2. Kurvenanpassung

- Oft ist das Modell eines Messsystems unbekannt und die stationäre Kennlinie liegt nur in Form von n Messpunkten vor.

Ziel der Kurvenanpassung ist die Konstruktion einer mathematischen Funktion, die die Messpunkte geeignet nachbildet. Dadurch können für beliebige Zwischenwerte u die zugehörigen Werte y angegeben werden.

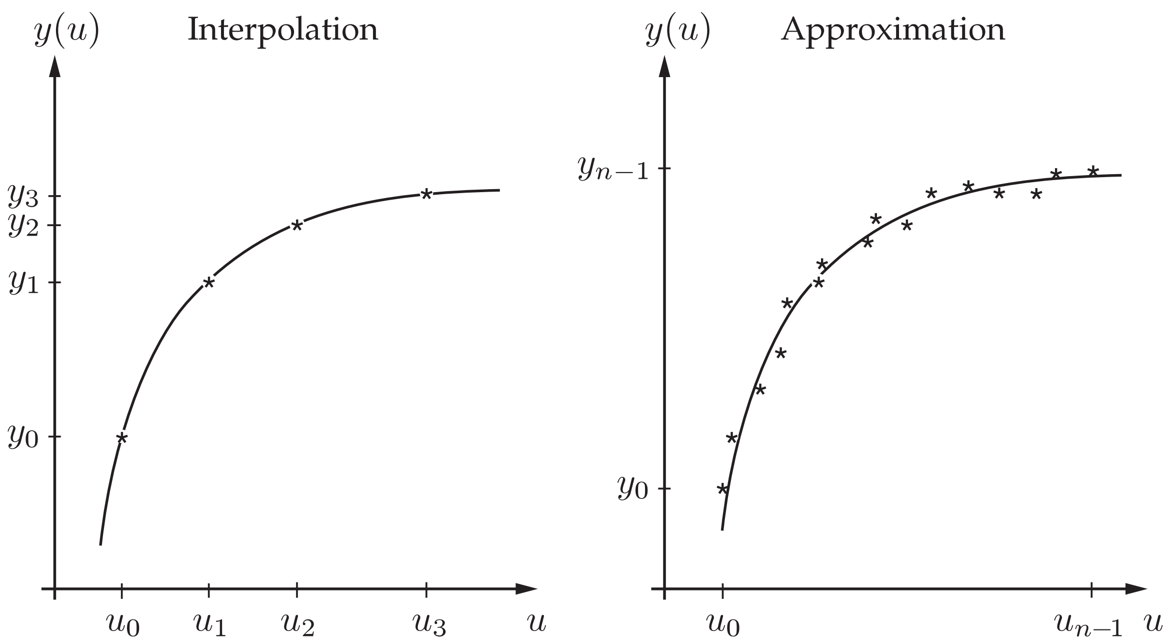
Ansätze

1. **Interpolation:** Die Kennlinie soll durch **alle** n Messpunkte gehen.

   ➡ kleines n

2. **Approximation:** Die Summe der Fehlerquadrate soll minimiert werden. Die Kennlinie geht i. d. R. **nicht** durch alle Messpunkte.

   ➡ großes n



➡ kleines n ➡ großes n

2.1 Approximation

Approximation mit Funktionensystemen

- **Ziel:** Darstellung der Kennlinie als **endliche** Reihe analytischer Funktionen (z. B. Polynome, trigonometrische Funktionen):

$$\hat{y}(u) = \sum_{i=0}^{m-1} a_i\,\varphi_i(u) \quad \text{bzw.} \quad \hat{y}_k = \sum_{i=0}^{m-1} a_i\,\varphi_i(u_k), \qquad k = 0, \ldots, n-1$$

- Bestimmung der Koeffizienten a_i über **Minimierung eines Gütemaßes** Q (hier: Energie des Approximationsfehlers):

$$Q = \sum_{k=0}^{n-1} (y_k - \hat{y}_k)^2 \quad \rightarrow \quad \min$$

$$\rightarrow \quad a_i, \quad i = 0, \ldots, m-1$$

- Berechnung der Koeffizienten a_i unter Umständen aufwendig

- Bei Verwendung einer **orthonormalen Basis**

$$\langle \varphi_i(u_k), \varphi_j(u_k) \rangle = \sum_{k=0}^{n-1} \varphi_i(u_k)\,\varphi_j^*(u_k) = \delta_{ij} = \begin{cases} 1 & , \quad i = j \\ 0 & , \quad i \neq j \end{cases}$$

werden die im Sinne des Gütemaßes Q optimalen **Koeffizienten** a_i einfach durch **Projektion** berechnet [SuS, Kap. 2]:

$$a_i = \langle y_k, \varphi_i(u_k) \rangle = \sum_{k=0}^{n-1} y_k\,\varphi_i^*(u_k)$$

