)$$

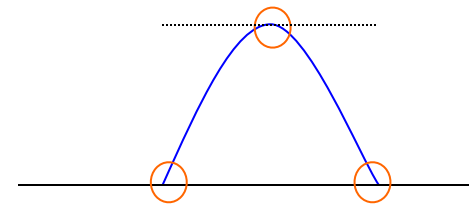
wobei $p(x)$ das interpolierende Polynom ist und K eine Konstante. D.h. die Nullstellen von g sind x_0, \dots, x_n

Wähle K so dass

$$K := \frac{f(\bar{x}) - p(\bar{x})}{(\bar{x} - x_0) \dots (\bar{x} - x_n)} \quad \Rightarrow \quad g(\bar{x}) = 0$$

Also hat die Funktion $g(x)$ $n+2$ Nullstellen!

Mit Mittelwertsatz besagt der Satz von Rolle, dass zwischen zwei Nullstellen einer stetig diff'baren Funktion f stets mindestens eine Nullstelle der Ableitung f' liegen muss (relatives Extremum mit waagrechter Tangente)



Beweis des Satzes (2)

Also hat die erste Ableitung g' mindestens noch $n+1$ Nullstellen im Intervall I .

Die zweite Ableitung g'' noch n Nullstellen, usw.

Die $n+1$ -te Ableitung hat noch eine Nullstelle in I

$$\begin{aligned} 0 &= g^{(n+1)}(\chi) \\ &= f^{(n+1)}(\chi) - p^{(n+1)}(\chi) - K \frac{d^{n+1}}{dx^{n+1}} ((x - x_0) \dots (x - x_n))|_{\chi} \\ &= f^{(n+1)}(\chi) - K(n + 1)! \end{aligned}$$

Daher folgt:
$$K = \frac{f^{(n+1)}(\chi)}{(n + 1)!}$$

Denn die $(n+1)$ -te Ableitung von $(x-x_0)\dots(x-x_n)$ ist gleich der $(n+1)$ -ten Ableitung von x^{n+1} allein.



Anwendung des Satzes

Die $(n+1)$ -te Ableitung ist auf dem Intervall I beschränkt, wenn f $(n+1)$ -mal stetig diff'bar ist. Dann existiert M mit

$$|K| \leq \frac{M}{(n+1)!} \quad , \text{ z.B. mit } \quad M = \max_{x \in I} |f^{(n+1)}(x)| < \infty$$

Insgesamt ergibt sich daher

$$\begin{aligned} |f(\bar{x}) - p(\bar{x})| &= |g(\bar{x}) + K(\bar{x} - x_0) \dots (\bar{x} - x_n)| \\ &= \left| \frac{f^{(n+1)}(\chi)}{(n+1)!} (\bar{x} - x_0) \dots (\bar{x} - x_n) \right| \\ &\leq \frac{M}{(n+1)!} |(\bar{x} - x_0) \dots (\bar{x} - x_n)| \end{aligned}$$



Folge dieser Abschätzung

Frage: Wie gut stellt $p(x)$ die gegebene Funktion $f(x)$ dar?

Entscheidend dafür sind

- die Größe der $n+1$ -ten Ableitung von f auf dem Intervall I
- die Größe der Funktion $w(x) := |(x - x_0) \dots (x - x_n)|$ an der Stelle \bar{x} abhängig von Wahl der Stützstellen x_i

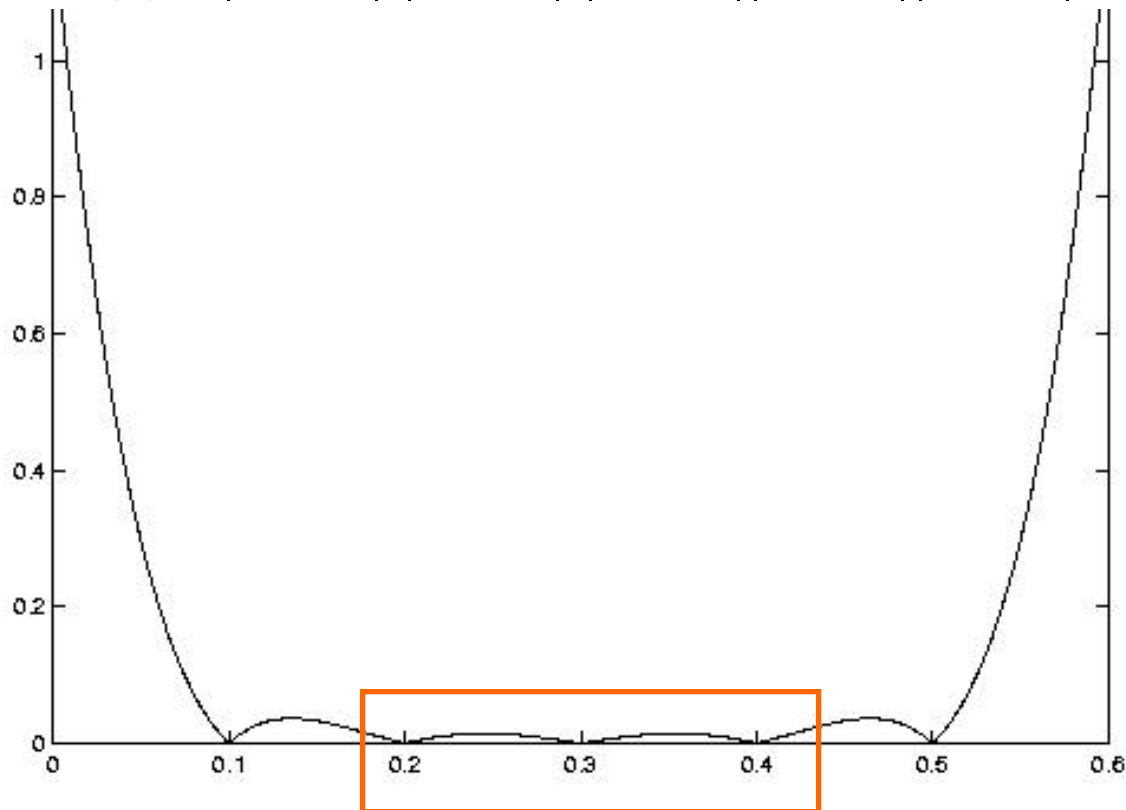


Folge dieser Abschätzung

Frage: Wie gut stellt $p(x)$ die gegebene Funktion $f(x)$ dar?

Beispiel: Verlauf der Funktion

$$w(x) = |x-0.1| |x-0.2| |x-0.3| |x-0.4| |x-0.5|$$



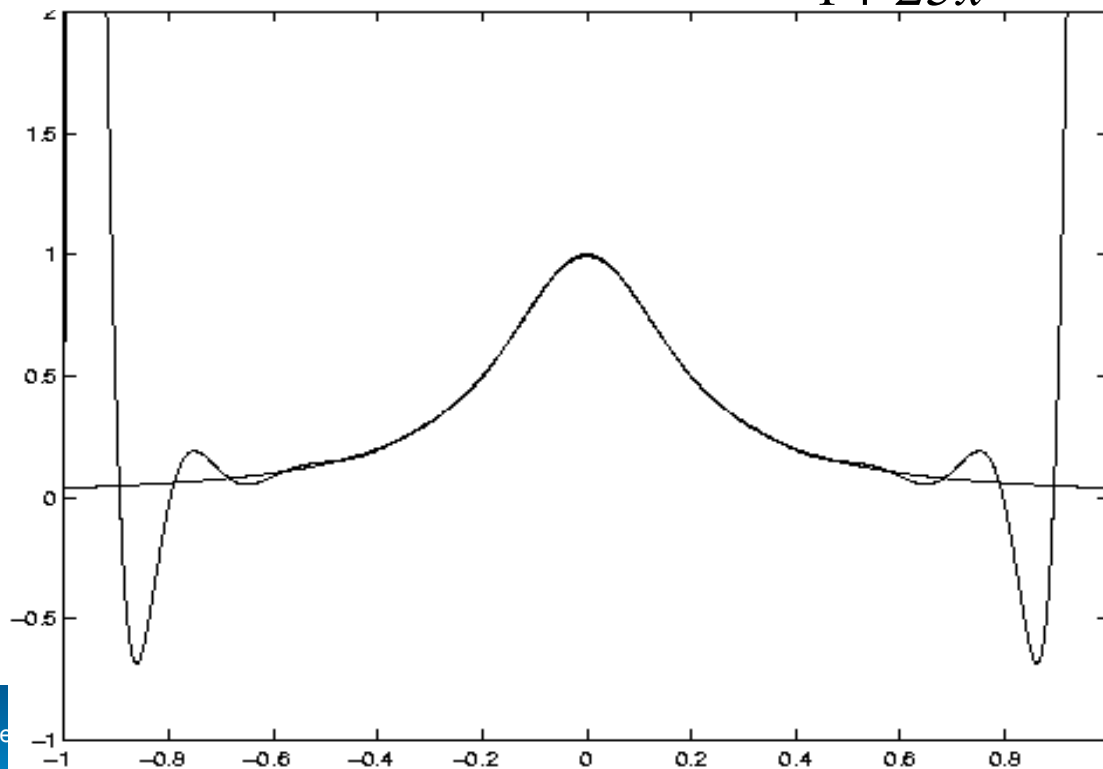
Folge dieser Abschätzung

Folgerung: Näherung am Rand schlechter!

Der Abstand zwischen Funktion und interpolierendem Polynom ist klein in der Mitte der Stützstelle.

Am Rand und außerhalb kann der Fehler groß werden.

Beispiel: Runge-Funktion $R(x) = \frac{1}{1 + 25x^2}$, $x \in [-1, 1]$



$R(x)$ und $p(x)$
vom Grad 20,
äquidistante
Stützstellen.

Wahl der Stützstellen

Bessere Wahl der Stützstellen, um den Fehler gleichmäßig auf dem ganzen Intervall klein zu halten:

Wähle die Stützstellen so, dass $w(x) = x^{n+1} + \dots = (x-x_0)\dots$ als Polynom vom Grad $(n+1)$ auf dem Intervall I möglichst **gleichmäßig klein** wird.

