

2. Kurvenanpassung

2. Kurvenanpassung

2.1 Approximation

2.2 Interpolation

2.3 Kennfeldinterpolation

2 Kurvenanpassung

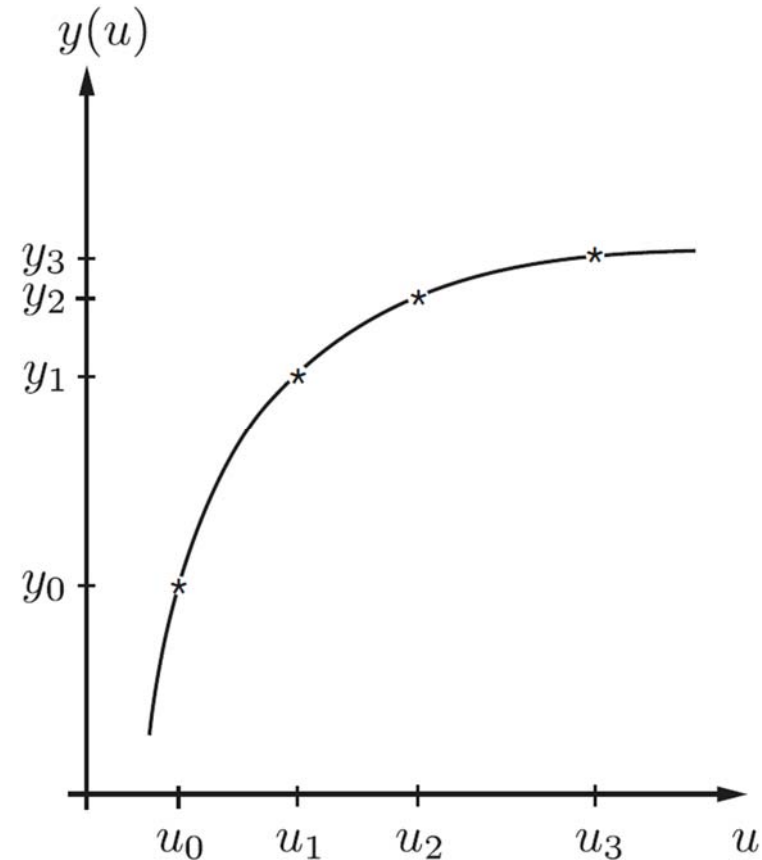
- Analytische Darstellung einer Messkennlinie erfordert Modellbildung: Wie lässt sich der Zusammenhang der Größen beschreiben?
- Tatsächliches Modell ist in der Praxis oft unbekannt, daher liegt Messkennlinie nicht in analytischer Form, sondern nur als Menge von n Messpunkten (u_k, y_k) , $k \in \{0, \dots, n - 1\}$ vor
- Gesucht: analytische Form der Messkennlinie, welche die gemessenen Punkte geeignet berücksichtigt
- Dadurch auch Bestimmung von Zwischenwerten (u, k) möglich
- Weitere Anpassungen der Messkennlinie: siehe Kap. 3

2 Kurvenanpassung

- Zwei Ansätze zur Konstruktion einer analytischen Kennlinie:

- **Interpolation:** analytische Kennlinie verläuft exakt durch alle Messpunkte

- Z. B. mittels Polynomen: $y = p(u)$:
Benötigt werden dazu bei n Messpunkten Polynome i. a. vom Grad $\deg\{p(u)\} = n - 1$
- Daher nur für wenige Messpunkte sinnvoll (d. h. kleine n)
- Bei großen n : Anfälligkeit für oszillierendes Verhalten
- Störungen an einem Messpunkt beeinflussen direkt die Kennlinie an dieser Stelle
- Siehe Kap. 2.2



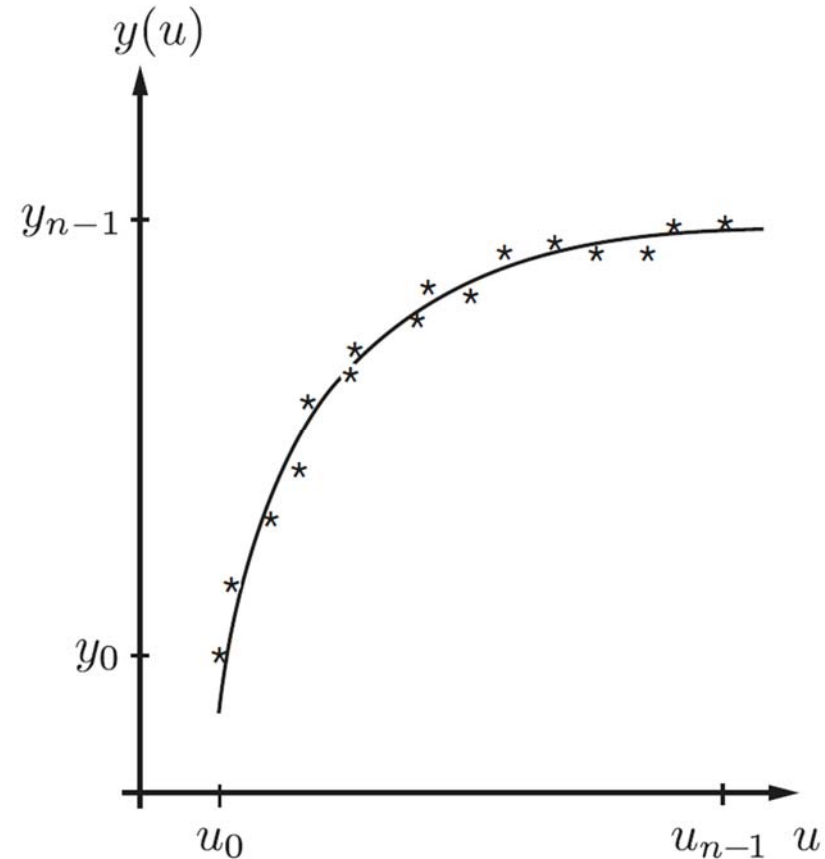
Bildquelle: F. Puente León: Messtechnik, 10. Auflage, Springer, 2015

© Michael Heizmann, IIT, KIT, alle Rechte einschließlich Kopier- und Weitergaberechte bei uns.

2 Kurvenanpassung

- Zwei Ansätze zur Konstruktion einer analytischen Kennlinie:

- **Approximation:** Suche nach einer möglichst einfachen Funktion, welche die Messpunkte möglichst gut berücksichtigt
 - Ansatz: Fehler zwischen Messpunkten und der analytischen Funktion minimieren
 - Auch für große n gut anwendbar
 - Siehe Kap. 2.1



Bildquelle: F. Puente León: Messtechnik, 10. Auflage, Springer, 2015

© Michael Heizmann, IIT, KIT, alle Rechte einschließlich Kopier- und Weitergaberechte bei uns.

2.1 Approximation

- Approximation mittels einer endlichen Reihe analytischer Funktionen $\varphi_i(u)$, $i \in \{0, \dots, m-1\}$:

$$\hat{y}(u) = \sum_{i=0}^{m-1} a_i \varphi_i(u)$$

- An den Messpunkten gilt: $\hat{y}_k = \hat{y}(u_k) = \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \approx y_k$,
d.h. Approximation muss die Messpunkte nicht exakt treffen
- Bestimmung der Koeffizienten a_i : Minimierung eines Gütemaßes Q :
z. B. Summe der Approximationsfehlerquadrate
(siehe Least-Squares-Schätzer):

$$Q = \sum_{k=0}^{n-1} (y_k - \hat{y}_k)^2 = \sum_{k=0}^{n-1} \left(y_k - \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \right)^2 \rightarrow \min$$

- Vorteil: Bildung einer analytischen Kennlinie bereits mit wenigen Basisfunktionen $\varphi_i(u)$, d. h. kleinen m , möglichst: $m \ll n$

Approximation mit orthonormalen Funktionensystemen

- Wiederholung orthonormale Funktionensysteme:
 - Beispiel: Vektoren im dreidimensionalen Raum \mathbb{R}^3 :
 - Repräsentation durch Komponenten in den drei Raumrichtungen
 $\mathbf{a} = (a_0, a_1, a_2)^T$
 - Mögliche Basis: Einheitsvektoren
 $\mathbf{e}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{e}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{e}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
 - $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2\}$ bildet orthonormale Basis:
$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \sum_{k=0}^{n-1} e_{i,k} \cdot e_{j,k} = \delta_i^j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$
 - Vorteil: Erweiterung um eine weitere Dimension erfordert nur Ergänzung einer Komponenten (z. B. a_n) und ändert nicht die bisherigen Komponenten
 - Damit Darstellung des Vektors \mathbf{a} :
 $\mathbf{a} = a_0 \mathbf{e}_0 + a_1 \mathbf{e}_1 + a_2 \mathbf{e}_2 = \sum_{i=0}^2 a_i \mathbf{e}_i$

Approximation mit orthonormalen Funktionensystemen

- Anwendung für Messkennlinien: Orthonormale Funktionensysteme:

$$\langle \varphi_i, \varphi_j \rangle = \sum_{k=0}^{n-1} \varphi_i(u_k) \cdot \varphi_j^*(u_k) = \delta_i^j$$

d. h. Orthonormalität an den Stützstellen: Funktionswerte $\varphi_i(u_k)$ an den Stützstellen u_k sind orthonormale, n -dimensionale Vektoren

- Im Folgenden Annahme: Stützstellen sind äquidistant verteilt
- Gütemaß:

$$Q = \sum_{k=0}^{n-1} \left(y_k - \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \right) \left(y_k - \sum_{j=0}^{m-1} a_j \varphi_j(u_k) \right)^* \rightarrow \min$$

Approximation mit orthonormalen Funktionensystemen

- Bestimmung der Koeffizienten a_j :

Ableitung des Gütemaßes nach a_j :

$$\begin{aligned}\frac{\partial Q}{\partial a_j} &= -2 \sum_{k=0}^{n-1} \left(y_k - \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \right) \varphi_j^*(u_k) \\ &= -2 \underbrace{\left(\sum_{k=0}^{n-1} y_k \varphi_j^*(u_k) - \left(\sum_{k=0}^{n-1} \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \varphi_j^*(u_k) \right) \right)} = 0\end{aligned}$$

$$\sum_{k=0}^{n-1} y_k \varphi_j^*(u_k) - \underbrace{\left(\sum_{i=0}^{m-1} a_i \sum_{k=0}^{n-1} \varphi_i(u_k) \varphi_j^*(u_k) \right)}_{= \delta_i^j} = 0$$

$$\Rightarrow a_j = \sum_{k=0}^{n-1} y_k \varphi_j^*(u_k)$$

D. h. Koeffizienten a_j hängen nur von den zugehörigen Basisfunktionen $\varphi_j(u_k)$ ab, auch bei Zufügung weiterer Basisfunktionen