Weitere Vorteile orthogonaler Basen

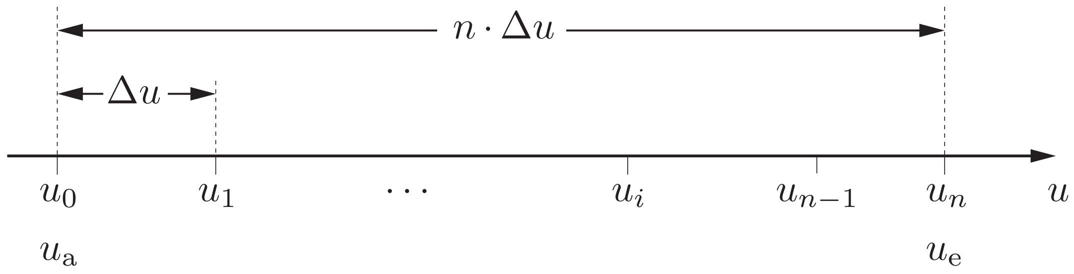
- Da die Koeffizienten a_i nur von den jeweiligen Basisfunktionen abhängen, bleiben diese beim Hinzunehmen weiterer Funktionen **unverändert**
- Mit wachsendem Grad m der Funktionenreihe wird der Approximationsfehler wegen der Bessel'schen Ungleichung geringer [SuS, Kap. 2]:

$$Q = \|y_k\|^2 - \|a_i\|^2$$

Karlsruher Institut für Technologie

Beispiel: Orthonormales Funktionensystem

- Funktionen der Fourier-Reihe: $$F_i\left(u\right) = \frac{1}{\sqrt{n}} \cdot \mathrm{e}^{\mathrm{j}2\pi i \frac{u-u_\mathrm{a}}{u_\mathrm{e}-u_\mathrm{a}}}$$
- Die Funktionen $F_i(u_k)$ bilden im Messbereichsintervall $[u_\mathrm{a}, u_\mathrm{e}]$ bei äquidistanten Stützstellen im Abstand Δu ein orthonormales Funktionensystem:



Intervallbreite: $(u_\mathrm{e} - u_\mathrm{a}) = n \cdot \Delta u$

k-te Stützstelle: $u_k = k \cdot \Delta u + u_\mathrm{a}$

$$\rightarrow \quad \frac{u_k - u_\mathrm{a}}{u_\mathrm{e} - u_\mathrm{a}} = \frac{k}{n}$$

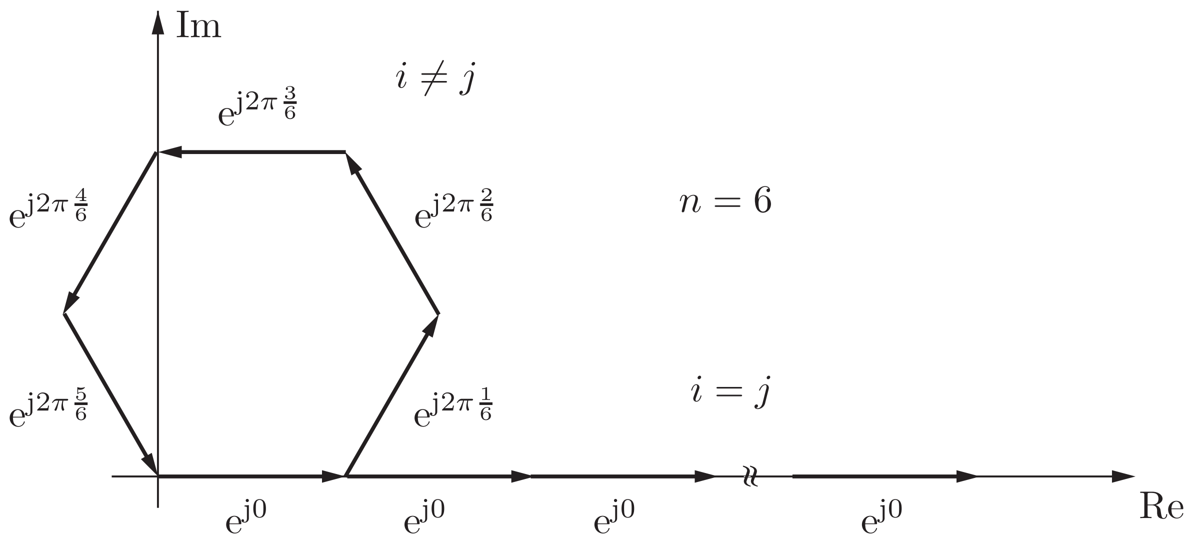
2.1.1 Approx. mit orthonormalen Funktionensystemen