Spezialfall $n=2$, $x_0=0$ und $x_2=1$;

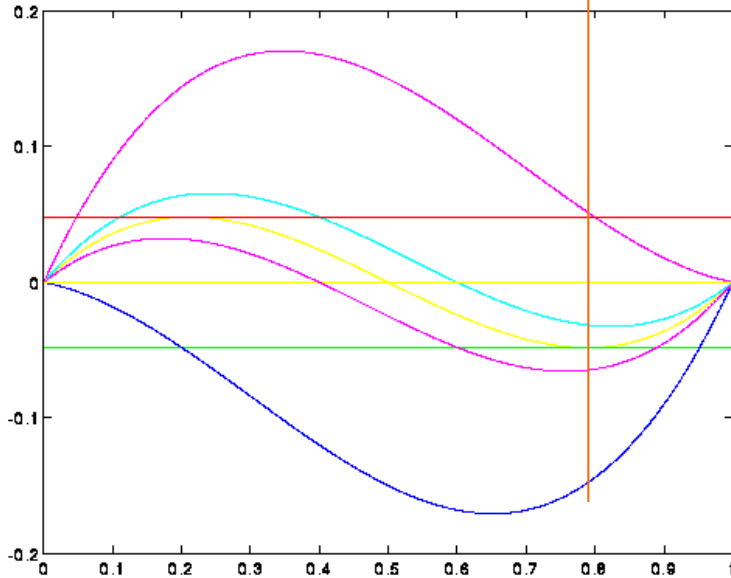
Bestimme x_1 so, dass $w(x)$ möglichst klein wird:

$$\begin{aligned} \min_{x_1} : \max_x |w(x)| &= \max_x |(x-0)(x-x_1)(x-1)| = \\ &= \max_x |x(x-1)(x-x_1)| \quad \text{in } [0,1] \end{aligned}$$

Lösung: $x_1=0.5$, gelbe Kurve



Wahl der Stützstellen

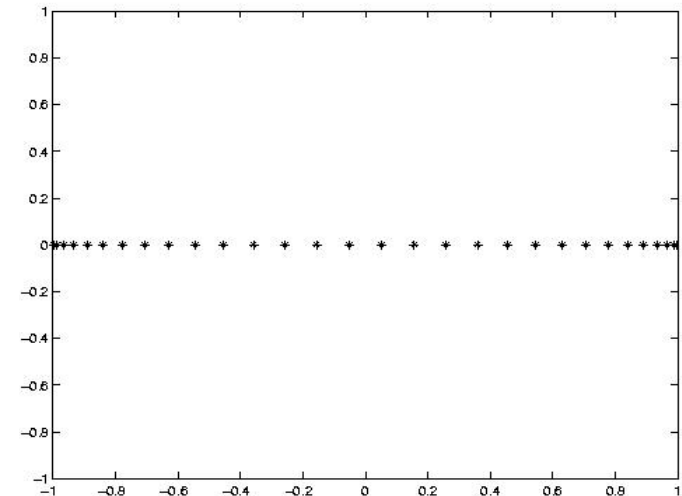


$w(x)$ für verschiedene x_1 .
Optimal, wenn alle Werte in
schmalem Band liegen.

Allgemeine Lösung im Intervall $[-1, 1]$: Tchebycheff-Polynom

Nullstellen:

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



Wahl der Stützstellen

Also verteile Stützstellen besser so, dass am Rand mehr Punkte sind, um den ev. großen Fehler dort auszugleichen.

Allgemein gilt:

Polynominterpolation mit Polynomen hohen Grades neigt zu Oszillationen (siehe Runge-Funktion) und wird kaum verwendet.

An Stelle von äquidistanten Stützstellen

$$x_j = a + \frac{j}{n}(b - a), \quad j = 0, 1, \dots, n$$

verwende man besser eine Verteilung, bei der am Rande mehr Stützstellen sind, z.B. Tchebycheff-knoten (s.o.)



Tchebycheff-Stützstellen

Allgemeines Stützstellenproblem im Intervall $[-1, 1]$:

$$\min_{a_0, \dots, a_n} \left\{ \max_{-1 \leq x \leq 1} |x^{n+1} + a_n x^n + \dots + a_1 x + a_0| \right\} =$$

$$\min_{x_0, \dots, x_n} \left\{ \max_{-1 \leq x \leq 1} |(x - x_0) \dots (x - x_n)| \right\}$$



Newtonform des Interpolationspolynoms

Manchmal ist man auch an einer expliziten Darstellung des Interpolationspolynoms selbst interessiert.

Allerdings eignet sich dazu nicht die übliche Form

$$p(x) = a_n x^n + \dots + a_1 x + a_0$$

Effiziente Auswertung dieser Form mittels **Hornerschema**:

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

durch das Programm:

```
y=a(n);  
FOR j=n-1,...,0:  
    y=y*x+a(j);  
ENDFOR
```



Newtonform

An Stelle der Standardform verwendet man $p(x)$ in einer Darstellung, in der die Stützstellen eingehen:

$$\begin{aligned} p(x) = & f[x_0] + f[x_0, x_1](x - x_0) \\ & + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\ & + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1}) \end{aligned}$$



Newtonform

An Stelle der Standardform verwendet man $p(x)$ in einer Darstellung, in der die Stützstellen eingehen:

$$\begin{aligned} p(x) = & f[x_0] + f[x_0, x_1](x - x_0) \\ & + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\ & + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1}) \end{aligned}$$

Die Koeffizienten $f[x_0, \dots, x_j]$ erhält man wieder aus der wesentlichen Formel (*) von oben:

$$\begin{aligned} p_{i, \dots, i+k}(x) &= \frac{(x - x_i)p_{i+1, \dots, i+k}(x) - (x - x_{i+k})p_{i, \dots, i+k-1}(x)}{x_{i+k} - x_i} \\ &= f[x_i] + f[x_i, x_{i+1}](x - x_i) + \dots + f[x_i, \dots, x_{i+k}](x - x_i) \dots (x - x_{i+k-1}) \end{aligned}$$



Newtonform

Sie lassen sich wieder aus einem Tableau (Tableau der ‚Dividierten Differenzen‘) der Reihe nach berechnen mittels

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$$

Dann kann durch Horner-artiges Schema $p(x)$ an der Stelle x ausgewertet werden.



Weitere Interpolationsansätze

(a) Hermite-Interpolation:

Um Oszillationen zu vermeiden will man auch die Ableitungen an den Stützstellen kontrollieren, im einfachsten Fall genau die ersten Ableitungen:

Vorgegeben sind $(x_j, y_j, y'_j), j=0, \dots, n$

Gesucht: ein Polynom vom Grad $2n+1$ mit

$$p(x_j) = y_j \quad \text{und} \quad p'(x_j) = y'_j, \quad j=0, 1, \dots, n$$



Hermite-Interpolation

- (1) Lösung durch Aufstellen des dadurch gegebenen linearen $2n+2 \times 2n+2$ Gleichungssystems für die gesuchten *Koeffizienten des Polynoms*:

$$\begin{pmatrix} 1 & x_0 & \dots & x_0^{2n+1} \\ 0 & 1 & \dots & (2n+1)x_0^{2n} \\ 1 & x_1 & \dots & x_1^{2n+1} \\ \vdots & \vdots & & \vdots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2n+1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y'_0 \\ y_1 \\ \vdots \end{pmatrix}$$

- (2) Lösung durch ‚Lagrange‘-Polynome, die genau an einer Stützstelle z.B. Wert 0 und Ableitung 1 haben, sonst an allen anderen Stützstellen Wert und Ableitung 0.



Hermite-Interpolation

$$L_j(x) := (x - x_0)^2 \prod_{k=0, k \neq j}^n (x - x_k)^2 (x - x_{j+1})^2 \prod_{k=j+2, k \leq n}^n (x - x_k)^2$$

1. Aufgabe:

Finde Polynom vom Grad $2n+1$, das

an der Stelle x_j den Wert 1 und Ableitung 0 hat, und

an allen anderen Stützstellen Wert und Ableitung 0:

Ansatz: $G_j(x) = L_j(x) \cdot (\alpha + \beta x)$

α und β sind daher so zu wählen, dass

$$L_j(x_j)(\alpha + \beta x_j) = 1 \quad \text{gilt und} \quad L_j'(x_j)(\alpha + \beta x_j) + L_j(x_j)\beta = 0$$

Dies liefert zwei Bedingungen für α und $\beta \rightarrow G_j(x)$



Hermite-Interpolation

2. Aufgabe:

Finde Polynom $H_j(x)$, das an der Stelle x_j den Wert 0 und Ableitung 1 hat, an allen anderen Stützstellen Wert und Ableitung 0.

Dann sind α und β so zu wählen, dass

$$L_j(x_j)(\alpha + \beta x_j) = 0 \quad \text{gilt} \quad \text{und} \quad L'_j(x_j)(\alpha + \beta x_j) + L_j(x_j)\beta = 1$$

Lösung liefert α und $\beta \rightarrow H_j(x)$

Damit erhält man als Lösung der Hermite-Interpolation:

$$p(x) = \sum_{j=0}^n \left(y_j G_j(x) + y'_j H_j(x) \right)$$



Spline-Interpolation

Man versucht, die Oszillationen zu verhindern, indem man die interpolierende Funktion stückweise aus Polynomen niedrigen Grades zusammenbaut.