2.1 Approximation

Approximation mit orthonormalen Funktionensystemen

- Resultierender quadratischer Fehler:

$$\begin{aligned} Q &= \sum_{k=0}^{n-1} \left(y_k - \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \right) \left(y_k - \sum_{j=0}^{m-1} a_j \varphi_j(u_k) \right)^* \\ &= \sum_{k=0}^{n-1} y_k^2 - \sum_{i=0}^{m-1} a_i \underbrace{\sum_{k=0}^{n-1} \varphi_i(u_k) y_k^*}_{= a_i^*} - \sum_{j=0}^{m-1} a_j^* \underbrace{\sum_{k=0}^{n-1} \varphi_j^*(u_k) y_k}_{= a_j} \\ &\quad + \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_i a_j^* \underbrace{\sum_{k=0}^{n-1} \varphi_i(u_k) \varphi_j^*(u_k)}_{= \delta_i^j} \end{aligned}$$

$$Q = \sum_{k=0}^{n-1} y_k^2 - \sum_{i=0}^{m-1} |a_i|^2$$

D. h. mit wachsendem Grad m der Funktionenreihe wird der Approximationsfehler geringer

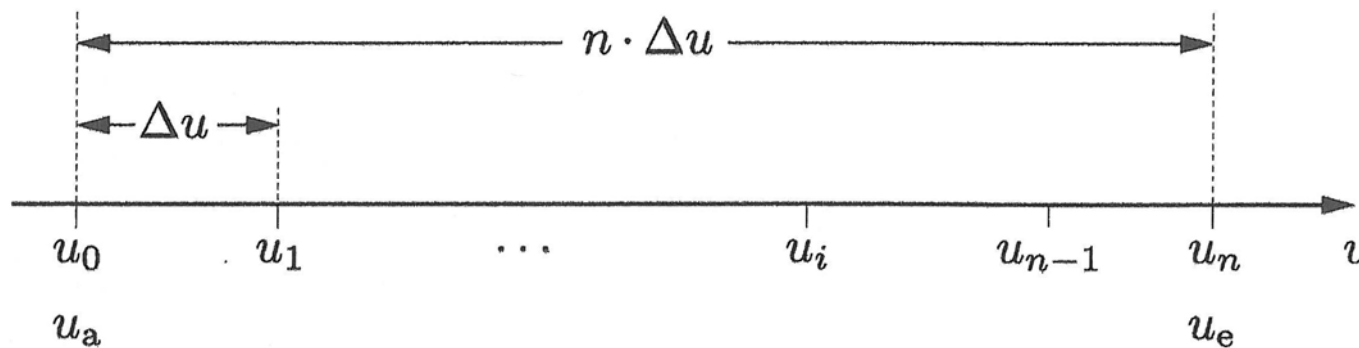
2.1 Approximation

Approximation mit der Fourier-Reihe

- Fourier-Reihe mit periodischen Funktionen:

$$F_i(u) = \frac{1}{\sqrt{n}} \cdot \exp\left(j2\pi i \frac{u - u_a}{u_e - u_a}\right)$$

- Messbereich $[u_a, u_e]$ mit n äquidistanten Stützstellen im Abstand Δu :



- Die Fourier-Reihe ist ein orthonormales Funktionensystem in diesem Messbereich:

$$\langle F_i(u_k), F_j(u_k) \rangle = \frac{1}{n} \sum_{k=0}^{n-1} \exp\left(j2\pi i \frac{u_k - u_a}{u_e - u_a}\right) \exp\left(-j2\pi j \frac{u_k - u_a}{u_e - u_a}\right)$$

2.1 Approximation

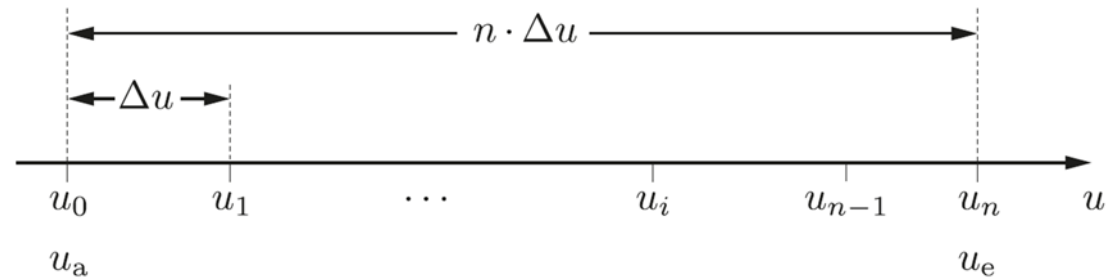
Approximation mit der Fourier-Reihe

- Beweis der Orthonormalität:

- Stützstellenabstand Δu , Intervallbreite

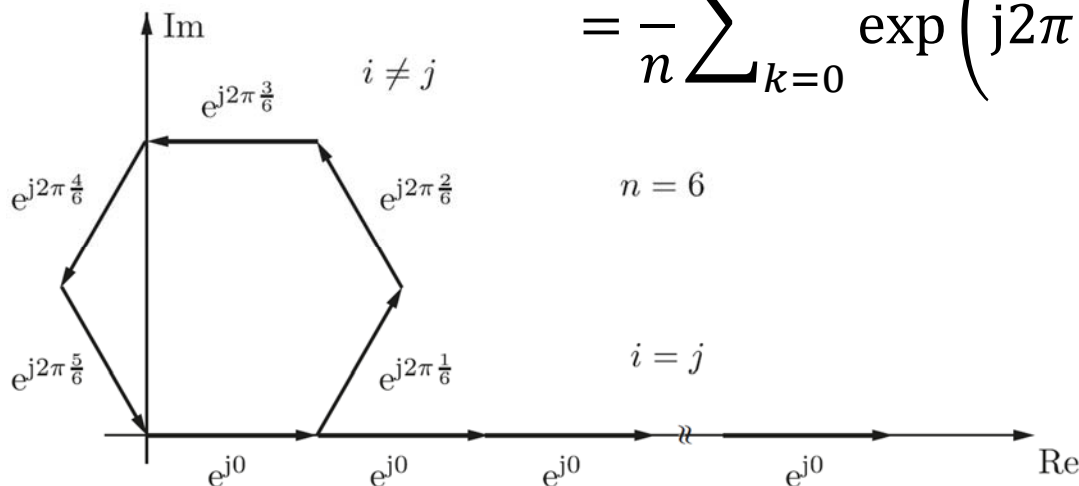
$$u_e - u_a = n \cdot \Delta u:$$

$$u_k = u_a + k \cdot \Delta u \Rightarrow \frac{u_k - u_a}{u_e - u_a} = \frac{k}{n}$$



- Eingesetzt:

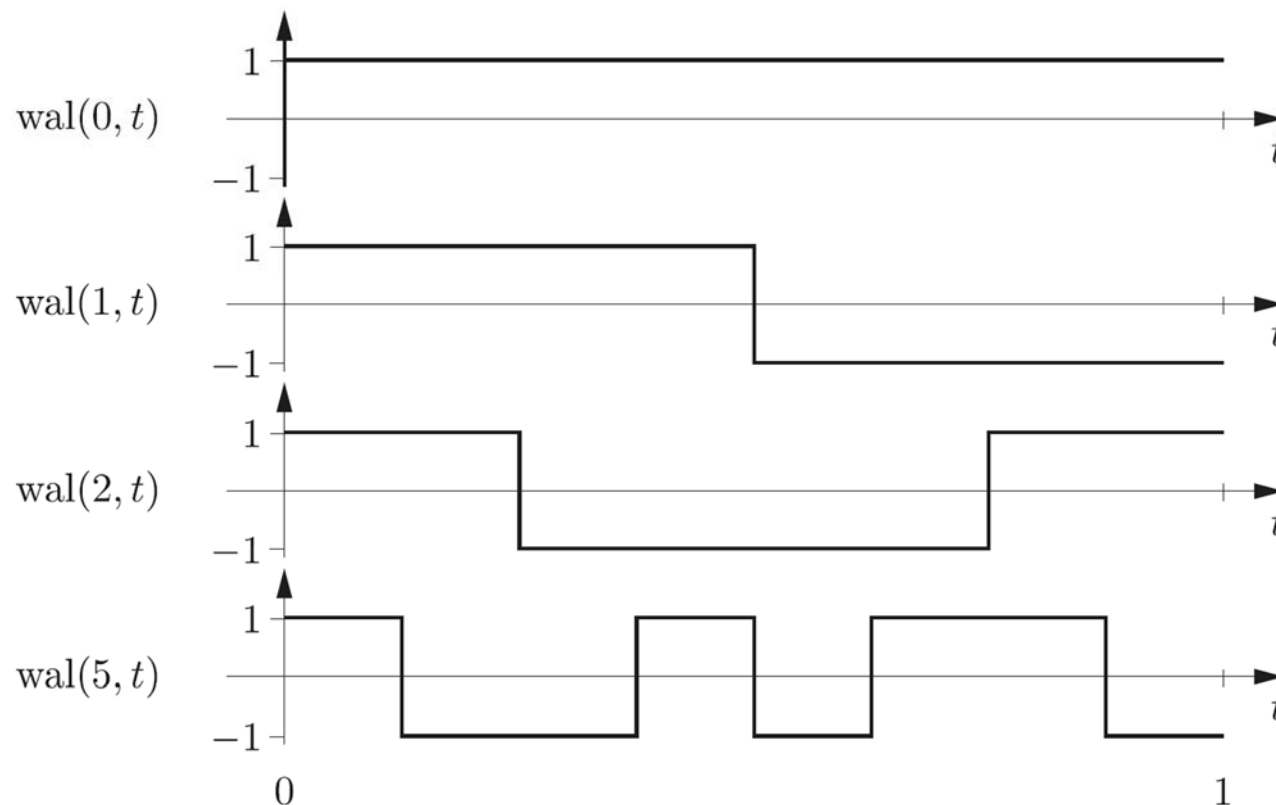
$$\begin{aligned} \langle F_i(u_k), F_j(u_k) \rangle &= \frac{1}{n} \sum_{k=0}^{n-1} \exp\left(j2\pi i \frac{u_k - u_a}{u_e - u_a}\right) \exp\left(-j2\pi j \frac{u_k - u_a}{u_e - u_a}\right) \\ &= \frac{1}{n} \sum_{k=0}^{n-1} \exp\left(j2\pi(i - j) \frac{k}{n}\right) = \delta_i^j \end{aligned}$$



2.1 Approximation

Approximation mit Walsh-Funktionen

- Nachteil der Fourier-Reihe: komplexe Exponentialfunktionen
- Einfacheres orthonormales Funktionensystem: Walsh-Funktionen
 $wal(i, u)$: Definitionsbereich $[0,1]$, Funktionswerte -1 und $+1$
- Regeln zur Erzeugung der Walsh-Funktionen: siehe Literatur
- Walsh-Funktionen: verallgemeinerte Frequenz, siehe Kap. 8

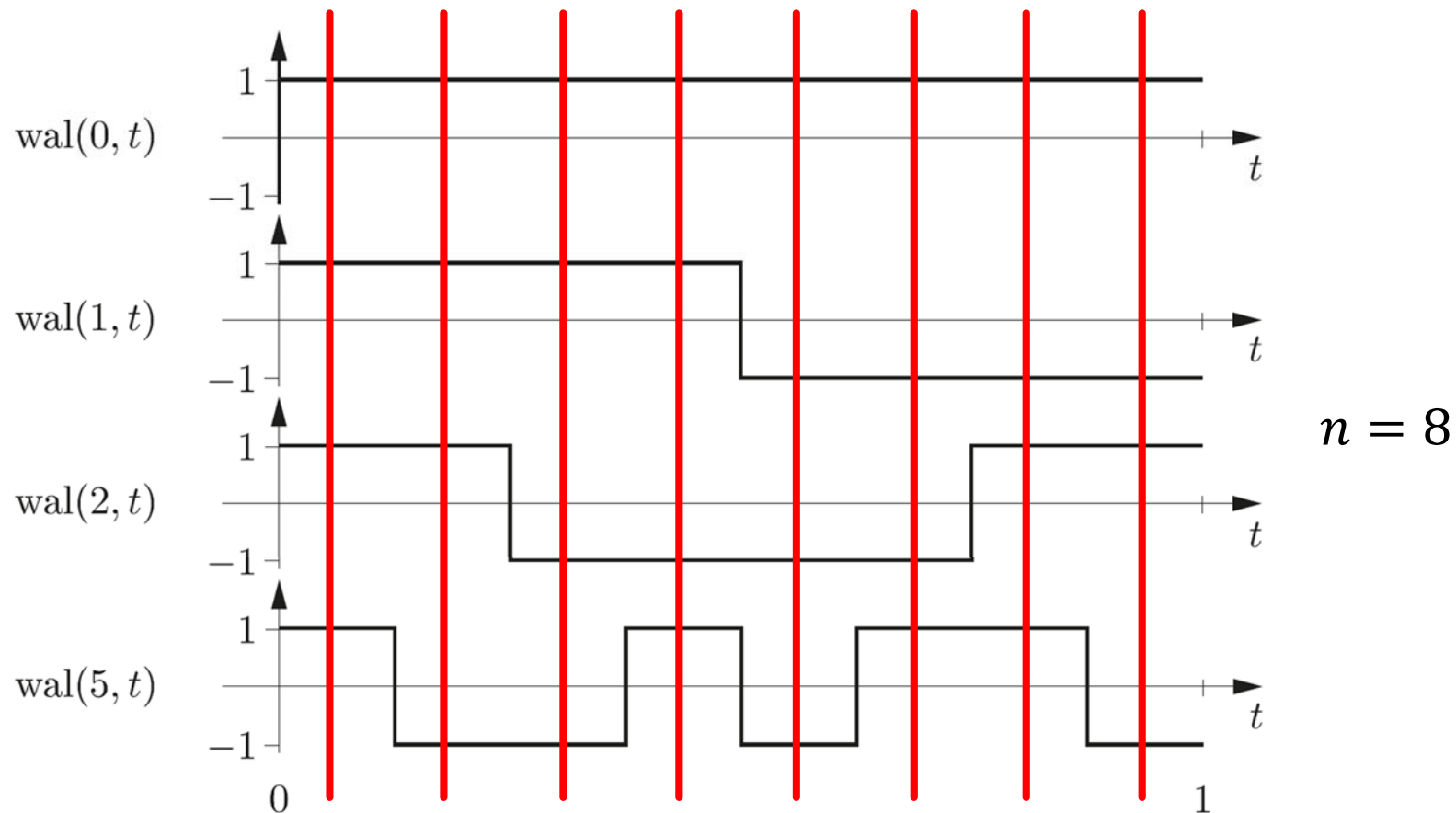


2.1 Approximation

Approximation mit Walsh-Funktionen

- Orthonormalität: Einfach zu erkennen durch Summation über äquidistant verteilte Stützstellen:

$$\langle \text{wal}(i, u), \text{wal}(j, u) \rangle = \sum_{k=0}^{n-1} \text{wal}(i, u_k) \cdot \text{wal}(j, u_k) = \delta_i^j$$



Least-Squares-Schätzer

- Approximation ohne Verwendung orthonormaler Basisfunktionen
- Ziel: Minimierung der quadratischen Summe der Approximationsfehler (Least-Squares-Fehler):

$$Q = \sum_{k=0}^{n-1} (y_k - \hat{y}_k)^2 = \sum_{k=0}^{n-1} \left(y_k - \sum_{i=0}^{m-1} a_i \varphi_i(u_k) \right)^2$$
$$= (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) \rightarrow \min$$

- Vektorieller Ansatz für n Messpunkte:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_0 \\ \vdots \\ \hat{y}_{n-1} \end{bmatrix} = \begin{bmatrix} \varphi_0(u_0) & \cdots & \varphi_{m-1}(u_0) \\ \vdots & \ddots & \vdots \\ \varphi_0(u_{n-1}) & \cdots & \varphi_{m-1}(u_{n-1}) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix} = \mathbf{\Phi} \mathbf{a}$$

- Eingesetzt in Least-Squares-Fehler:

$$Q = (\mathbf{y} - \hat{\mathbf{y}})^T (\mathbf{y} - \hat{\mathbf{y}}) = (\mathbf{y} - \mathbf{\Phi} \mathbf{a})^T (\mathbf{y} - \mathbf{\Phi} \mathbf{a})$$
$$= \mathbf{y}^T \mathbf{y} - 2\mathbf{a}^T \mathbf{\Phi}^T \mathbf{y} + \mathbf{a}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{a} \rightarrow \min$$

- Notwendige Bedingung für Minimum:

$$\frac{dQ}{d\mathbf{a}} = -2\mathbf{\Phi}^T \mathbf{y} + 2\mathbf{\Phi}^T \mathbf{\Phi} \mathbf{a} = 0 \Rightarrow \mathbf{a} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{y}$$

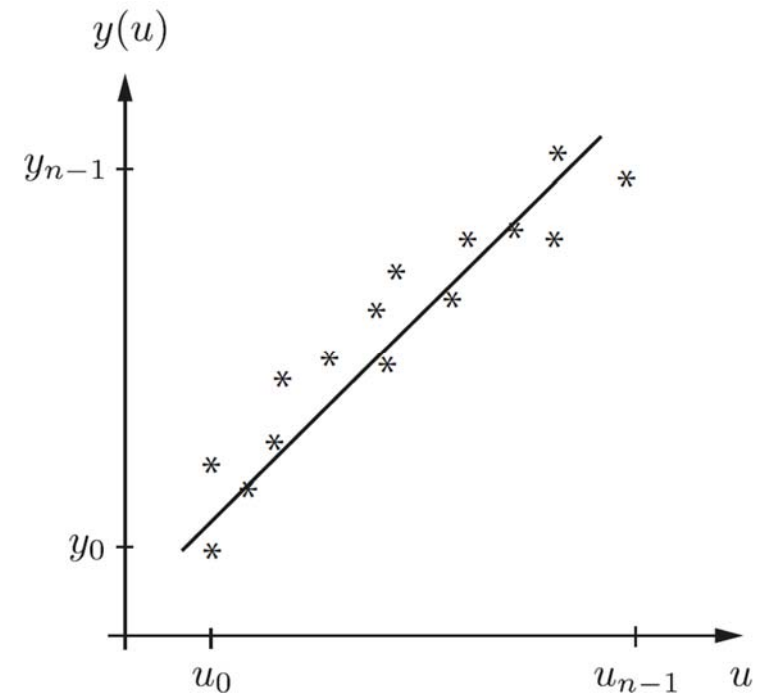
Least-Squares-Schätzer

- Gesuchter Parametervektor ist also: $\mathbf{a} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$
mit der Pseudoinversen (Moore-Penrose-Pseudoinversen)
 $(\Phi^T \Phi)^{-1} \Phi^T$
- Bedeutung des LS-Schätzers:
 - Wird oft benutzt, um aus stark verrauschten Messungen Kennlinienfunktionen zu bestimmen: Regressionsanalyse (s. unten)
 - Optimalfilter (siehe Kap. 6): Bestmögliche Approximation für die gegebenen Funktionen $\varphi_i(u)$
 - Berücksichtigung von Vorwissen: Signalmodell, bestimmt durch Wahl der Funktionen $\varphi_i(u)$ und damit Aufbau der Matrix Φ
 - Funktionaler Zusammenhang $\hat{y} = \sum_{i=0}^{m-1} a_i \varphi_i(u)$:
damit wird Extrapolation (Vorhersage von Werten außerhalb des Messbereichs, z. B. künftige Werte eines Zeitsignals) möglich,
mögliche Anwendung: Prozessüberwachung (siehe Kap. 4)
 - Anwendungen zur Parameterschätzung (siehe z. B. Kap. 8.2)

2.1 Approximation

Regressionsanalyse

- Ziel: Herstellung eines funktionalen Zusammenhangs zwischen Variablen
- Meist Verwendung von Polynomen als Funktionen und Minimierung der quadratischen Summe der Approximationsfehler (Least-Squares-Fehler), s. o.; auch andere Modell-Funktionen und Optimierungsziele können verwendet werden
- Im Folgenden: lineare Regression (d. h. Polynom vom Grad 1) und Minimierung der quadratischen Summe der Approximationsfehler
- Modell-Funktion: Gerade:
 $\hat{y}(u) = a_0 + a_1 u$
mit unbekannten Parametern a_0, a_1



Regressionsanalyse

- Gütefunktion:

$$Q = \sum_{k=0}^{n-1} (y_k - \hat{y}(u_k))^2 = \sum_{k=0}^{n-1} (y_k - a_0 - a_1 u_k)^2$$

- Notwendige Bedingung für optimale Parameter a_0, a_1 :

$$\frac{dQ}{da_0} = -2 \sum_{k=0}^{n-1} (y_k - a_0 - a_1 u_k) = 0$$

$$\Rightarrow a_1 \sum_{k=0}^{n-1} u_k + n a_0 = \sum_{k=0}^{n-1} y_k$$

$$\frac{dQ}{da_1} = -2 \sum_{k=0}^{n-1} u_k (y_k - a_0 - a_1 u_k) = 0$$

$$\Rightarrow a_1 \sum_{k=0}^{n-1} u_k^2 + a_0 \sum_{k=0}^{n-1} u_k = \sum_{k=0}^{n-1} u_k y_k$$

$$\Rightarrow a_0 = \frac{1}{n} \sum_{k=0}^{n-1} y_k - a_1 \frac{1}{n} \sum_{k=0}^{n-1} u_k,$$

$$a_1 = \frac{n \sum_{k=0}^{n-1} u_k y_k - \sum_{k=0}^{n-1} u_k \sum_{k=0}^{n-1} y_k}{n \sum_{k=0}^{n-1} u_k^2 - (\sum_{k=0}^{n-1} u_k)^2}$$

2.1 Approximation

Regressionsanalyse

- Gleiches Ergebnis (natürlich) bei Anwendung der oben beim LS-Schätzer dargestellten Vorgehensweise:

$$\hat{y} = a_0 + a_1 u = \sum_{i=0}^{m-1} a_i \varphi_i(u)$$

$$\Rightarrow \varphi_0(u) = 1, \quad \varphi_1(u) = u$$

$$\begin{aligned} \Rightarrow \hat{\mathbf{y}} &= \begin{bmatrix} \hat{y}_0 \\ \vdots \\ \hat{y}_{n-1} \end{bmatrix} = \begin{bmatrix} \varphi_0(u_0) & \cdots & \varphi_{m-1}(u_0) \\ \vdots & \ddots & \vdots \\ \varphi_0(u_{n-1}) & \cdots & \varphi_{m-1}(u_{n-1}) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & u_0 \\ \vdots & \vdots \\ 1 & u_{n-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \mathbf{\Phi} \mathbf{a} \end{aligned}$$

- Prüfung auf Übereinstimmung: Übung zu Hause...

Regressionsanalyse

- Diskussion:
 - Analoge Vorgehensweise bei Polynomen höherer Ordnung:
$$\hat{y} = a_0 + a_1 u + a_2 u^2 + \dots + a_{m-1} u^{m-1}$$
 - Unterschied zur Approximation mit orthonormalen Funktionen:
Alle Koeffizienten a_i müssen neu berechnet werden, wenn die Ordnung des Polynoms erhöht wird
 - Terme $\frac{1}{n} \sum_{k=0}^{n-1} u_k$, $n \sum_{k=0}^{n-1} u_k^2 - \left(\sum_{k=0}^{n-1} u_k \right)^2$:
Mittelwert und mit $\frac{n-1}{n}$ gewichtete Varianz der Größe u , siehe Kap. 4

Polynominterpolation

- Ansatz: Polynom

$$\hat{y}(u) = \sum_{i=0}^{n-1} a_i u^i = \mathbf{a}^T \mathbf{p} = \mathbf{p}^T \mathbf{a}$$

mit $\mathbf{a} = (a_0, \dots, a_{n-1})^T$, $\mathbf{p} = (1, u, u^2, \dots, u^{n-1})^T$

- $\hat{y}(u)$ soll die gegebenen Messpunkte (u_k, y_k) exakt wiedergeben:
Eindeutige Lösung durch Gleichungssystem aus n Gleichungen
für n Koeffizienten:

$$y_k(u_k) = \sum_{i=0}^{n-1} a_i u_k^i, \quad k \in \{0, \dots, n-1\}$$

- In Matrixschreibweise:

$$\mathbf{y} = \begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & u_0 & \cdots & u_0^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} & \cdots & u_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix} = \mathbf{V} \mathbf{a}$$

mit \mathbf{y} : Messvektor, \mathbf{V} : Vandermonde-Matrix

- Lösung durch Invertierung: $\mathbf{a} = \mathbf{V}^{-1} \mathbf{y}$
- Resultierende Interpolationsgleichung: $\hat{y}(u) = \mathbf{p}^T \mathbf{V}^{-1} \mathbf{y}$

2.2 Interpolation

Polynominterpolation

- Voraussetzung für Invertierbarkeit von V :
Determinante $|V| \neq 0 \Leftrightarrow$ Stützstellen u_k sind paarweise verschieden
- Rang: $\text{rg}(V) = r = n$