Karlsruher Institut für Technologie

- Durch das Innenprodukt lässt sich die Orthogonalität zeigen:

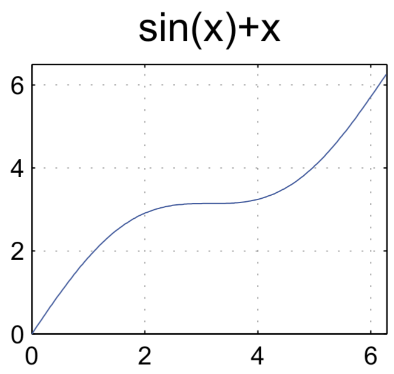
$$\langle F_i(u_k), F_j(u_k)\rangle = \sum_{k=0}^{n-1} F_i(u_k) \cdot F_j^*(u_k) = \frac{1}{n}\sum_{k=0}^{n-1} \mathrm{e}^{\mathrm{j}2\pi(i-j)\frac{k}{n}} = \delta_{ij}$$

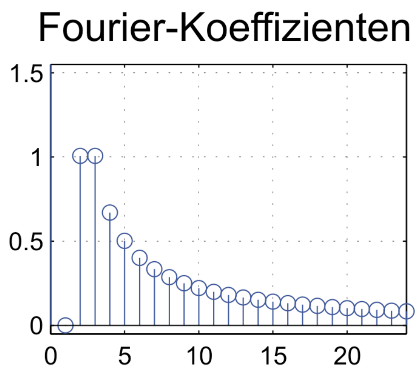
- Veranschaulichung in der komplexen Ebene:

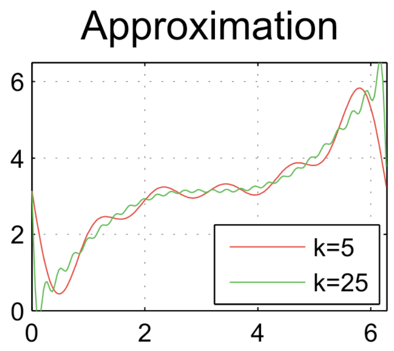


2.1.1 Approx. mit orthonormalen Funktionensystemen

Beispiel: Approximation mit den Fourier-Koeffizienten







Eigenschaften

- Periodizität
- Oszillationen

2.1.2 Least-Squares-Schätzer

- Vielfach **allgemeine** (nicht orthogonale) **Basisfunktionen** erwünscht

> Zur Bestimmung der Koeffizienten a_i wird bei **nicht orthogonalen** Basisfunktionen $\varphi_i(u)$ der **Least-Squares-Schätzer** verwendet

- Ansatz (Energie des Approximationsfehlers) in Vektorschreibweise:

$$\boxed{Q = \sum_{k=0}^{n-1} (y_k - \hat{y}_k)^2 = (\mathbf{y} - \hat{\mathbf{y}})^{\mathrm{T}} (\mathbf{y} - \hat{\mathbf{y}})} \quad \rightarrow \quad \min$$

- Messpunkte liegen diskret vor → Approximationsansatz in Matrixnotation:

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

$$\Rightarrow \quad Q = (\mathbf{y} - \mathbf{\Phi}\,\mathbf{a})^{\mathrm{T}} (\mathbf{y} - \mathbf{\Phi}\,\mathbf{a}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}^{\mathrm{T}}\mathbf{y} + \mathbf{a}^{\mathrm{T}}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\,\mathbf{a}$$