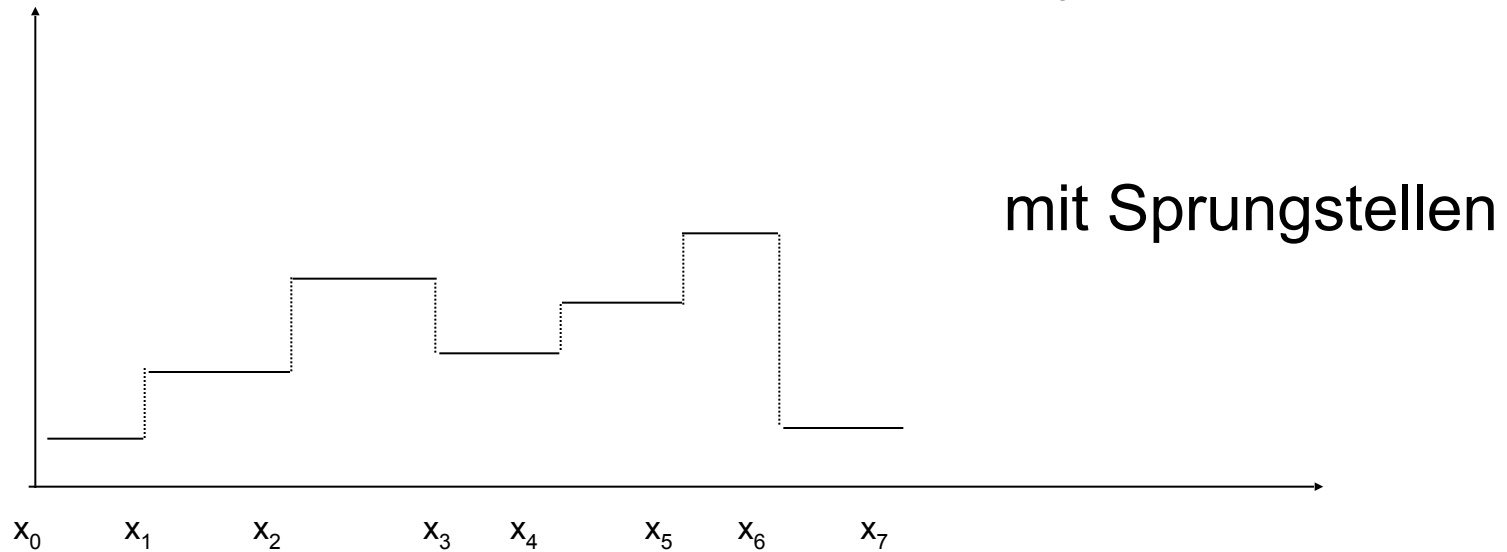
Definition einer Splinefunktion S vom Grade k zu Stützstellen x_0, \dots, x_n :

$S(x)$ sei insgesamt $(k-1)$ -mal stetig differenzierbar und auf den einzelnen Teil-Intervallen $[x_i, x_{i+1}]$ jeweils ein Polynom k -ten Grades.

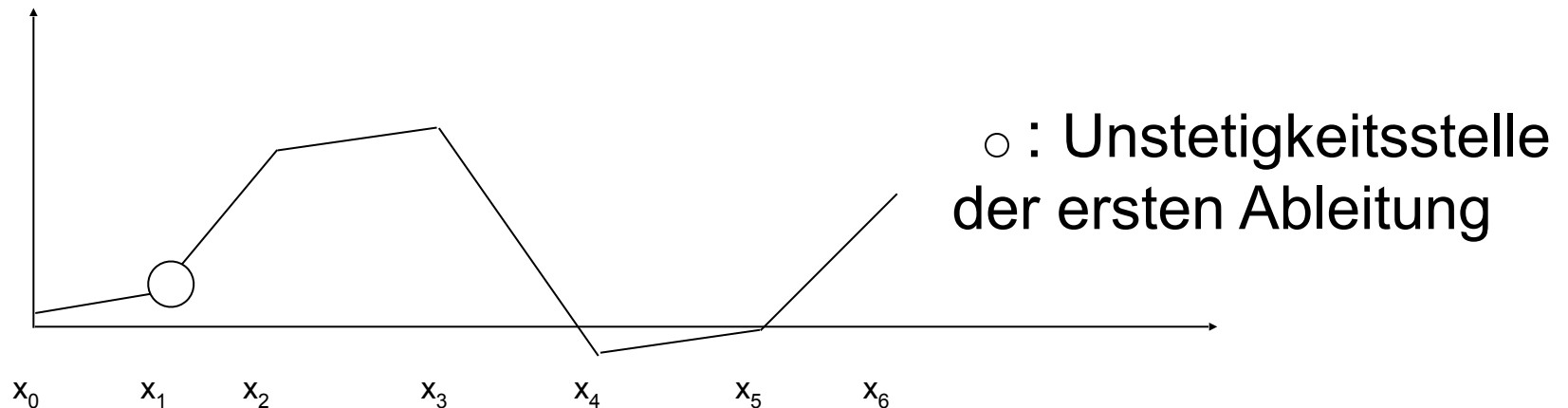


Beispiele

Für $k=0$ stückweise konstante Funktionen (Treppenfunktionen):

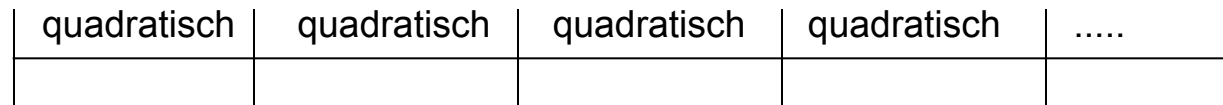


Für $k=1$ stetig, stückweise linear:



Beispiele

Für $k=2$ stückweise quadratische Polynome, stetig diff'bar verbunden.



Für $k=3$ (kubische Splines) stückweise Polynome dritten Grades; an den Nahtstellen sind die Funktion samt erster und zweiter Ableitung stetig. Diese Splines werden am häufigsten verwendet.

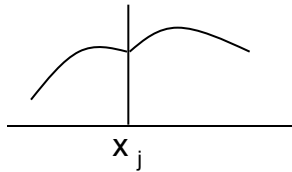
Das Interpolationsproblem mit Splinefunktionen lautet dann:
Bestimme Spline $S(x)$ zu k so, dass $S(x_j) = y_j$, $j=0, \dots, n$

Dies führt auf **lineares Gleichungssystem** für die Parameter, die $S(x)$ bestimmen, abhängig von k .



Anzahl der Gleichungen

Neben den $n+1$ Interpolationsbedingungen gehen noch die Bedingungen an den Nahtstellen ein (stetig, stetig diff'bar,...), z.B. in der Form



$$p_{links}(x_j) = p_{rechts}(x_j)$$

$$p'_{links}(x_j) = p'_{rechts}(x_j) \quad , \text{ usw.}$$

p sollen dabei Polynome k -ten Grades sein und S, S', \dots stetig bis einschließlich $(k-1)$ -te Ableitung.

Bestimmt werden die Koeffizienten der Teilpolynome vom Grad k , so dass die Interpolationsbedingungen erfüllt sind und die Übergänge entsprechend stetig sind.



Anzahl der Unbekannten

Wir brauchen n Polynome vom Grad k : $n(k+1)$ Koeff.

Anzahl Bedingungen: $n+1$ Interpol.-Bedingungen

$S(x_j) = y_j$ und $(n-1)k$ stetige Übergänge $p_j^{(i)}(x_j) = p_{j+1}^{(i)}(x_j)$

für $i=0, \dots, k-1$ an den inneren Stellen $j=1, \dots, n-1$

Insgesamt also $nk+n-k+1 = n(k+1) + (1-k)$ Bedingungen.

Füge noch $k-1$ neue Bedingungen hinzu, um ein quadratisches lineares Gleichungssystem mit genauso vielen Gleichungen wie Unbekannten zu erhalten,

z.B. - Bedingungen über Ableitungen an den Endpunkten,
oder - Periodizität, o.ä.

also $S''(a) = S''(b) = 0$, oder $S_j(a) = S_j(b)$.

Dann ergibt sich dann eine $n(k+1) \times n(k+1)$ - Matrix.



Alternative Methode

Idee: Bestimme wieder geeignete **Basis** von Splinefunktionen und löse das Interpolationsproblem in dieser Basis →

B-Spline-Basis (analog zu Lagrange-Polynomen):

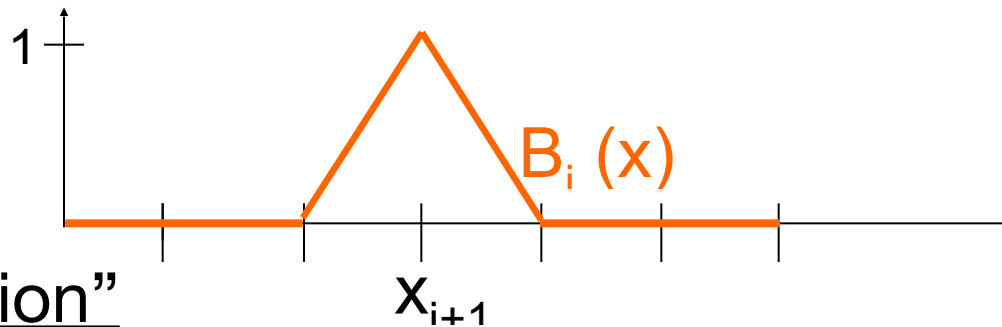
Elementare Splinefunktion vom Grade k , die

- höchstens an einer Stützstelle den Wert 1 hat,
- und an ‚fast‘ allen anderen 0.



Bestimmung einer Basis

Sei $k=1$. Gesucht: stückweise lineare, stetige Funktion, die genau bei x_{j+1} den Wert 1 annimmt, und an anderen Stützstellen 0.