Polynominterpolation

- Geschlossene Bestimmung der Determinanten durch rekursive Zerlegung in Unterdeterminanten:

- Äquivalenzumformung für die Spalten $i \in \{2, \dots, n\}$:
 $(i - 1).$ Spalte $- (i - 2).$ Spalte $\cdot u_0$:

$$|V| = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 1 & u_1 - u_0 & \cdots & u_1^{n-1} - u_1^{n-2}u_0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} - u_0 & \cdots & u_{n-1}^{n-1} - u_{n-1}^{n-2}u_0 \end{vmatrix}$$

\uparrow \uparrow
 2. Spalte $-$ 1. Spalte $\cdot u_0$ $(n - 1).$ Spalte $- (n - 2).$ Spalte $\cdot u_0$

- Entwicklung nach der ersten Zeile, Herausziehung der Terme $(u_j - u_0)$ für $j \in \{1, \dots, n - 1\}$:

$$|V| = (u_1 - u_0) \cdots (u_{n-1} - u_0) \begin{vmatrix} 1 & u_1 & \cdots & u_1^{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} & \cdots & u_{n-1}^{n-2} \end{vmatrix}$$

2.2 Interpolation

Polynominterpolation

- Wieder Äquivalenzumformung für die Spalten i , $i \in \{2, \dots, n-1\}$:
 $(i-1)$. Spalte $- (i-2)$. Spalte $\cdot u_0$:

$$|V| = (u_1 - u_0) \cdots (u_{n-1} - u_0) \begin{vmatrix} 1 & 0 & \cdots & 0 \\ 1 & u_2 - u_1 & \cdots & u_2^{n-1} - u_2^{n-2}u_1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} - u_1 & \cdots & u_{n-1}^{n-1} - u_{n-1}^{n-2}u_1 \end{vmatrix}$$

- Entwicklung nach der ersten Zeile, Herausziehung der Terme $(u_j - u_0)$ für $j \in \{2, \dots, n-1\}$:

$$|V| = (u_1 - u_0) \cdots (u_{n-1} - u_0) \cdot (u_2 - u_1) \cdots (u_{n-1} - u_1) \begin{vmatrix} 1 & u_2 & \cdots & u_2^{n-3} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} & \cdots & u_{n-1}^{n-3} \end{vmatrix}$$

- Usw.

2.2 Interpolation

Polynominterpolation

- Ergebnis:

$$\begin{aligned} |V| = & (u_1 - u_0)(u_2 - u_0)(u_3 - u_0) \cdots (u_{n-1} - u_0) \cdot \\ & \cdot (u_2 - u_1)(u_3 - u_1) \cdots (u_{n-1} - u_1) \cdot \\ & \cdot (u_3 - u_2) \cdots (u_{n-1} - u_2) \cdot \\ & \cdot (u_{n-1} - u_{n-2}) \end{aligned}$$

- Produkt aller möglichen Differenzen zwischen den Stützstellen u_k
- Inversion von V ist daher möglich, wenn die Stützstellen nicht nahe zueinander liegen

Interpolation durch Lagrange-Polynome

- Nachteil der Polynominterpolation: relativ großer Aufwand für die Invertierung der Gleichung $\mathbf{y} = \mathbf{V}\mathbf{a}$
- Abhilfe: Verwendung von Lagrange-Polynomen
- Ansatz: direkte Gewichtung von Polynomen $l_i(u)$ mit Messwerten y_i :

$$\hat{y}(u) = \sum_{i=0}^{n-1} y_i l_i(u) = \mathbf{l}^T \mathbf{y}$$

$$\text{mit } \mathbf{y} = (y_0, \dots, y_{n-1})^T, \mathbf{l} = (l_0(u), \dots, l_{n-1}(u))^T$$

- Dadurch Vermeidung von weiteren Berechnungen der Koeffizienten
- Vergleich mit Polynominterpolation ($\hat{y}(u) = \mathbf{p}^T \mathbf{V}^{-1} \mathbf{y}$):

$$\mathbf{l}^T = \mathbf{p}^T \mathbf{V}^{-1} \Rightarrow \mathbf{l} = (\mathbf{V}^{-1})^T \mathbf{p}$$

- Daraus folgt (ohne Beweis):

$$\begin{aligned} l_i(u) &= \prod_{j=0, j \neq i}^{n-1} \frac{u - u_j}{u_i - u_j} \\ &= \frac{(u - u_0) \cdots (u - u_{i-1}) \cdot (u - u_{i+1}) \cdots (u - u_{n-1})}{(u_i - u_0) \cdots (u_i - u_{i-1}) \cdot (u_i - u_{i+1}) \cdots (u_i - u_{n-1})} \end{aligned}$$

2.2 Interpolation

Interpolation durch Lagrange-Polynome

- $$l_i(u) = \prod_{j=0, j \neq i}^{n-1} \frac{u - u_j}{u_i - u_j}$$
$$= \frac{(u - u_0) \cdots (u - u_{i-1}) \cdot (u - u_{i+1}) \cdots (u - u_{n-1})}{(u_i - u_0) \cdots (u_i - u_{i-1}) \cdot (u_i - u_{i+1}) \cdots (u_i - u_{n-1})}$$

- Daraus folgt: $l_i(u_j) = \delta_i^j$
- Damit erhält man an den Stützstellen u_j :

$$\hat{y}(u_j) = \sum_{i=0}^{n-1} y_i l_i(u_j) = \sum_{i=0}^{n-1} y_i \delta_i^j = y_j$$

d. h. die Stützstellen werden (wie gefordert) exakt interpoliert

2.2 Interpolation

Interpolation durch Lagrange-Polynome

- Beispiel: 3 Messpunkte (Stützstellen)

$(u_a, y_a), (u_m, y_m), (u_e, y_e)$

mit $u_m - u_a = u_e - u_m = h$

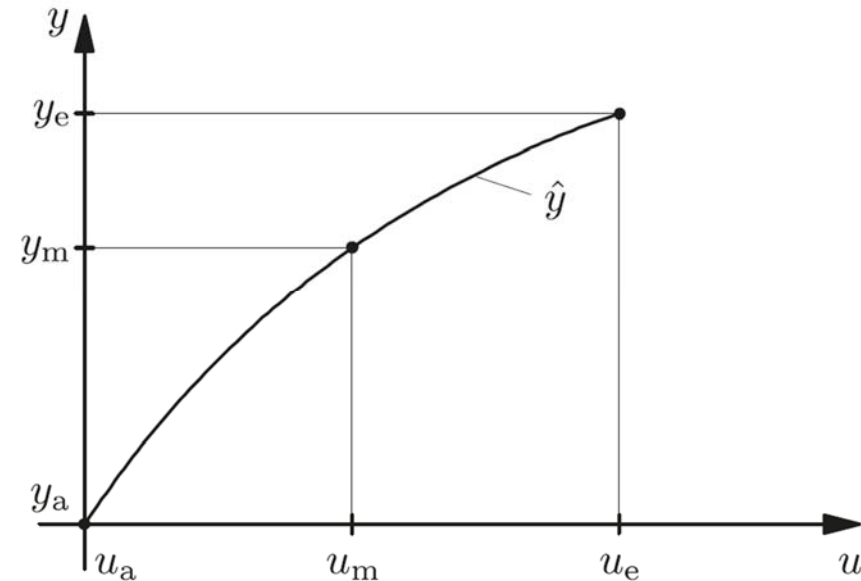
(äquidistante Abtastung)

- Lagrange-Polynome:

$$\begin{aligned} l_0(u) &= \frac{(u - u_m)(u - u_e)}{(u_a - u_m)(u_a - u_e)} \\ &= \frac{1}{2h^2} (u - u_m)(u - u_e) \end{aligned}$$

$$\begin{aligned} l_1(u) &= \frac{(u - u_a)(u - u_e)}{(u_m - u_a)(u_m - u_e)} \\ &= -\frac{1}{h^2} (u - u_a)(u - u_e) \end{aligned}$$

$$\begin{aligned} l_2(u) &= \frac{(u - u_a)(u - u_m)}{(u_e - u_a)(u_e - u_m)} \\ &= \frac{1}{2h^2} (u - u_a)(u - u_m) \end{aligned}$$



Bildquelle: F. Puente León: Messtechnik, 10. Auflage, Springer, 2015

© Michael Heizmann, IIT, KIT, alle Rechte einschließlich Kopier- und Weitergaberechte bei uns.

2.2 Interpolation

Interpolation durch Lagrange-Polynome

- Beispiel: 3 Messpunkte (Stützstellen)

$(u_a, y_a), (u_m, y_m), (u_e, y_e)$

mit $u_m - u_a = u_e - u_m = h$

(äquidistante Abtastung)

- Einsetzen in Interpolationsgleichung:

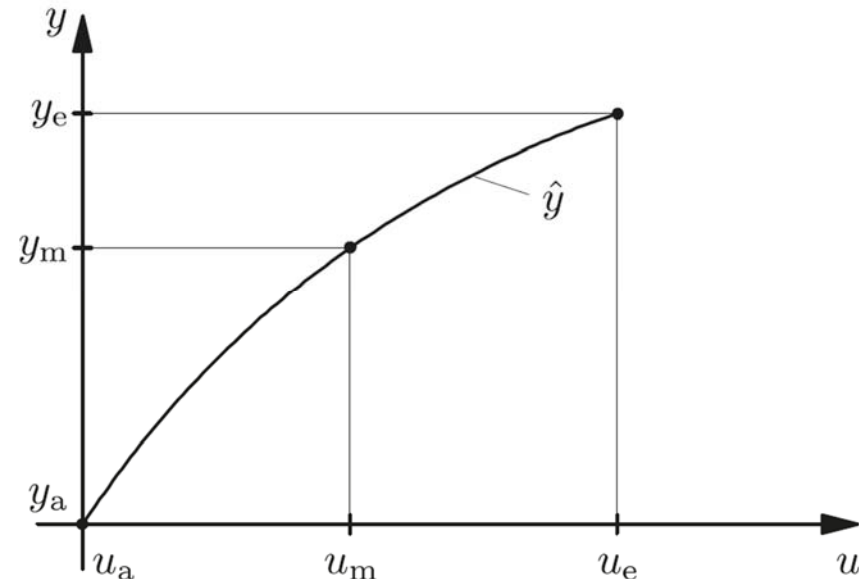
$$\hat{y}(u) = \sum_{i=0}^2 y_i l_i(u)$$

$$= \frac{y_a}{2h^2} (u - u_m)(u - u_e) - \frac{y_m}{h^2} (u - u_a)(u - u_e) + \frac{y_e}{2h^2} (u - u_a)(u - u_m)$$

- Mit $u_a = 0, y_a = 0, u_m = h, u_e = 2h$:

$$\hat{y}(u) = u \left(\frac{2y_m}{h} - \frac{y_e}{2h} \right) + u^2 \left(\frac{y_e}{2h^2} - \frac{y_m}{h^2} \right)$$

- Falls Stützstellen auf einer Geraden ($y_m = \frac{y_e}{2}$): $\hat{y}(u) = u \cdot \frac{y_e}{2h}$



2.2 Interpolation

Interpolation durch Newton-Polynome

- Ansatz: Verwendung von Newton-Polynomen:

$$\hat{y}(u) = a_0 + a_1(u - u_0) + a_2(u - u_0)(u - u_1) + \dots \\ + a_{n-1}(u - u_0)(u - u_1) \dots (u - u_{n-2})$$

- Rekursive Bestimmung der Koeffizienten aus den Interpolationsbedingungen:

$$u_0: \quad y_0 = a_0$$

$$u_1: \quad y_1 = a_0 + a_1(u_1 - u_0)$$

$$\vdots$$

$$u_{n-1}: \quad y_{n-1} = a_0 + a_1(u_{n-1} - u_0) + a_2(u_{n-1} - u_0)(u_{n-1} - u_1) + \dots \\ + a_{n-1}(u_{n-1} - u_0)(u_{n-1} - u_1) \dots (u_{n-1} - u_{n-2})$$

- Vorteil: leichte Erweiterbarkeit beim Hinzufügen von Stützstellen, einfachere Basispolynome (als Lagrange-Polynome)