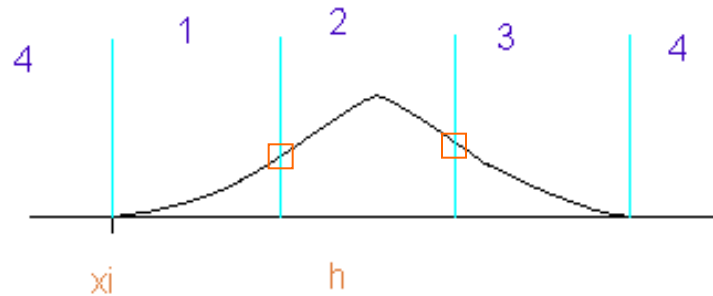
$$B_j(x) := \begin{cases} \frac{x - x_j}{x_{j+1} - x_j} & \text{für } x \in [x_j, x_{j+1}] \\ \frac{x - x_{j+2}}{x_{j+1} - x_{j+2}} & \text{für } x \in [x_{j+1}, x_{j+2}] \\ 0 & \text{sonst} \end{cases}$$

Lösung der Interpolations-Aufgabe: $S(x) = \sum_{j=0}^n y_j B_{j-1}(x)$



Bestimmung einer Basis

B-Spline für $k=2$ (quadratisch):



$$B_i(x) := \frac{1}{2h^2} \begin{cases} (x - x_i)^2 & \text{für } x \in]x_i, x_{i+1}[& (1) \\ h^2 + 2h(x - x_{i+1}) - 2(x - x_{i+1})^2 & \text{für } x \in [x_{i+1}, x_{i+2}] & (2) \\ (x_{i+3} - x)^2 & \text{für } x \in]x_{i+2}, x_{i+3}[& (3) \\ 0 & \text{sonst} & (4) \end{cases}$$

Stetig differenzierbar, stückweise quadratisch.

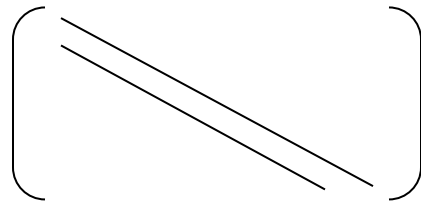


Interpolationsproblem

Ansatz:
$$p(x) = \sum_{i=-1}^{n-1} a_i B_i(x), \quad a_i = ?$$

$$y_j = p(x_j) = \sum_{i=-1}^{n-1} a_i B_i(x_j) = a_{j-2} B_{j-2}(x_j) + a_{j-1} B_{j-1}(x_j) \quad \text{für } j=0,1,\dots,n.$$

Ergibt bidiagonales lineares Gleichungssystem zur Berechnung der Koeffizienten a_j :



Trivial zu lösen! (Hier $B_{-2}(x) \equiv 0$, und x_{-1} dazu)



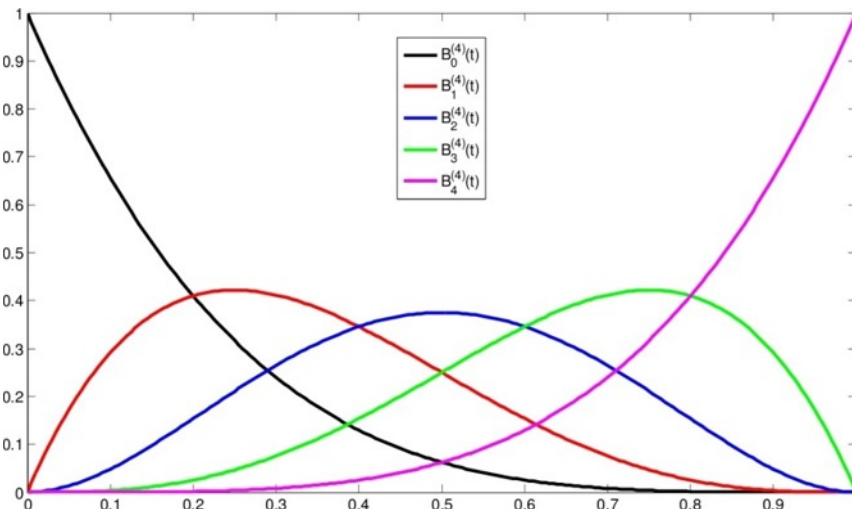
Bezier-Kurven

Gesucht sind Kurven, die nicht exakt durch die Stützpunkte laufen, aber sich durch die Stützpunkte gut kontrollieren lassen (keine Interpolation!).

(lokale Änderungen sollen nur lokal wirken)

Ausgangspunkt:

Bernsteinpolynome in $[0,1]$



$$B_i^{(n)}(t) := \binom{n}{i} \cdot (1-t)^{n-i} t^i \quad \text{für } i = 0, 1, \dots, n$$

Für $n=4$ sind dies also 5 Polynome.

Es gilt:
$$\sum_{i=0}^n B_i^{(n)}(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i = 1$$



Bezier-Kurven

Jedes $B_i^{(4)}(t)$ hat einen Teilbereich, indem es die anderen Polynome dominiert.

Gegeben: Stützpunkte $\mathbf{b}_i \in \mathbb{R}^k$, $i=0, \dots, n$, als Vektoren im \mathbb{R}^k .

Die Bezierkurve zu diesen Punkten ist definiert durch:

$$X(t) := \sum_{i=0}^n \mathbf{b}_i B_i^{(n)}(t) \quad t \in [0, 1]$$

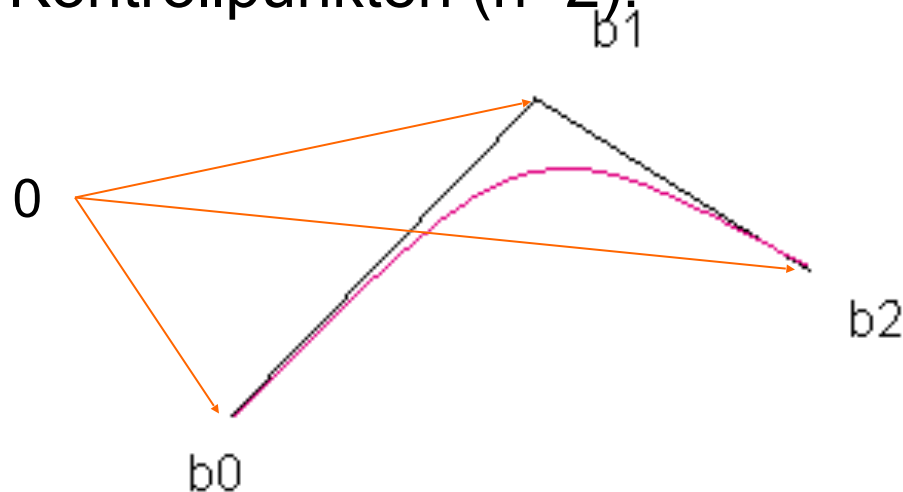
Beginnend mit $t=0$ hat der Vektor $X(t)$ zunächst den Wert \mathbf{b}_0 , und fährt dann für wachsendes t der Reihe nach die anderen Kontrollpunkte \mathbf{b}_i näherungsweise ab, bis er für $t=1$ bei \mathbf{b}_n anlangt.



Bezier-Kurven

Dabei überwiegt in einem Teilbereich von $[0,1]$ jeweils eines der Bernsteinpolynome $B_i(x)$ die anderen, und sorgt dafür, dass die Kurve zu dem entsprechenden Kontrollpunkt b_i gezogen wird.

Verlauf bei drei Kontrollpunkten ($n=2$):

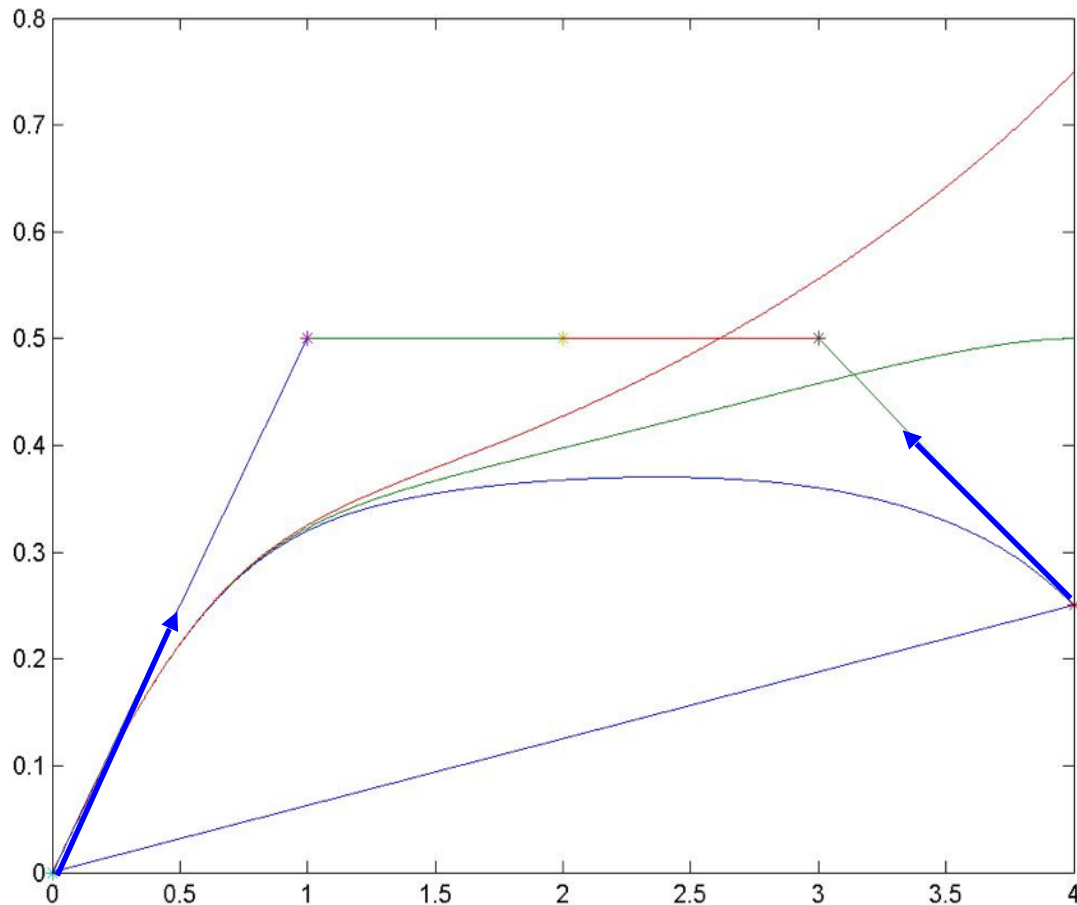


Die sich ergebende Bezier-Kurve ist eine gewichtete Summe von Vektoren!