Interpolation durch Newton-Polynome

- Geschlossene Bestimmung der Koeffizienten für konstanten Stützstellenabstand h :
- Dazu Einführung des Differenzenoperators: $\Delta y_i = y_{i+1} - y_i$:
 - Es gelten das Distributivgesetz und das Kommutativgesetz bzgl. einer Konstanten
 - Höhere Differenzen: $\Delta^j y_i = \Delta^{j-1}(\Delta y_i) = \Delta^{j-1}(y_{i+1} - y_i)$,
z. B. $\Delta^2 y_i = \Delta(\Delta y_i) = \Delta(y_{i+1} - y_i) = \Delta y_{i+1} - \Delta y_i$
$$= (y_{i+2} - y_{i+1}) - (y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i$$
- Für den Differenzenoperator gilt: $y_i = (1 + \Delta)^i y_0$
 - Beweis mittels vollständiger Induktion:
 - Anfang $i = 1$: $y_1 = (1 + \Delta)y_0 = y_0 + (y_1 - y_0) = y_1$
 - Schluss von $i \rightarrow i + 1$: $y_{i+1} = (1 + \Delta)^{i+1} y_0 = (1 + \Delta) \underbrace{(1 + \Delta)^i y_0}_{= y_i}$
$$= (1 + \Delta)y_i$$
$$= y_i + (y_{i+1} - y_i) = y_{i+1}$$

2.2 Interpolation

Interpolation durch Newton-Polynome

- Geschlossene Bestimmung der Koeffizienten für konstanten Stützstellenabstand h :

- Ausmultiplizieren von $y_i = (1 + \Delta)^i y_0$:

$$y_i = y_0 + \frac{i}{1!} \Delta y_0 + \frac{i(i-1)}{2!} \Delta^2 y_0 + \dots$$
$$+ \frac{i(i-1) \cdots (i-j+1)}{j!} \Delta^j y_0 + \dots + \frac{i!}{i!} \Delta^i y_0$$

- Rekursive Bestimmung der Koeffizienten aus den Interpolationsbedingungen bei konstantem Stützstellenabstand h :

$$u_0: \quad y_0 = a_0$$

$$u_1: \quad y_1 = a_0 + a_1 h$$

$$\vdots$$

$$u_i: \quad y_i = a_0 + a_1 i h + \dots + a_j i(i-1) \cdots (i-j+1) h^j + \dots + a_i i! h^i$$

$$\vdots$$

$$u_{n-1}: \quad y_{n-1} = a_0 + a_1 (n-1) h + a_2 (n-1)(n-2) h^2 + \dots$$
$$+ a_{n-1} (n-1)! h^{n-1}$$

Vergleich

2.2 Interpolation

Interpolation durch Newton-Polynome

- Geschlossene Bestimmung der Koeffizienten für konstanten Stützstellenabstand h :

- Vergleich:

$$\frac{i(i-1)\cdots(i-j+1)}{j!} \Delta^j y_0 = a_j i(i-1)\cdots(i-j+1) h^j$$
$$\Rightarrow a_j = \frac{\Delta^j y_0}{j! h^j}$$

- Damit also Interpolationsgleichung für konstanten Stützstellenabstand h :

$$\hat{y}(u) = y_0 + \frac{\Delta y_0}{h} (u - u_0) + \frac{\Delta^2 y_0}{2h^2} (u - u_0)(u - u_1) + \cdots$$
$$+ \frac{\Delta^{n-1} y_0}{(n-1)! h^{n-1}} (u - u_0)(u - u_1) \cdots (u - u_{n-2})$$

2.2 Interpolation

Interpolation durch Newton-Polynome

- Differenzenschema für $\Delta^j y_0$ für konstanten Stützstellenabstand h :

u	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0	y_0				
		Δy_0 $= y_1 - y_0$			
h	y_1		$\Delta^2 y_0$ $= \Delta y_1 - \Delta y_0$		
		Δy_1 $= y_2 - y_1$		$\Delta^3 y_0$ $= \Delta^2 y_1 - \Delta^2 y_0$	
$2h$	y_2		$\Delta^2 y_1$ $= \Delta y_2 - \Delta y_1$		$\Delta^4 y_0$ $= \Delta^3 y_1 - \Delta^3 y_0$
		Δy_2 $= y_3 - y_2$		$\Delta^3 y_1$ $= \Delta^2 y_2 - \Delta^2 y_1$	
$3h$	y_3		$\Delta^2 y_2$ $= \Delta y_3 - \Delta y_2$		
		Δy_3 $= y_4 - y_3$			
$4h$	y_4				

2.2 Interpolation

Interpolation durch Newton-Polynome

- Interpolationsgleichung für nicht konstanten Stützstellenabstand:

$$\hat{y}(u) = a_0 + a_1(u - u_0) + a_2(u - u_0)(u - u_1) + \dots \\ + a_{n-1}(u - u_0)(u - u_1) \dots (u - u_{n-2})$$

mit $a_j = \Delta^j y_0$,

$$\Delta^{j+1} y_k = \frac{\Delta^j y_{k+1} - \Delta^j y_k}{u_{k+j+1} - u_k}, \quad \Delta^0 y_k = y_k$$

2.2 Interpolation

Interpolation durch Newton-Polynome

- Verallgemeinertes Differenzenschema für $\Delta^j y_0$ für nicht konstanten Stützstellenabstand

u	y	Δy	$\Delta^2 y$	$\Delta^3 y$
u_0	$\Delta^0 y_0 = y_0$			
		$\Delta y_0 = \frac{y_1 - y_0}{u_1 - u_0}$		
u_1	$\Delta^0 y_1 = y_1$		$\Delta^2 y_0 = \frac{\Delta y_1 - \Delta y_0}{u_2 - u_0}$	
		$\Delta y_1 = \frac{y_2 - y_1}{u_2 - u_1}$		$\Delta^3 y_0 = \frac{\Delta^2 y_1 - \Delta^2 y_0}{u_3 - u_0}$
u_2	$\Delta^0 y_2 = y_2$		$\Delta^2 y_1 = \frac{\Delta y_2 - \Delta y_1}{u_3 - u_1}$	
		$\Delta y_2 = \frac{y_3 - y_2}{u_3 - u_2}$		
u_3	$\Delta^0 y_3 = y_3$			

Interpolation durch Newton-Polynome

- Vorgehen beim Hinzufügen einer Stützstelle (u_n, y_n) :
 - 1. Möglichkeit: Erweiterung des Differenzenschemas
 - 2. Möglichkeit: Rekursiver Ansatz:
 - Ausgangspunkt: aus $n - 1$ Stützstellen berechnetes Polynom $\hat{y}_{n-1}(u)$ mit Grad $n - 1$
 - Zufügung eines Polynoms für (u_n, y_n) :
$$\hat{y}_n(u) = \hat{y}_{n-1}(u) + a_n(u - u_0)(u - u_1) \cdots (u - u_{n-1})$$
 - Neuer Punkt (u_n, y_n) muss interpoliert werden:
$$y_n = \hat{y}_{n-1}(u_n) + a_n(u_n - u_0)(u_n - u_1) \cdots (u_n - u_{n-1})$$
$$\Rightarrow a_n = \frac{y_n - \hat{y}_{n-1}(u_n)}{(u_n - u_0)(u_n - u_1) \cdots (u_n - u_{n-1})}$$

2.2 Interpolation

Interpolation durch Newton-Polynome

- Beispiel: 3 Messpunkte (Stützstellen)

$(u_a, y_a), (u_m, y_m), (u_e, y_e)$

mit $u_m - u_a = u_e - u_m = h$

(äquidistante Abtastung),

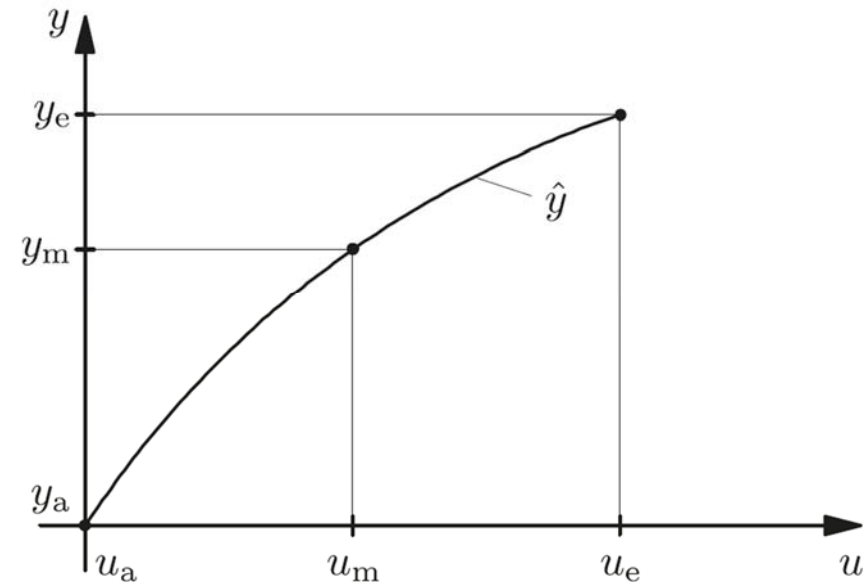
$u_a = 0, y_a = 0, u_m = h, u_e = 2h$

- Differenzenschema:

u	y	Δy	$\Delta^2 y$
0	y_0		
		Δy_0	
h	y_1		$\Delta^2 y_0$
		Δy_1	
$2h$	y_2		

- Interpolationsfunktion:

$$\hat{y}(u) = y_0 + \frac{\Delta y_0}{h} (u - u_0) + \frac{\Delta^2 y_0}{2h^2} (u - u_0)(u - u_1)$$



Bildquelle: F. Puente León: Messtechnik, 10. Auflage, Springer, 2015

© Michael Heizmann, IIT, KIT, alle Rechte einschließlich Kopier- und Weitergaberechte bei uns.

2.2 Interpolation

Interpolation durch Newton-Polynome

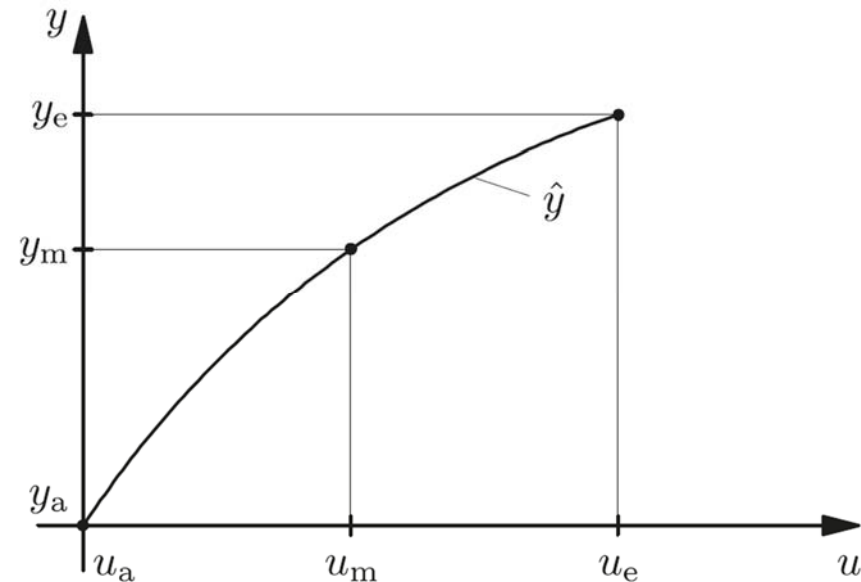
- Beispiel: 3 Messpunkte (Stützstellen)

$$(u_a, y_a), (u_m, y_m), (u_e, y_e)$$

$$\text{mit } u_m - u_a = u_e - u_m = h$$

(äquidistante Abtastung),

$$u_a = 0, y_a = 0, u_m = h, u_e = 2h$$



- Interpolationsfunktion:

$$\begin{aligned}\hat{y}(u) &= y_0 + \frac{\Delta y_0}{h} (u - u_0) + \frac{\Delta^2 y_0}{2h^2} (u - u_0)(u - u_1) \\ &= y_a + \frac{(y_m - y_a)}{h} (u - u_a) + \frac{(y_e - 2y_m + y_a)}{2h^2} (u - u_a)(u - u_m) \\ &= \frac{y_m}{h} u + \frac{(y_e - 2y_m)}{2h^2} u(u - h)\end{aligned}$$

Identisch zum Ergebnis bei Interpolation durch Lagrange-Polynome

- Falls Stützstellen auf einer Geraden ($y_m = \frac{y_e}{2}$): $\hat{y}(u) = u \cdot \frac{y_e}{2h}$,
siehe Lagrange-Polynome

2.2 Interpolation

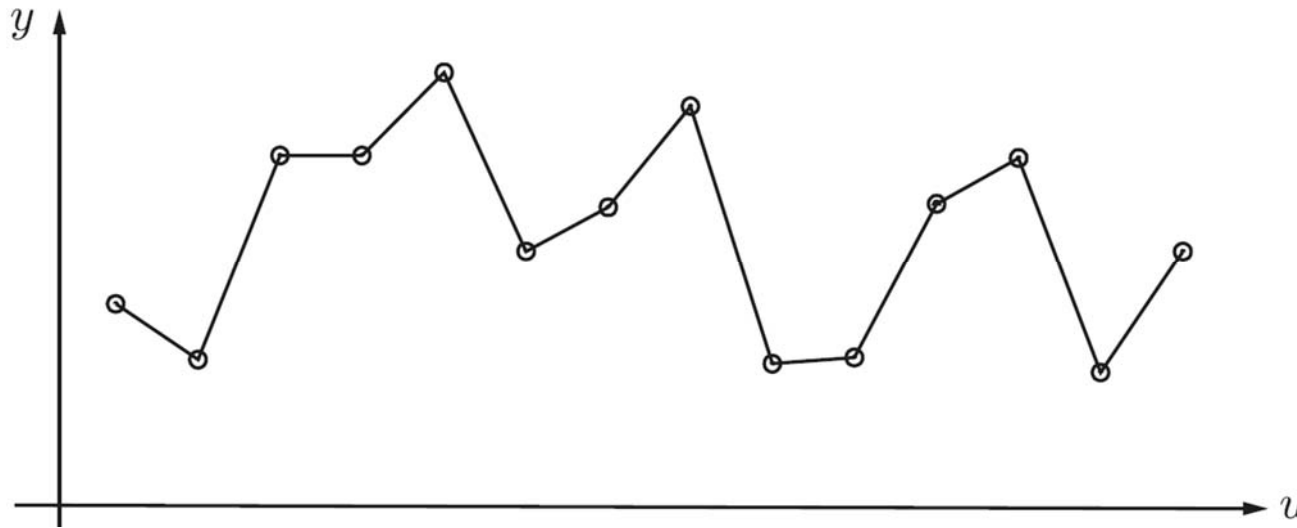
Spline-Interpolation

- Nachteil der Interpolation durch Polynome:
 - Bei großer Anzahl von Messpunkten hohe Grade von Interpolationspolynomen erforderlich
 - Dadurch stark oszillierendes Verhalten der Interpolationsfunktion möglich
- Alternative: Zuweisung von Polynomen niedrigen Grades (z. B. vom Grad 3: kubische Polynome) zu Teilintervallen

2.2 Interpolation

Spline-Interpolation

- Einfachstes Verfahren: Polynome vom Grad 1: Geradenstücke
 - Interpolationsfunktion: $n - 1$ Geradenstücke in den Intervallen $[u_i, u_{i+1}]$ für $i \in \{0, \dots, n - 2\}$



- Nachteil: bereits 1. Ableitung (entspricht Empfindlichkeit, s. Kap. 3) ist an den Stützstellen nicht mehr stetig
- Für Kennlinieninterpolation daher nicht geeignet (s. Kap 3)

Spline-Interpolation

- Abhilfe: Interpolation mit Splines
- Splines: mechanisches Modell von dünnen Latten (engl. *splines*)
- Idee: durch gegebene Stützstellen wird eine dünne, homogene Latte gelegt, keine Momente in den Stützstellen
- Entstehende Biegelinie wird als Lösung der Interpolationsaufgabe verwendet
- Ansatz zur Bestimmung der Biegelinie: Berechnung der in der Latte gespeicherten Energie in Abhängigkeit von deren Krümmung, Minimum der Energie führt zur Biegelinie
- Näherung für Krümmung: zweite Ableitung $s''(u)$
- Technische Mechanik:

$$E = \frac{1}{2} \int_{u_0}^{n-1} (s''(u))^2 du$$

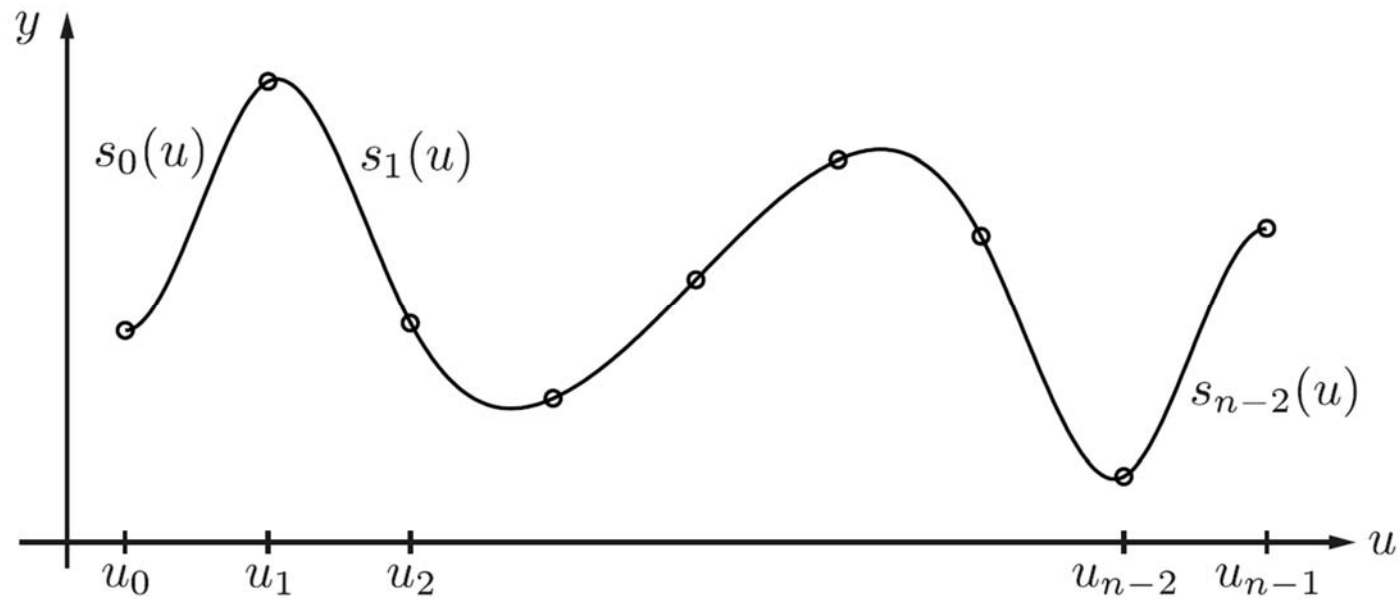
Spline-Interpolation

- Biegelinie: $\int_{u_0}^{n-1} (s''(u))^2 du \rightarrow \min$
- Wunsch nach stetiger Ableitung: $s(u)$ muss mindestens einmal stetig differenzierbar sein
- Variationsrechnung liefert:
 - $s_i(u_i) = y_i$ für $i \in \{0, 1, \dots, n-1\}$
(Interpolationsbedingung)
 - $s_i''(u_i + 0) = s_{i-1}''(u_i - 0)$ für $i \in \{0, 1, \dots, n-2\}$
(d. h. auch zweite Ableitung ist stetig)
 - $s_0''(u_0) = s_{n-2}''(u_{n-1})$
 - $s''''(u) = 0$ für $u \notin u_0, \dots, u_{n-1}$
(d. h. $s(u)$ ist in jedem Teilintervall ein kubisches Polynom:
kubische Splines)
- Erweiterung zu Splines höherer Ordnung möglich, aber ohne physikalische Motivation

2.2 Interpolation

Spline-Interpolation

- Kubische Splines:



Spline-Interpolation

- Berechnung:

- Ansatz: allgemeines Polynom 3. Grades:

$$s_i(u) = a_i(u - u_i)^3 + b_i(u - u_i)^2 + c_i(u - u_i) + d_i$$

für jedes Teilintervall $[u_i, u_{i+1}]$ der Länge $h_i = u_{i+1} - u_i$

- Anwendung der Bedingungen (4 Gleichungen für 4 Unbekannte):

$$s_i(u_i) = d_i = y_i$$

$$s_i(u_{i+1}) = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i = y_{i+1}$$

$$s_i'(u_i) = c_i$$

$$s_i'(u_{i+1}) = 3a_i h_i^2 + 2b_i h_i + c_i$$

$$s_i''(u_i) = 2b_i = y_i''$$

$$s_i''(u_{i+1}) = 6a_i h_i + 2b_i = y_{i+1}''$$

- Daraus Bestimmung der Koeffizienten:

$$a_i = \frac{1}{6h_i}(y_{i+1}'' - y_i''), \quad b_i = \frac{1}{2}y_i''$$

$$c_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{6}h_i(y_{i+1}'' + 2y_i''), \quad d_i = y_i$$

Spline-Interpolation

- Berechnung:

- Bestimmung der 2. Ableitungen y_i'' , y_{i+1}'' :

- Dazu Nutzung der Stetigkeit der 1. Ableitung an den Stützstellen:

$$s'_i(u_{i+1}) = s'_{i+1}(u_{i+1})$$

- $$\begin{aligned} s'_i(u_{i+1}) &= 3a_i h_i^2 + 2b_i h_i + c_i \\ &= 3 \frac{1}{6h_i} (y_{i+1}'' - y_i'') h_i^2 + 2 \frac{1}{2} y_i'' h_i + \frac{1}{h_i} (y_{i+1} - y_i) - \frac{1}{6} (y_{i+1}'' + 2y_i'') \\ &= \frac{1}{h_i} (y_{i+1} - y_i) + \frac{h_i}{2} (y_{i+1}'' + y_i'') - \frac{1}{6} (y_{i+1}'' + 2y_i'') \\ &= \frac{1}{h_i} (y_{i+1} - y_i) + \frac{h_i}{6} (2y_{i+1}'' + y_i'') \end{aligned}$$
- $$s'_{i+1}(u_{i+1}) = c_{i+1} = \frac{1}{h_{i+1}} (y_{i+2} - y_{i+1}) - \frac{1}{6} h_{i+1} (y_{i+2}'' + 2y_{i+1}'')$$

Spline-Interpolation

- Gleichsetzen:

$$h_i y_i'' + 2(h_i + h_{i+1}) y_{i+1}'' + h_{i+1} y_{i+2}'' = \frac{6}{h_{i+1}} (y_{i+2} - y_{i+1}) - \frac{6}{h_i} (y_{i+1} - y_i)$$

- Mit $y_0'' = y_{n-1}'' = 0$ folgen also $n - 2$ lineare Gleichungen für die unbekannten 2. Ableitungen $y_1'', y_2'', \dots, y_{n-2}''$:

$$\begin{bmatrix} 2(h_0 + h_1) & h_1 & 0 & \dots & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & \dots & 0 & 0 & h_{n-3} & 2(h_{n-3} + h_{n-2}) \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ \vdots \\ y_{n-2}'' \end{bmatrix} = \begin{bmatrix} \frac{6}{h_1} (y_2 - y_1) - \frac{6}{h_0} (y_1 - y_0) \\ \frac{6}{h_2} (y_3 - y_2) - \frac{6}{h_1} (y_2 - y_1) \\ \vdots \\ \frac{6}{h_{n-2}} (y_{n-1} - y_{n-2}) - \frac{6}{h_{n-3}} (y_{n-2} - y_{n-3}) \end{bmatrix}$$