Ergibt keine interpolierende Kurve, aber gut steuerbar →
Computergraphik



Beispiel Bezier-Kurve



Hier sind drei verschiedene Bezierkurven dargestellt, die sich nur im vierten Kontrollpunkt unterscheiden.



Bezier-Kurven

- Die Kurve verläuft stets innerhalb der konvexen Hülle der Kontrollpunkte (als Polygon eingezeichnet für den Fall, dass der rechte Kontrollpunkt den kleinsten Wert hat.).
 - Am Anfangs-, bzw. Endpunkt zeigt die Tangente in Richtung des nächsten Nachbarpunktes (blauer Pfeil).
 - Bei Veränderung des rechten Kontrollpunktes bleibt die linke Kurve fast unverändert; d.h. lokale Änderungen wirken sich kaum auf den Rest der Kurve aus!



B-Splines

Verbesserung:

Ersetze Bernsteinpolynome durch B-Splines (Nurbs)

Vorteile:

- bessere Approximationseigenschaften
- einstellbare Differenzierbarkeit
(k -mal stetig diff'bar)
- bessere Lokalitätseigenschaften
(lokale Änderungen wirken wirklich nur lokal)



Zusammenfassung

**Eines der Hauptprobleme der Numerik:
Finde geeignete Ansatzfunktionen (Basis)**

Klassisch: Polynome $p(x)$,

rationale Funktionen $p(x)/q(x)$,

Taylor-, Potenz-, Laurent-Reihe , z.B. $\sum a_k x^k$

Fourierreihen, $\cos(kx)$, $\sin(kx)$, z.B. $\sum a_k e^{ikx}$

Analytische Darstellung

Heutzutage erforderlich:

Algorithmische Darstellung



Anforderungen für Ansatzfunktionen

- Variable Diff'barkeit, angepasst an das Problem
- Variable Approximationsgüte, kleiner Fehler
- Adaptivität, Auflösung angepasst an (lokale) Anforderungen
- Lokalität, d.h. lokale Änderungen wirken nur lokal
- Schnelle Algorithmen, dünne Gleichungen
- *Analysemöglichkeiten, vgl. Fouriertransformation*

Die ersten Punkte sind z.B. durch B-Splines erfüllt.

Es fehlt noch die Möglichkeit einer **„Frequenzanalyse“**, die wir bei der Fouriertransformation kennenlernen werden → Multiskalendarstellung, Wavelet, Multigrid



Vergleich der Interpolationsverfahren

- Standard-Polynom-Interpolation:

Gegeben $(x_j, y_j), j=0,1,\dots,n$

Gesucht interpolierendes Polynom vom Grad n

- Spline-Funktionen:

stückweise Polynome vom Grad k , $(k-1)$ -mal stetig diff'bar, aneinandergeheftet, so dass sie die Interpolationsbedingungen erfüllen.

- Hermite-Interpolation:

Gegeben $(x_j, y_j, y'_j), j=0,1,\dots,n$

Gesucht interpolierendes Polynom vom Grad $2n+1$



Vergleich der Interpolationsverfahren

- Nicht-interpolierende Kurven der Form

$\sum_{i=1,\dots,n} b_j \cdot B_j(t)$, $t \in I$, mit Kontrollpunkten b_j und Gewichten $B_j(t)$ (z.B. Bezier-, B-Spline-Gewichte).

- Stückweise Hermite-Interpolation, d.h.

Gegeben (x_j, y_j, y'_j) , $j=0,1,\dots,n$

Gesucht stückweise quadratische, interpolierende Polynome

- Stückweise Bezierkurven, z.B. fasse für $n=3$ jeweils 4

Kontrollpunkte zusammen zur Bestimmung einer Teil-Bezierkurve. Hefte Teilkurven ‚glatt‘ aneinander.



Zweidimensionale Interpolation

Problem: Gegeben Punkte (x_j, y_j) , Werte $z_j, j=0, \dots, N$

Gesucht Polynom in x und y mit $p(x_j, y_j) = z_j, j=0, \dots, N$,

z.B.
$$p(x, y) = \sum_{r=0}^R \sum_{s=0}^S a_{r,s} x^r y^s \quad \text{mit} \quad (R+1)(S+1) = N+1$$

Ergibt $N+1 \times N+1$ lineares Gleichungssystem für die unbekanntenen Koeffizienten $a_{r,s}$.

Spezialfall ‚konstant‘: $R=S=0, N=0$;
Eine Stützstelle (x_0, y_0) mit Wert z_0
Lösung: $p(x,y) \equiv z_0$



Zweidimensionale Interpolation

Spezialfall ‚bilinear‘: $R=S=1, N=3$:

4 Stützstellen, z.B. $(0,0), (1,0), (0,1), (1,1)$

bilineares Polynom $p(x,y)=a+b\cdot x+c\cdot y+d\cdot xy$



Lösung der Interpolationsaufgabe :

$$\begin{aligned} a &= z_0 \\ a + b &= z_1 \\ a + c &= z_2 \\ a + b + c + d &= z_3 \end{aligned} \Leftrightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

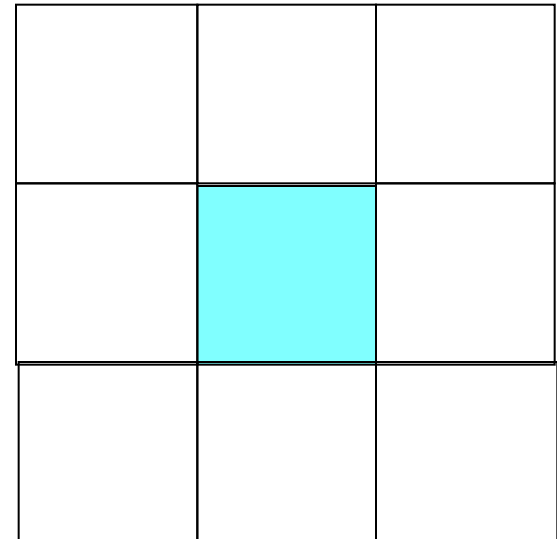
$$p(x, y) = z_0 + (z_1 - z_0)x + (z_2 - z_0)y + (z_3 + z_0 - z_1 - z_2)xy$$



Zweidimensionale Interpolation

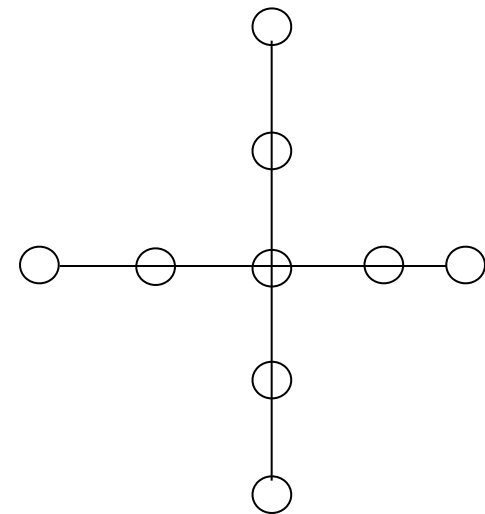
Spezialfall $R=S=3$, $N=15$,
16 Stützstellen:

bikubisch



Spezialfall $R=S=2$, $N=8$,
9 Stützstellen:

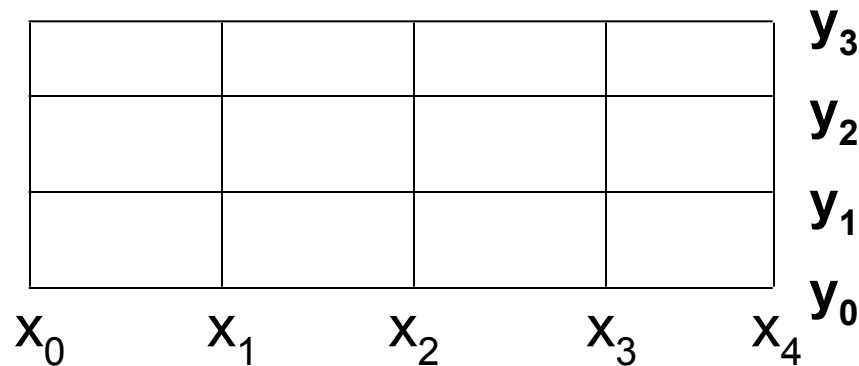
biquadratisch



Zweidimensionale Interpolation

Allg. Problemstellung: Funktion auf Rechteck.

(x_0, \dots, x_n) und (y_0, \dots, y_m) führen zu Stützstellen auf Gitter (x_j, y_k) , $j=0, \dots, n$ und $k=0, \dots, m$ mit Stützwerten $z_{j,k}$



Lösung durch 1D-Lagrange-Polynome:

$$L_{j,k}(x, y) := L_{x_j}(x) \cdot L_{y_k}(y)$$

in der Form:

$$p(x, y) = \sum_{j=0}^n \sum_{k=0}^m z_{j,k} \cdot L_{j,k}(x, y)$$



Zweidimensionale Interpolation

Andere Problemstellung: Grad $r + s \leq n$ (Gesamtgrad $\leq n$)

$$p(x, y) = \sum_{r+s \leq n} a_{r,s} x^r y^s$$

z.B. $n=2$: $p(x, y) = a + bx + cy + dx^2 + exy + fy^2$

Hier ist der Gesamtgrad $r+s$ aller Terme $x^r y^s$ durch n beschränkt!