2.2 Interpolation

- Beispiel: Interpolation einer Sinusfunktion $y(u) = \sin\left(\frac{\pi}{2}u\right)$ im Intervall $[-2; 2]$ durch Lagrange-Funktionen und Spline-Funktionen

- Stützstellen:

$$\mathbf{u} = [-2; -1; 0; 1; 2]; \mathbf{y} = [0; -1; 0; 1; 0]$$

- Ergebnis der Lagrange-Interpolation:

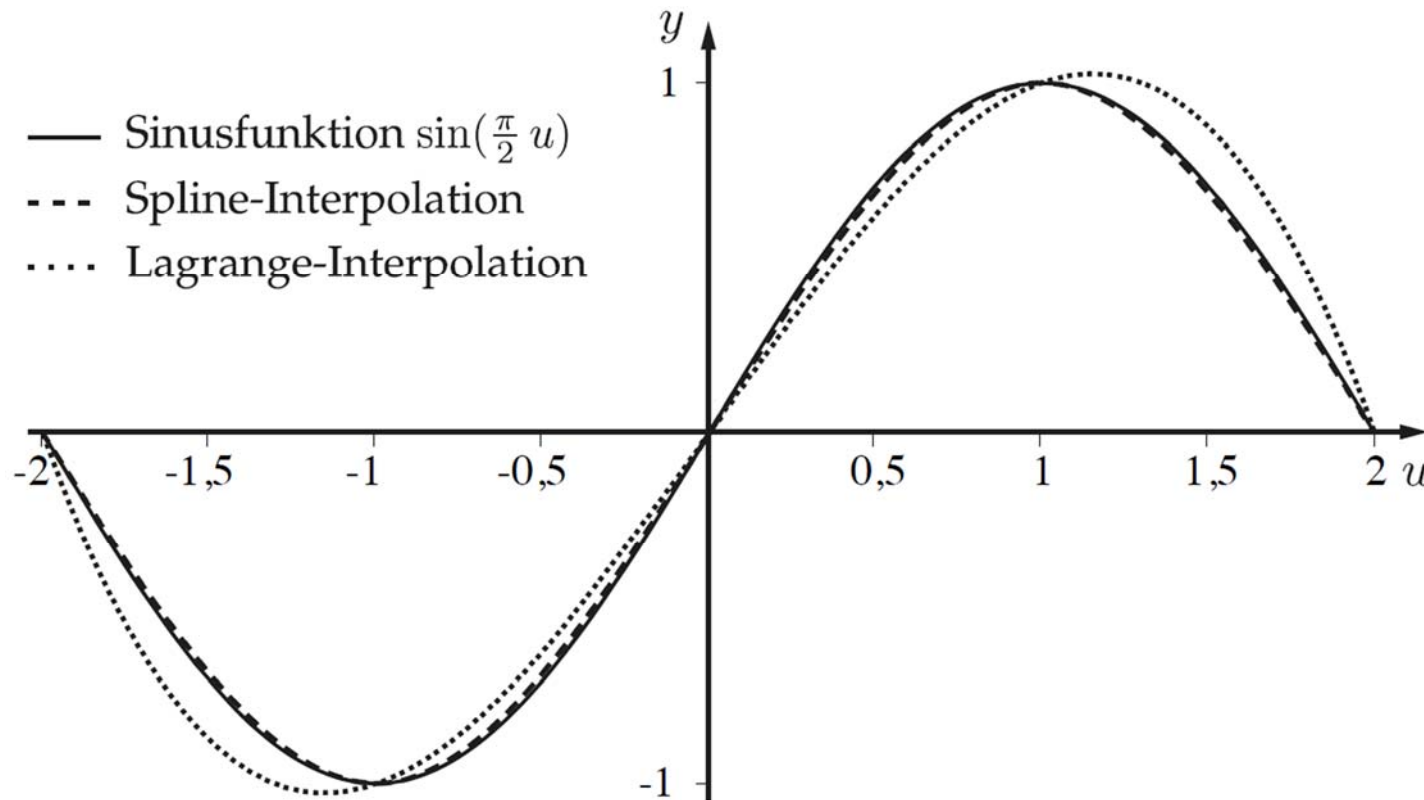
$$\hat{y}_L(u) = -\frac{1}{3}u^3 + \frac{4}{3}u, \quad -2 \leq u \leq 2$$

- Spline-Interpolation: $\mathbf{y}'' = [0; 3; 0; -3; 0]$,
Interpolationsfunktion:

$$\hat{y}_S(u) = s(u) = \begin{cases} \frac{1}{2}u^3 + 3u^2 + \frac{9}{2}u + 1 & \text{für } u < -1 \\ -\frac{1}{2}u^3 + \frac{3}{2}u & \text{für } -1 \leq u < 1 \\ \frac{1}{2}u^3 - 3u^2 + \frac{9}{2}u - 1 & \text{für } u \geq 1 \end{cases}$$

2.2 Interpolation

- Beispiel: Interpolation einer Sinusfunktion $y(u) = \sin\left(\frac{\pi}{2}u\right)$ im Intervall $[-2; 2]$ durch Lagrange-Funktionen und Spline-Funktionen



- Spline-Interpolation liefert in diesem Fall die bessere Näherung

Systemtheoretische Deutung der Interpolation

- Für äquidistante Stützstellen: Beobachtung lässt sich durch Multiplikation einer kontinuierlichen Funktion $y(u)$ mit einer Impulsreihe (mit Δu : Stützstellenabstand) ausdrücken:

$$y_*(u) = y(u) \cdot \sum_{n=-\infty}^{\infty} \delta(u - n \Delta u)$$

- Dadurch verschwindet die abgetastete Funktion $y_*(u)$ für alle $u \neq n \Delta u, n \in \mathbb{Z}$
- Bestimmte Interpolationsverfahren können als Faltung dieses abgetasteten Signals $y_*(u)$ mit einem linearen, verschiebungsinvarianten Interpolationsfilter mit der Impulsantwort $i(u)$ dargestellt werden:

$$\hat{y}(u) = \left[y(u) \cdot \sum_{n=-\infty}^{\infty} \delta(u - n \Delta u) \right] * i(u)$$

- Fouriertransformierte:

$$\hat{Y}(f) = \left[\frac{1}{\Delta u} \cdot \sum_{k=-\infty}^{\infty} Y \left(f - \frac{k}{\Delta u} \right) \right] \cdot I(f)$$

Systemtheoretische Deutung der Interpolation

- D. h. Faltung des Abtastsignals $y_*(u)$ mit der Impulsantwort $i(u)$ entspricht im Frequenzbereich der Fourier-Transformation einer Multiplikation des periodisch fortgesetzten Spektrums $Y(f)$ mit der Übertragungsfunktion $I(f)$ des Interpolationsfilters
- Für Interpolationsfunktionen $i(u)$ mit geeigneter Charakteristik (z. B. Tiefpassverhalten): Rekonstruktion des ursprünglichen Signals $y(u)$ möglich (siehe Kap. 7)

2.2 Interpolation

Systemtheoretische Deutung der Interpolation

- Beispiele:

Interpolationsart	Ordnung	$i(u)$	$I(f)$
Nächster-Nachbar-Interpolation	0	$\text{rect}\left(\frac{u}{\Delta u}\right)$	$\Delta u \text{sinc}(f \Delta u)$
Lineare Interpolation	1	$\Lambda\left(\frac{u}{\Delta u}\right)$	$\Delta u \text{sinc}^2(f \Delta u)$
Ideale Interpolation	∞	$\text{sinc}\left(\frac{u}{\Delta u}\right)$	$\Delta u \text{rect}(f \Delta u)$

$$\text{rect}\left(\frac{x}{B}\right) = \begin{cases} 1 & \text{für } |x| < \frac{B}{2} \\ 0 & \text{sonst} \end{cases}$$

$$\Lambda\left(\frac{x}{B}\right) = \text{rect}\left(\frac{x}{B}\right) * \text{rect}\left(\frac{x}{B}\right) = \begin{cases} 1 - \left|\frac{x}{B}\right| & \text{für } |x| < B \\ 0 & \text{sonst} \end{cases}$$

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

2.3 Kennfeldinterpolation

- Bisher: Interpolation bzw. Approximation einer eindimensionalen Funktion
- Jetzt: zweidimensionale Interpolation
- Anwendung: z. B. Abhängigkeit der Ausgangsgröße y von der Messgröße u und einer systematischen Störgröße z , dazu Messung von Kennfeldwerten y_i an den (zweidimensionalen) Stützstellen (u_i, z_i)
- Zweidimensionale Polynominterpolation:

$$\hat{y}(u, z) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_{ij} u^i z^j$$

2.3 Kennfeldinterpolation

- Bei Polynomen 1. Grades (lineare Interpolation): Abbruch der Taylor-Reihe nach dem linearen Glied:

$$y(u, z) = f(u_i + \Delta u, z_j + \Delta z) \\ \approx f(u_i, z_j) + \frac{\partial f}{\partial u}(u_i, z_j) \Delta u + \frac{\partial f}{\partial z}(u_i, z_j) \Delta z + \frac{\partial^2 f}{\partial u \partial z}(u_i, z_j) \Delta u \Delta z$$

(mit abkürzender Schreibweise: $\frac{\partial f}{\partial u}(u_i, z_j) := \left. \frac{\partial f(u, z)}{\partial u} \right|_{u_i, z_j}$)

- Approximation der Ableitungen durch Differenzenquotienten:

$$y(u, z) \approx y(u_i, z_j) + \frac{\Delta y(u_i)}{\Delta u_i} \Delta u + \frac{\Delta y(z_j)}{\Delta z_j} \Delta z + \frac{\Delta^2 y(u_i, z_j)}{\Delta u_i \Delta z_j} \Delta u \Delta z$$

Bilineare Interpolation

2.3 Kennfeldinterpolation

- Abkürzende Schreibweise: $y_{kl} = y(u_{i+k}, z_{j+l})$:

$$\Delta y(u_i) = y(u_{i+1}, z_j) - y(u_i, z_j) = y_{10} - y_{00},$$

$$\Delta y(z_j) = y(u_i, z_{j+1}) - y(u_i, z_j) = y_{01} - y_{00},$$

$$\begin{aligned}\Delta y^2(u_i, z_j) &= y(u_{i+1}, z_{j+1}) - y(u_{i+1}, z_j) - y(u_i, z_{j+1}) + y(u_i, z_j) \\ &= y_{11} - y_{10} - y_{01} + y_{00}\end{aligned}$$