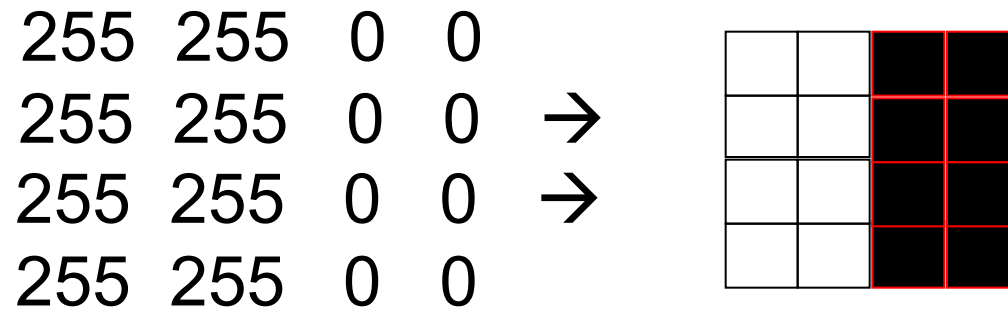
Bestimme Koeffizienten wieder aus linearem Gleichungssystem.



Beispiele aus der Bildverarbeitung

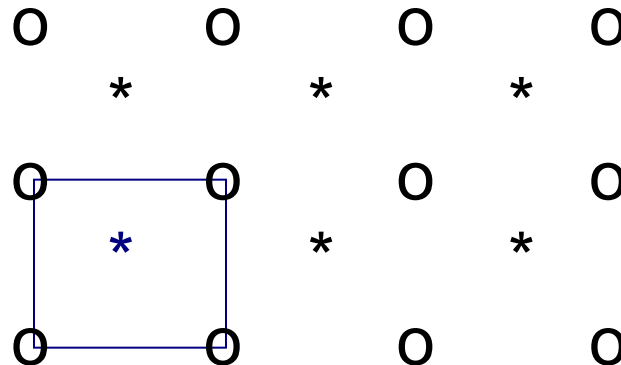
Bild als Matrix von Werten zwischen 0 und 255.

Jeder Wert gibt Grauwert an einer Stelle (einem Pixel) an.



(a) Umskalierung, Drehung, Verschiebung führt auf neues Gitter!

Wie werden Pixelwerte des neuen Gitters bestimmt?



Bestimmung des neuen Wertes *
z.B. durch bilineare Interpolation



Beispiele aus der Bildverarbeitung

(b) Registrierung von Bilddaten:

Man verfügt über zwei verschiedene Aufnahmen desselben Objekts, z.B. zu verschiedenen Zeiten, von verschiedenen Aufnahmemedien, oder von verschiedenen Blickwinkeln aus aufgenommen.

Ziel ist es, diese Bilder zur Deckung zu bringen. Z.B. eine neue Aufnahme mittels einer geeigneten Transformation (Verschiebung, Drehung) in das Referenzobjekt abzubilden.

Bestimme dazu a , b , und φ , so dass

$$\min_{\varphi, a, b} \left\| \left(\begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \right) - \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} \right\|$$



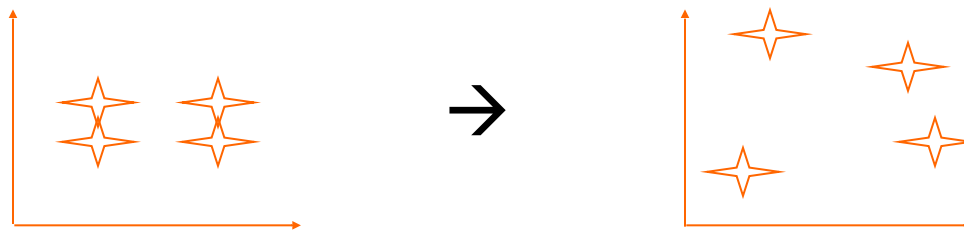
Beispiele aus der Bildverarbeitung

(c) Kalibrierung von Bilddaten:

Wieder zwei verschiedene (aber deckungsgleiche) Aufnahmen desselben Objekts.

Nun ist aber eine der Aufnahmen verzerrt, z.B. im Falle von Röntgenaufnahmen durch Erdmagnetfeld, usw.

Wir nehmen an, dass wir Markierungspunkte in beiden Aufnahmen haben, die zur Übereinstimmung gebracht werden sollen, diesmal aber mit einer nichtlinearen Transformation.



Transformiere die Koordinaten der verzerrten Markierungspunkte möglichst gut auf die originalen Positionen, z.B. durch zwei Polynome dritten Grades und Minimierung des Fehlers über alle Punkte



Beispiele aus der Bildverarbeitung

(x_d, y_d) verzerrte Position von (x_p, y_p) :

Minimiere über alle Markierungspunkte den Abstand

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} a_0 + a_1 x_d + a_2 y_d + a_3 x_d y_d + a_4 x_d^2 + a_5 y_d^2 + a_6 y_d x_d^2 + a_7 x_d y_d^2 + a_8 x_d^3 + a_9 y_d^3 \\ b_0 + b_1 x_d + b_2 y_d + b_3 x_d y_d + b_4 x_d^2 + b_5 y_d^2 + b_6 y_d x_d^2 + b_7 x_d y_d^2 + b_8 x_d^3 + b_9 y_d^3 \end{pmatrix}$$

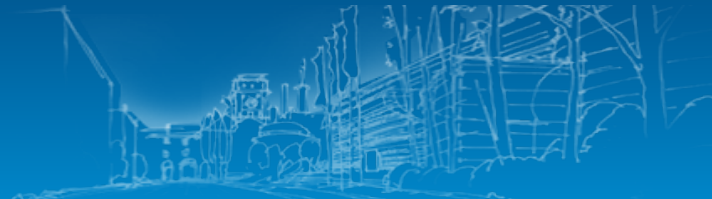
Betrachte z.B. $x_p = a_0 + a_1 x_{d,k} + \dots + a_9 y_{d,k}$ über alle

Markierungspunkte:

ergibt Normalgleichungen zur Bestimmung der optimalen Koeffizienten a_0, \dots, a_9 .

Bestimme b_0, \dots, b_9 genauso.





Interpolation und Quadratur

Quadratur

Quadratur

Integration \leftrightarrow *Flächenberechnung*

Problem:

$$I(f) := \int_a^b f(x) \, dx$$

In einigen Fällen kann das Integral mittels Stammfunktion berechnet werden:

Stammfunktion von x^k ist $x^{k+1}/(k+1)$;

Daher
$$\int_a^b x^k \, dx = \left[\frac{x^{k+1}}{k+1} \right]_a^b = \frac{b^{k+1} - a^{k+1}}{k+1}$$

Stammfunktion von $\exp(x)$ ist $\exp(x)$, usw. (Formelsammlung)
In vielen Fällen existiert keine explizite Formel für die Stammfunktion! Dann muss das Integral numerisch angenähert werden (vgl. Riemann'sche Summen, Obersumme, Untersumme)

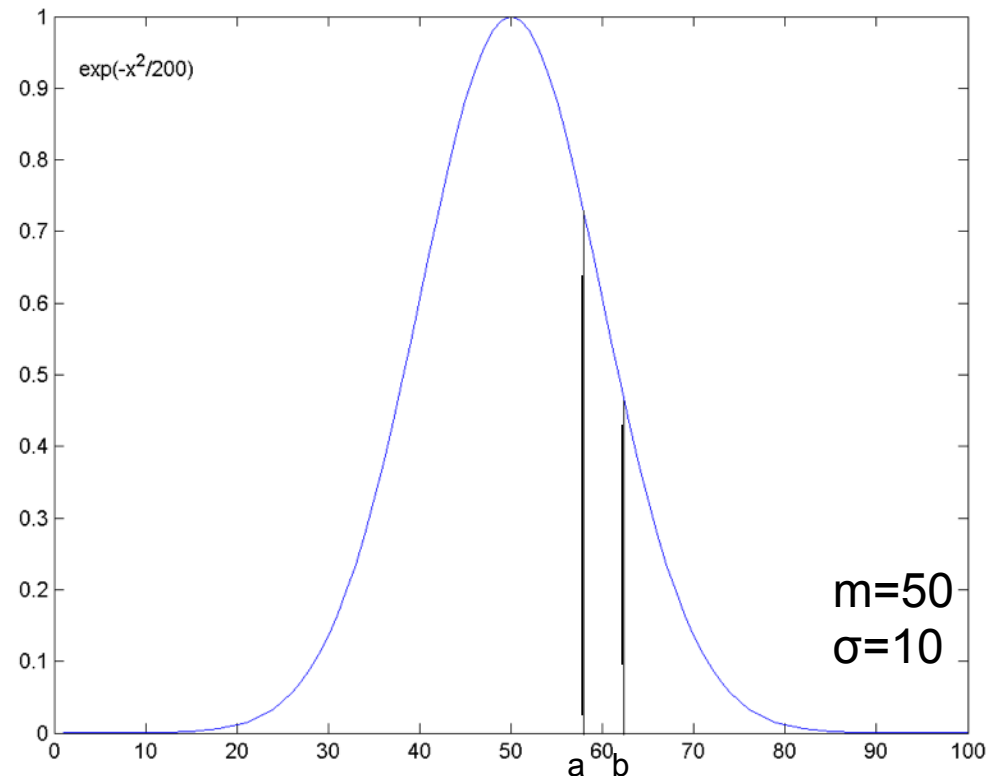


Beispiel: Wahrscheinlichkeitsrechnung

Gauss-Verteilung mit Mittelwert m und Standardabweichung σ ; gesucht ist die Wahrscheinlichkeit dafür, dass ein gemessener Wert zwischen a und b liegt:

$$\frac{1}{\sigma \sqrt{2\pi}} \cdot \int_a^b \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) dx$$

Entspricht der Fläche unter der Kurve:



Allgemeine Form

Numerische Quadraturregel zur näherungsweise Berechnung solcher Integrale:

$$I(f) = \int_a^b f(x) \, dx \approx \sum_{i=0}^n w_i f(x_i)$$

$$I(f) = \int_a^b f(x) \, dx \approx \sum_{i=0}^n w_i f(x_i)$$

- Bestandteile:
- Stützstellen x_i mit Werten $f(x_i)$
 - Gewichte w_i



Numerische Quadratur

Frage: Wie sind die Stützstellen und Gewichte zu wählen, damit der Näherungswert möglichst gut ist?

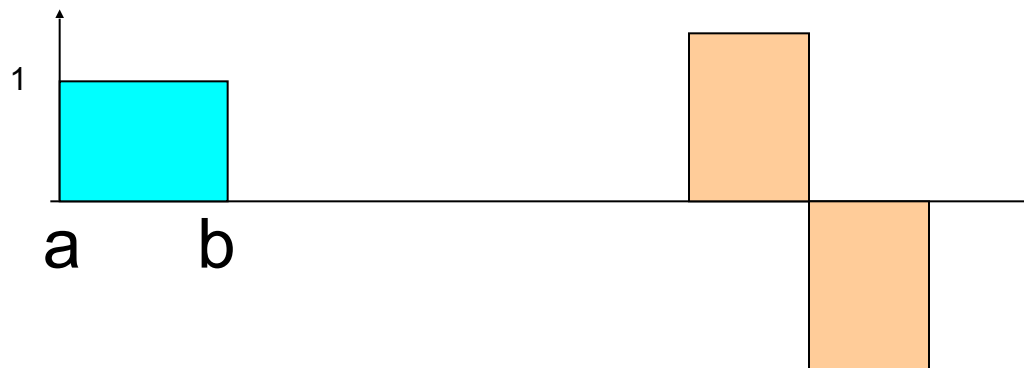
Die Gewichte sollten dabei größer oder gleich 0 sein, um Rundungsfehler durch Auslöschung in der obigen Summe zu vermeiden!

Möglichst wenige Funktionsauswertungen bei möglichst genauer Näherung!



Quadratur als Flächenberechnung

Das Integral zwischen a und b gibt den Flächeninhalt der Kurve an, die zwischen $f(x)$ und der x -Achse eingeschlossen ist.



Die blaue, linke Fläche entspricht $(b-a) \cdot 1$

Und die rosa, rechte Fläche entspricht dem Wert 0, da obere und untere Fläche sich weg heben.

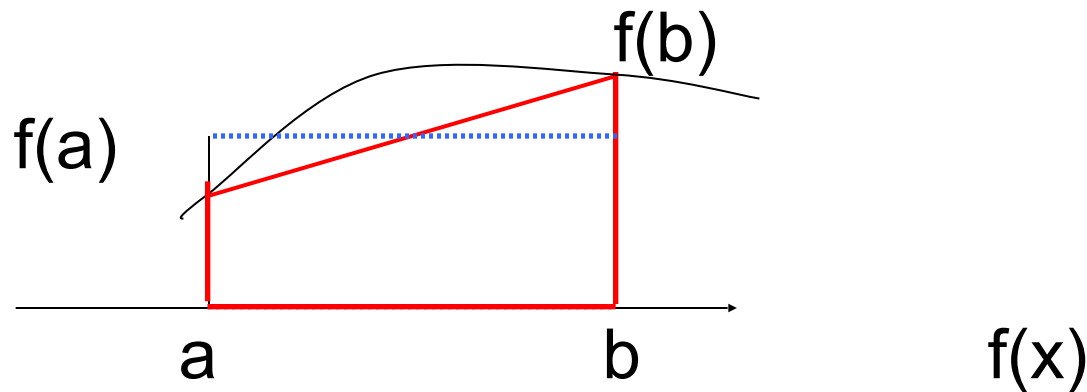


Die Trapezregel

Sei $n=1$:

$$I(f) = \int_a^b f(x) dx \approx \frac{f(a) + f(b)}{2} \cdot (b - a) =$$
$$= \frac{b - a}{2} f(a) + \frac{b - a}{2} f(b)$$

Stützstellen a, b ; Gewichte $w_0 = w_1 = (b-a)/2$

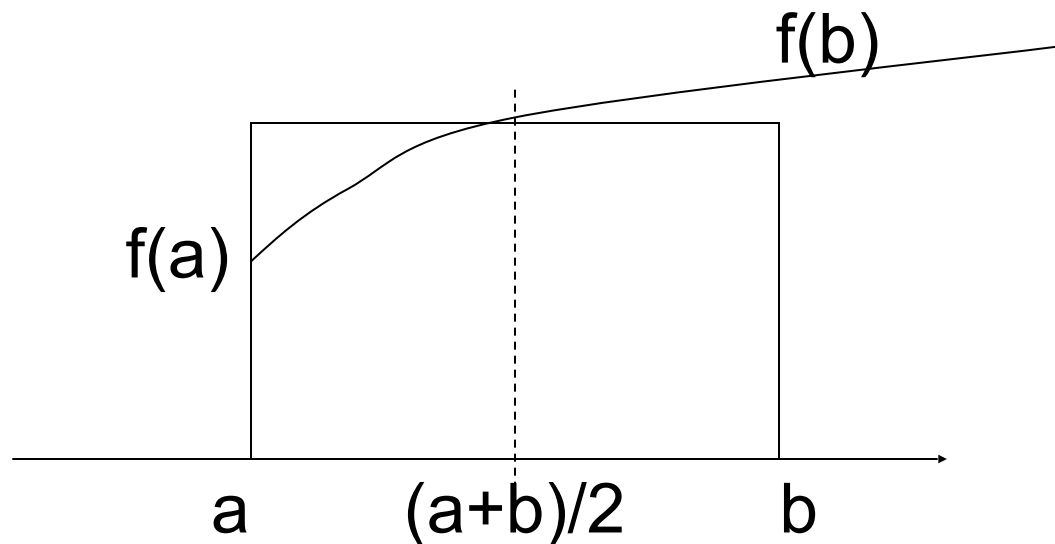


Die Mittelpunktsregel

Sei $n=0$: Betrachte das Rechteck, begrenzt durch die Punkte $(a,0)$, $(b,0)$ und $((a+b)/2, f((a+b)/2))$

$$I(f) \approx (b - a) \cdot f\left(\frac{a + b}{2}\right)$$

Also mit $n=0$ nur eine Stützstelle $(a+b)/2$ und ein Gewicht $w_0 = (b-a)$.



Regeln aus der Interpolation

Idee: Nähere $f(x)$ durch einfach zu integrierende Funktion an, z.B. durch interpolierendes Polynom.

Der Einfachheit halber wählen wir äquidistante Stützstellen

$$x_i = a + ih, \quad i = 0, 1, \dots, n, \quad h = (b - a)/n$$

Zu bestimmen sind noch passende Gewichte w_i . Sie ergeben sich aus der Integration des $f(x)$ interpolierenden Polynoms in der Lagrangedarstellung:

$$p(x) = \sum_{i=0}^n f(x_i)L_i(x)$$

$$\text{also } \int_a^b f(x)dx \approx \int_a^b p(x)dx = \sum_{i=0}^n f(x_i) \int_a^b L_i(x)dx = \sum_{i=0}^n f(x_i)w_i$$



Regeln aus der Interpolation

Dadurch lassen sich allgemein Integrationsregeln herleiten.

Für $n > 6$ ergeben sich auch negative Gewichte!
Numerisch problematisch (Auslöschung).

Daher ist dieses Vorgehen nur für kleine n sinnvoll.



Beispiel

Sei: $n=1$, nur $x_0=a$, $x_1=b$, $h=b-a$:

Dann ist das linear interpolierende Polynom:

$$p(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$$

Daher ergibt sich als Näherung:

$$\begin{aligned} \int_a^b f(x) dx &\approx \int_a^b p(x) dx = \int_a^b \left(f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \right) dx = \\ &= f(a)(b - a) + \frac{f(b) - f(a)}{b - a} \left(\frac{b^2 - a^2}{2} - a(b - a) \right) = \\ &= h \left(\frac{1}{2} f(a) + \frac{1}{2} f(b) \right) = \frac{b - a}{2} (f(a) + f(b)) \end{aligned}$$

Also Gewichte $w_0 = w_1 = h/2 = (b-a)/2$. Dies ist die Trapezregel!



Fehlerabschätzung

$$\begin{aligned} \left| \int_a^b (f(x) - p(x)) dx \right| &= \left| \int_a^b \left(\frac{f^{(2)}(\xi)}{2} (x-b)(x-a) \right) dx \right| \leq \\ &\leq \frac{M_2}{2} \int_a^b (x-a)(b-x) dx = \frac{M_2}{2} \cdot \frac{(b-a)^3}{6} = \frac{M_2(b-a)^3}{12} = \frac{M_2}{12} h^3 \end{aligned}$$

mit $M_k := \max_{x \in [a,b]} |f^{(k)}(x)|$

Daher ist die Trapezregel exakt für Polynome vom Grad 1 (da dann $M_2 = 0$),

d.h. Näherungswert und exakter Wert sind gleich falls $f(x)$ eine Polynom vom Grad ≤ 1 ist.



Die Simpson-Regel

$$x_0 = a, \quad x_1 = \frac{a+b}{2}, \quad x_2 = b, \quad h = \frac{b-a}{2}$$

$$\Rightarrow w_0 = w_2 = \frac{h}{3} = \frac{b-a}{6}, \quad w_1 = \frac{4}{3}h = \frac{4(b-a)}{6}$$

$$I(f) \approx \frac{h}{3} \cdot \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

Fehlerabschätzung (ohne Beweis):

$$\left| \int_a^b (f(x) - p(x)) dx \right| \leq \frac{M_4 (b-a)^5}{2880} = \frac{M_4}{90} h^5$$

Daher ist die Simpsonregel sogar exakt für Polynome dritten Grades (weil dann $M_4 = 0$).



Quadratur mittels Interpolation

Im Allgemeinen lautet die Quadraturregel basierend auf interpolierenden Polynomen:

$$\int_a^b f(x)dx \approx \int_a^b p_{0,\dots,n}(x) = h \sum_{i=0}^n \alpha_i f(a + ih)$$

Aber: für $n > 6$ treten aber auch negative Gewichte $h\alpha_i$ auf (s.o.), daher ist dies nicht geeignet für großes n .