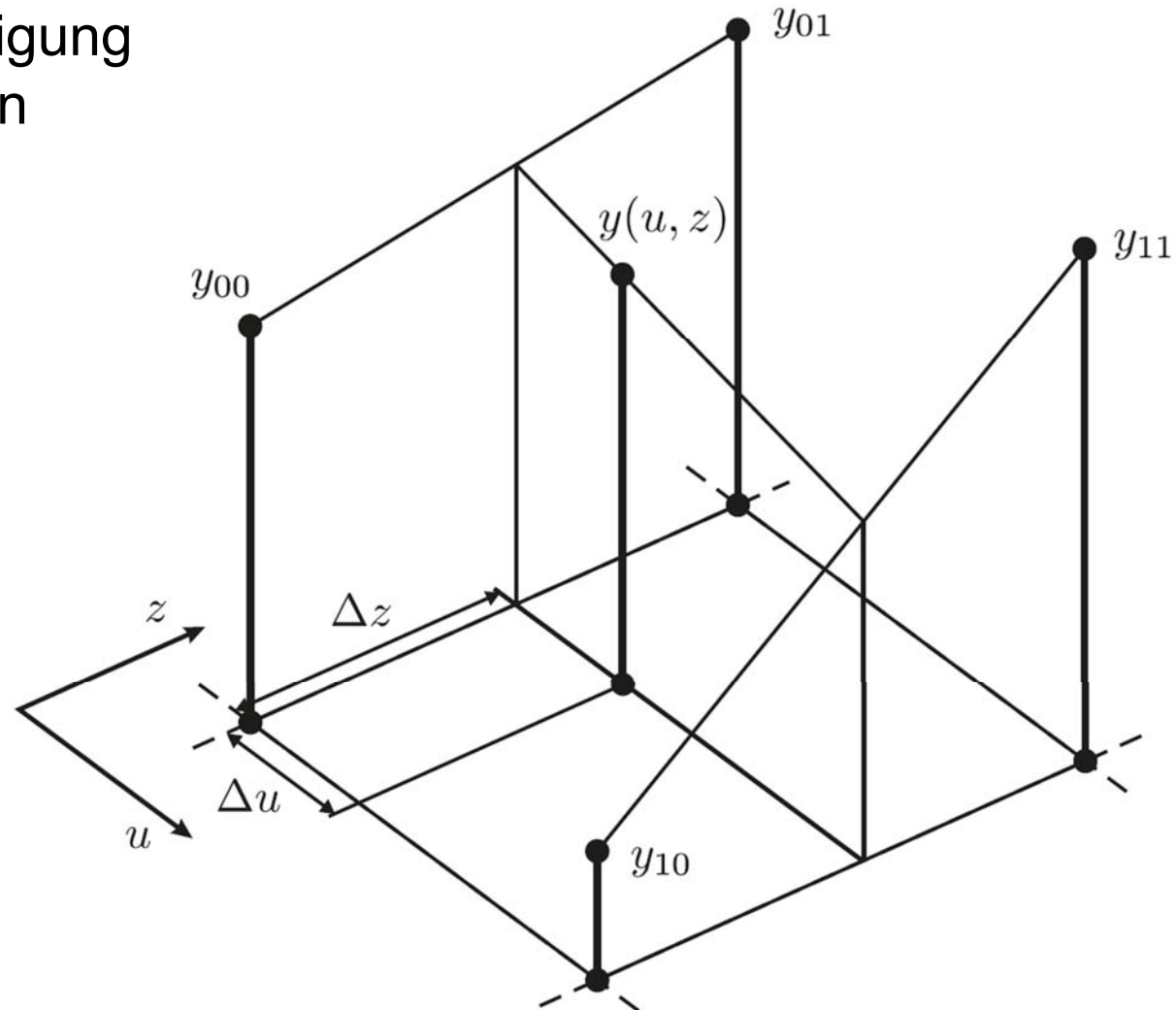
- Bei normierter Stützstellenweite $\Delta u_i = \Delta z_j = 1$:

$$\begin{aligned}y(u, z) &\approx y(u_i, z_j) + \frac{\Delta y(u_i)}{\Delta u_i} \Delta u + \frac{\Delta y(z_j)}{\Delta z_j} \Delta z + \frac{\Delta^2 y(u_i, z_j)}{\Delta u_i \Delta z_j} \Delta u \Delta z \\ &= y_{00} + (y_{10} - y_{00}) \Delta u + (y_{01} - y_{00}) \Delta z + (y_{11} - y_{10} - y_{01} + y_{00}) \Delta u \Delta z \\ &= y_{00}(1 - \Delta u)(1 - \Delta z) + y_{10} \Delta u(1 - \Delta z) + y_{01}(1 - \Delta u) \Delta z + y_{11} \Delta u \Delta z\end{aligned}$$

d. h. Gewichtung der Stützstellen in Abhängigkeit von ihrem Abstand zur Interpolationsstelle

2.3 Kennfeldinterpolation

- D. h. Gewichtung der Stützstellen in Abhängigkeit von ihrem Abstand zur Interpolationsstelle
- Lineare Interpolation:
Unstetigkeiten der Steigung an den Intervallgrenzen



Rechnergestützte Kennfeldberechnung

- Gesucht: günstige Darstellung der Kennfeldinterpolation für die Implementierung auf einem Rechner
- Dazu meist Beschränkung der Berechnung auf ein festes Raster der Breite q_u bzw. q_z zwischen den Stützstellen
- Annahme: Intervallbreite von u und z besitzen die gleiche Auflösung 2^r bei vorgegebener Quantisierung q_u und q_z
- Damit Rasterung des Intervalls in Zweierpotenzen:
 $\Delta u_i = u_{i+1} - u_i = 2^r q_u, \quad \Delta z_j = z_{j+1} - z_j = 2^r q_z$
- $$y(u, z) \approx y(u_i, z_j) + \frac{\Delta y(u_i)}{\Delta u_i} \Delta u + \frac{\Delta y(z_j)}{\Delta z_j} \Delta z + \frac{\Delta^2 y(u_i, z_j)}{\Delta u_i \Delta z_j} \Delta u \Delta z$$
$$= y(u_i, z_j) + \frac{y(u_{i+1}, z_j) - y(u_i, z_j)}{2^r} \frac{\Delta u}{q_u} + \frac{y(u_i, z_{j+1}) - y(u_i, z_j)}{2^r} \frac{\Delta z}{q_u}$$
$$+ \frac{y(u_{i+1}, z_{j+1}) - y(u_i, z_{j+1}) - y(u_{i+1}, z_j) + y(u_i, z_j)}{2^{2r}} \frac{\Delta u \Delta z}{q_u q_z}$$

2.3 Kennfeldinterpolation

Rechnergestützte Kennfeldberechnung

- Ordnen der Summanden nach den Stützstellen:

$$\begin{aligned} y(u, z) &\approx y(u_i, z_j) \left[1 - \frac{1}{2^r} \frac{\Delta u}{q_u} - \frac{1}{2^r} \frac{\Delta z}{q_z} + \frac{1}{2^{2r}} \frac{\Delta u \Delta z}{q_u q_z} \right] \\ &\quad + y(u_{i+1}, z_j) \left[\frac{1}{2^r} \frac{\Delta u}{q_u} - \frac{1}{2^{2r}} \frac{\Delta u \Delta z}{q_u q_z} \right] \\ &\quad + y(u_i, z_{j+1}) \left[\frac{1}{2^r} \frac{\Delta z}{q_z} - \frac{1}{2^{2r}} \frac{\Delta u \Delta z}{q_u q_z} \right] \\ &\quad + y(u_{i+1}, z_{j+1}) \frac{1}{2^{2r}} \frac{\Delta u \Delta z}{q_u q_z} \\ &= \frac{1}{2^{2r}} \left[y(u_i, z_j) \left(2^r - \frac{\Delta u}{q_u} \right) \left(2^r - \frac{\Delta z}{q_z} \right) + y(u_{i+1}, z_{j+1}) \frac{\Delta u \Delta z}{q_u q_z} \right. \\ &\quad \left. + y(u_{i+1}, z_j) \frac{\Delta u}{q_u} \left(2^r - \frac{\Delta z}{q_z} \right) + y(u_i, z_{j+1}) \frac{\Delta z}{q_z} \left(2^r - \frac{\Delta u}{q_u} \right) \right] \\ &= \frac{1}{2^{2r}} \sum_{m=0}^1 \sum_{n=0}^1 k_{m,n} y(u_{i+m}, z_{j+n}) \end{aligned}$$

2.3 Kennfeldinterpolation

Rechnergestützte Kennfeldberechnung

- In Summenschreibweise:

$$y(u, z) = \frac{1}{2^{2r}} \sum_{m=0}^1 \sum_{n=0}^1 k_{m,n} \cdot y(u_{i+m}, z_{j+n})$$

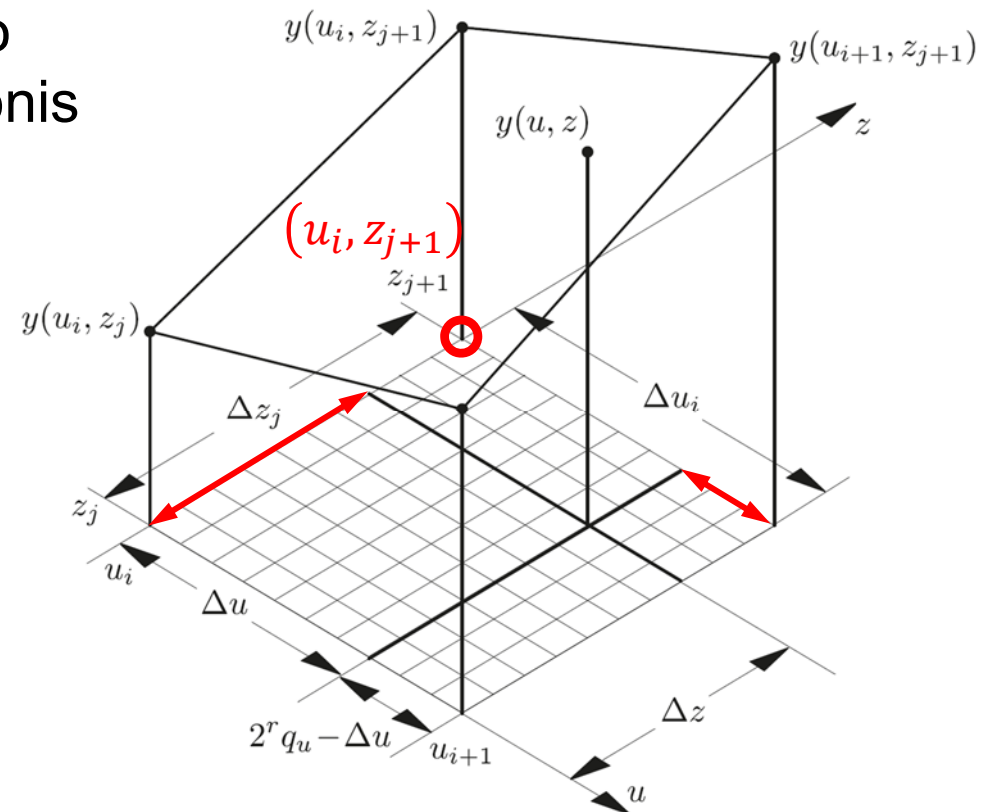
$$\text{mit } k_{m,n} = \left((m-1)2^r + (-1)^m \frac{\Delta u}{q_u} \right) \left((n-1)2^r + (-1)^n \frac{\Delta z}{q_z} \right)$$

d. h. Gewichtung mit dem Produkt der gegenüberliegenden Teilintervalle

2.3 Kennfeldinterpolation

Rechnergestützte Kennfeldberechnung

- Gewichtung mit dem Produkt der gegenüberliegenden Teilintervalle, z. B. $k_{0,1} = \left(-2^r + \frac{\Delta u}{q_u}\right) \left(-\frac{\Delta z}{q_z}\right) = \frac{2^r q_u - \Delta u}{q_u} \cdot \frac{\Delta z}{q_z}$
- Produkt der Teilintervalle $(u_{i+1} - \Delta u)\Delta z$
- Je näher die Interpolationsstelle (u, z) an einer Stützstelle rückt, desto stärker geht diese in das Ergebnis ein



2.3 Kennfeldinterpolation

Rechnergestützte Kennfeldberechnung

- Summe der Koeffizienten:

$$\begin{aligned}\sum_{m=0}^1 \sum_{n=0}^1 k_{m,n} &= \left(-2^r + \frac{\Delta u}{q_u}\right) \left(-2^r + \frac{\Delta z}{q_z}\right) + \left(-2^r + \frac{\Delta u}{q_u}\right) \left(-\frac{\Delta z}{q_z}\right) \\ &\quad + \left(-\frac{\Delta u}{q_u}\right) \left(-2^r + \frac{\Delta z}{q_z}\right) + \left(-\frac{\Delta u}{q_u}\right) \left(-\frac{\Delta z}{q_z}\right) \\ &= 2^{2r} - 2^r \frac{\Delta u}{q_u} - 2^r \frac{\Delta z}{q_z} + \frac{\Delta u \Delta z}{q_u q_z} + 2^r \frac{\Delta z}{q_z} - \frac{\Delta u \Delta z}{q_u q_z} \\ &\quad + 2^r \frac{\Delta u}{q_u} - \frac{\Delta u \Delta z}{q_u q_z} + \frac{\Delta u \Delta z}{q_u q_z} \\ &= 2^{2r}\end{aligned}$$