Gauß-Quadratur

Bis jetzt waren die Stützstellen vorgegeben (äquidistant), und nur die Gewichte wurden optimal gewählt.

Aus Interpolation wissen wir aber, dass am Rand mehr Stützstellen sein sollten!

Daher neue Problemstellung:

Wie sind Gewichte und Stützstellen zu wählen, dass sich eine möglichst ‚gute‘ Quadratur-Regel ergibt.

‚Gut‘ heisst: die Regel soll Polynome möglichst hohen Grades exakt integrieren!

(Trapezregel: 2 Stützstellen, Grad 1 exakt,

Simpsonregel: 3 Stützstellen, Grad 3 exakt)



Gauss-Quadratur

Finde $x_i \in [a, b]$, w_i , $i=0, \dots, n$, so dass für $j=0, \dots, m$:

$$\frac{b^{j+1} - a^{j+1}}{j+1} = \int_a^b x^j dx = I(x^j) = \sum_{i=0}^n w_i x_i^j$$

Dies sind $m+1$ nichtlineare Gleichungen für die $2n+2$ Unbekannten x_i und w_i .



Gauß-Quadratur

Fallstudie n=1:

$$\frac{b^{j+1} - a^{j+1}}{j+1} = \int_a^b x^j dx \stackrel{!}{=} w_0 x_0^j + w_1 x_1^j$$

$j = 0 :$	$b - a$	$=$	$w_0 + w_1$
$j = 1 :$	$\frac{b^2 - a^2}{2}$	$=$	$w_0 x_0 + w_1 x_1$
$j = 2 :$	$\frac{b^3 - a^3}{3}$	$=$	$w_0 x_0^2 + w_1 x_1^2$
$j = 3 :$	$\frac{b^4 - a^4}{4}$	$=$	$w_0 x_0^3 + w_1 x_1^3$
$(j = 4 :)$	$\frac{b^5 - a^5}{5}$	$=$	$w_0 x_0^4 + w_1 x_1^4$

Aus den ersten vier Gleichungen ergibt sich (Lösung per Hand)

$$x_{0,1} = \frac{a+b}{2} \pm \frac{b-a}{2\sqrt{3}}, \quad w_{0,1} = \frac{b-a}{2}$$



Gauß-Quadratur

Gleichung $j=4$ ist mit diesen Werten nicht erfüllt!

Also werden durch diese Parameter Polynome bis zum Grad 3 exakt integriert.

Im Spezialfall: $\int_{-1}^1 f(x) \frac{dx}{\sqrt{1-x^2}}$ ergeben sich als optimale Stützstellen, die Polynome möglichst hohen Grades exakt integrieren, wieder die Nullstellen der Tchebycheff-Polynome

$$\cos\left(\frac{(2j+1)\pi}{2n+2}\right), j=0,1,\dots,n$$

Allgemein lässt sich zeigen, dass die optimalen Stützstellen gerade die Nullstellen von Orthogonalpolynomen sind, z.B. Legendre-, Hermite-, Laguerre-Polynome.
(siehe Formelsammlung)



Gauß-Quadratur

Vorteil der Gauss-Quadratur mit optimalen Stützstellen :

Für festes n ergibt sich optimales Ergebnis!

Nachteil: Falls Resultat nicht genau genug, will man mehr Stützstellen verwenden; die neuen Stützstellen sind dann aber Nullstellen eines anderen Polynoms, also muss man an allen Stellen $f(x_i)$ neu berechnen.

Ausweg: Behalte die n alten Stützstellen. Suche nun n neue Stützstellen und $2n$ Gewichte zu den $2n$ Stützstellen, so dass Polynome möglichst hohen Grades exakt integriert werden.



Monte-Carlo-Methoden

Idee: Wähle n Zufallspunkte aus dem Gebiet $Q \in \mathbb{R}^d$,
berechne Funktionswerte an diesen Stellen und setze

$$I = \int_Q f(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^n f(\mathbf{x}_i) \frac{|Q|}{n} \quad \mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$$

Besonders nützlich bei hochdimensionalen Integralen!

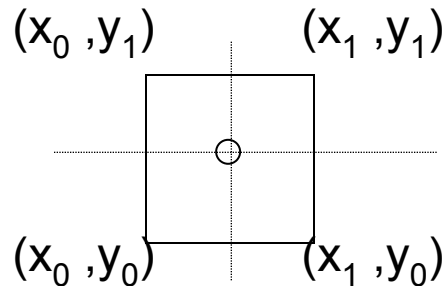
Stützstellen: Zufallsstellen $f(\mathbf{x}_i)$

Gewichte: immer $|Q| / n$.



Zweidimensionale Integration

Mittelpunktsregel:



$$\begin{aligned} \int_Q f(x, y) dx dy &\approx (y_1 - y_0) \int_{x_0}^{x_1} f\left(x, \frac{y_0 + y_1}{2}\right) dx \approx \\ &\approx (x_1 - x_0)(y_1 - y_0) f\left(\frac{x_1 + x_0}{2}, \frac{y_1 + y_0}{2}\right) \end{aligned}$$



Zweidimensionale Integration

Allgemein Quadratur-Regeln aus der 2D-Interpolation:

Bestimme zu $f(x,y)$ ein interpolierendes Polynom $p(x,y)$, so dass Stützstellen und Grad zusammenpassen

Gesucht: Integral über Quadrat Q mit Ecken (x_0, y_0) , (x_0, y_1) , (x_1, y_0) , (x_1, y_1)

$$\begin{aligned} \int_Q f(x, y) dx dy &\approx \int_Q p(x, y) dx dy = \int_Q \sum_{r,s} a_{r,s} x^r y^s dx dy = \\ &= \sum_{r,s} a_{r,s} \int_{x_0}^{x_1} x^r dx \int_{y_0}^{y_1} y^s dy = \sum_{r,s} \frac{a_{r,s} (x_1^{r+1} - x_0^{r+1}) (y_1^{s+1} - y_0^{s+1})}{(r+1)(s+1)} \end{aligned}$$

Andere Möglichkeit: Lagrange-Ansatz wie oben.

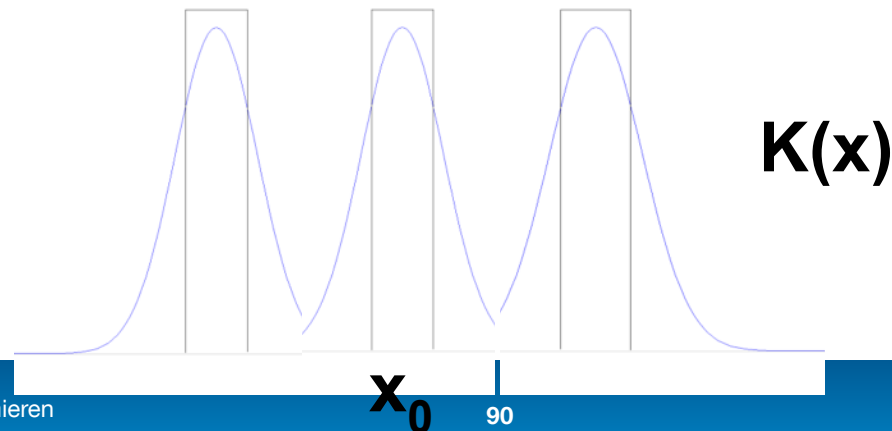


Gaußscher Weichzeichner

Anwendung: Deblurring bei verschmierten Bildern

Modell: Jeder Pixelpunkt wird entsprechend einer Gauss-Verteilung, verschmiert, z.B. Aufnahme durch Atmosphäre. (Zur Vereinfachung hier nur eindimensional)

An der Stelle x_0 wird der ursprüngliche Wert $f(x_0)$ ersetzt durch gestörten Wert $g(x_0)$, der sich aus der Überlagerung von Nachbarpunkte ergibt.



Gaußscher Weichzeichner

Das beobachtete Bild g entsteht durch eine ‚Faltung‘ des ursprünglichen Bildes f mit dem Gauss-Kern $K(x)=\exp(-x^2/(2\pi\sigma^2))$, der genau die obige Funktion beschreibt.

Um ein diskretes Modell zu erhalten, ersetze man das Faltungs-Integral an jeder diskreten Stelle x_j z.B. durch die Mittelpunktsumme (oder Trapezsumme) \rightarrow

$$g(x_j) = \int \exp\left(-\frac{(x_j - t)^2}{2\pi\sigma^2}\right) \cdot f(t) dt \approx \\ \approx h \sum_k \exp\left(-\frac{(x_j - x_k)^2}{2\pi\sigma^2}\right) \cdot f(x_k)$$

Dies führt auf ein lineares Gleichungssystem, das die Beziehung zwischen ursprünglichen und gestörten Werten beschreibt



Gaußscher Weichzeichner

$$\left(\exp \left(-\frac{(j-k)^2}{2\pi\sigma^2 n^2} \right) \right)_{j,k=1}^n \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

$$x_j - x_k = \frac{j-k}{n}$$

da alle
 x_1, \dots, x_n
äquidistant sind

Durch Lösung des Gleichungssystems kann man aus den gestörten Daten $g(x_j)$ das Originalbild $f(x_k)$ wieder gewinnen.

Aber das Gleichungssystem $K\mathbf{f} = \mathbf{g}$

ist sehr schlecht konditioniert.

Daher verwendet man zusätzlich Regularisierung, z.B.:

$$(K^T K + \rho^2 I)\mathbf{f} = K^T \mathbf{g}$$



Glättung

Die Beziehung

$$\left(\exp \left(-\frac{(j-k)^2}{2\pi\sigma^2 n^2} \right) \right)_{j,k=1}^n \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

beschreibt eine Glättung des Signals f . Dies wird z.B. bei der Rauschunterdrückung verwendet.

Intuitiv: “Verwasche“ die Pixel durch Mittelwertbildung mit Nachbarn,

$$\text{z.B. } x_k \rightarrow (x_{k-1} + 2x_k + x_{k+1}) / 4$$