- Minimierung des Gütemaßes zur Bestimmung von $\mathbf{a}$:

$$\frac{\mathrm{d}Q}{\mathrm{d}\mathbf{a}} = -2\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{y} + 2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\,\mathbf{a} = 0$$

$$\rightarrow \quad \boxed{\mathbf{a} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\,\mathbf{y}}$$

➡ Der Parametervektor $\mathbf{a}$ berechnet sich aus der **Pseudoinversen** von $\boldsymbol{\Phi}$ (Verallgemeinerung für nichtquadr. Matrizen) und dem Messpunktevektor

Diskussion

- LS-Schätzer hat eine große praktische Bedeutung, um Kennlinien oder Signalverläufe aus stark verrauschten Daten zu bestimmen (Regressionsrechnung)
- LS-Schätzer ist ein **Optimalfilter**
- Voraussetzung für den erfolgreichen Einsatz ist das Vorhandensein von Vorwissen über den Kennlinienverlauf (sog. **Signalmodell**), welches in die Matrix $\boldsymbol{\Phi}$ der Basisfunktionen eingeht

Beispiel: Lineare Regression

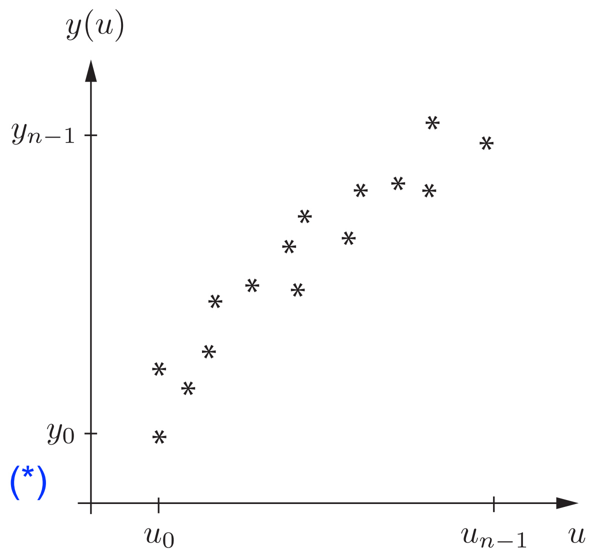
- Gesucht: Gerade durch eine Menge von Messpunkten $\boxed{\hat{y}(u) = a_1 u + a_0}$
- Gütemaß: $Q = \sum_{k=0}^{n-1} (y_k - a_1 u_k - a_0)^2$
- Minimierung:

$$\frac{\mathrm{d}Q}{\mathrm{d}a_1} = -2\sum_{k=0}^{n-1} u_k\,(y_k - a_1 u_k - a_0) \stackrel{!}{=} 0$$

$$\frac{\mathrm{d}Q}{\mathrm{d}a_0} = -2\sum_{k=0}^{n-1} (y_k - a_1 u_k - a_0) \stackrel{!}{=} 0$$

$$\rightarrow \quad a_1 \sum_{k=0}^{n-1} u_k^2 \;+\; a_0 \sum_{k=0}^{n-1} u_k = \sum_{k=0}^{n-1} u_k\, y_k \qquad (*)$$

$$a_1 \sum_{k=0}^{n-1} u_k \;+\; n\, a_0 = \sum_{k=0}^{n-1} y_k \qquad (**)$$



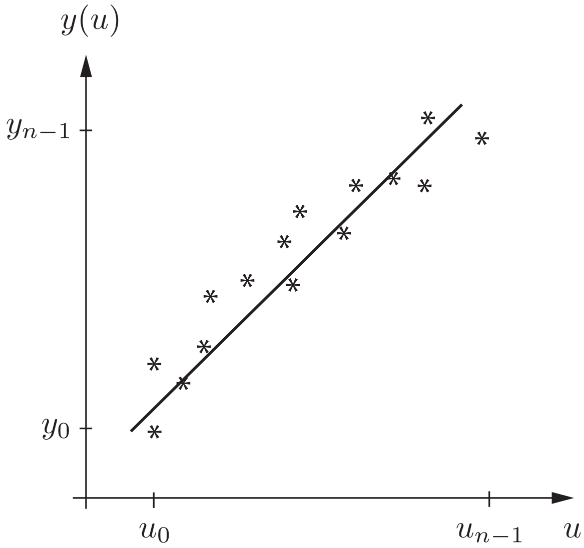
2.1.3 Regressionsanalyse

- Auflösen von (**) nach dem Parameter a_0:

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

- Einsetzen in (*) und Auflösen nach a_1 ergibt:

$$a_1 = \frac{n \cdot \sum\limits_{k=0}^{n-1} u_k y_k - \sum\limits_{k=0}^{n-1} u_k \sum\limits_{k=0}^{n-1} y_k}{n \cdot \sum\limits_{k=0}^{n-1} u_k^2 - \left(\sum\limits_{k=0}^{n-1} u_k\right)^2}$$



- Regressionsrechnung liefert **gleiches Ergebnis** wie der LS-Schätzer

2.1.3 Regressionsanalyse

Alternative Vorgehensweise

- Direkte Verwendung der Gleichung für den LS-Schätzer:

$$\mathbf{a} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\,\mathbf{y}$$

- Signalmodell $\hat{y}(u_k) = a_1 u_k + a_0$ in Matrixnotation:

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_0 \\ \vdots \\ \hat{y}_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & u_0 \\ \vdots & \vdots \\ 1 & u_{n-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \boldsymbol{\Phi}\,\mathbf{a}$$

- Es können natürlich auch Polynome höheren Grades verwendet werden!

Beim Least-Squares-Schätzer müssen **alle** Koeffizienten a_k neu berechnet werden, wenn die Ordnung des Regressionspolynoms erhöht wird.

■

2.2 Interpolation

2.2.1 Polynominterpolation

- **Annahme:** wenige Punkte einer Kennlinie gegeben → **Interpolation**

> Bei der Interpolation werden die Messwerte $(u_k,\, y_k)$ der Kennlinie in den Stützstellen **exakt** wiedergegeben.

- Meist **Polynomansatz** in der Messgröße u:

$$\hat{y}(u) = \sum_{i=0}^{n-1} a_i\, u^i = \mathbf{a}^{\mathrm{T}}\mathbf{p} \qquad (2.37)$$

mit $\mathbf{a}$: Koeffizienten, $\mathbf{p} = (1,\, u, \ldots,\, u^{n-1})^{\mathrm{T}}$: Potenzen u^i von u

- **Vorgehensweise:** Bestimmung der n Koeffizienten a_i durch n Gleichungen in den Stützstellen (Wertepaare u_k, y_k):

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

2.2.1 Polynominterpolation

- Gleichungssystem lässt sich in Matrixschreibweise formulieren:

$$\mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & u_0 & u_0^2 & \cdots & u_0^{n-1} \\ 1 & u_1 & u_1^2 & \cdots & u_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n-1} & u_{n-1}^2 & \cdots & u_{n-1}^{n-1} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \mathbf{V} \cdot \mathbf{a}$$

$\mathbf{V}$: Vandermonde-Matrix

- Die Matrix $\mathbf{V}$ ist invertierbar (regulär), falls die u_i paarweise verschieden sind – und ferner gut konditioniert, wenn die Stützstellen hinreichend weit auseinander liegen. Dann lassen sich die Koeffizienten berechnen:

$$\mathbf{a} = \mathbf{V}^{-1}\,\mathbf{y}$$

→ Interpolationsgleichung: $\hat{y} = \mathbf{a}^{\mathrm{T}}\mathbf{p} = \mathbf{p}^{\mathrm{T}}\mathbf{a} = \mathbf{p}^{\mathrm{T}}\,\mathbf{V}^{-1}\,\mathbf{y}$

2.2.2 Interpolation durch Lagrange-Polynome

- In der Praxis arbeitet man lieber mit **Lagrange-Polynomen**
- **Grund:** Dank komplexerer Basispolynome $L_i(u)$ anstelle der Monome u^i werden Lagrange-Polynome direkt mit den Messwerten y_i gewichtet → keine Berechnung der Koeffizienten erforderlich:

Ansatz: $\hat{y} = \sum_{i=0}^{n-1} y_i\, L_i(u)$

$L_i(u)$: Polynom in u (Lagrange-Polynom)

Vektorschreibweise: $\hat{y} = \mathbf{L}^{\mathrm{T}}\,\mathbf{y}$

→ $\mathbf{L} = \left(\mathbf{V}^{-1}\right)^{\mathrm{T}}\mathbf{p}$

Vergleich:

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

$$\hat{y} = \mathbf{a}^{\mathrm{T}}\mathbf{p} = \underbrace{\mathbf{p}^{\mathrm{T}}\,\mathbf{V}^{-1}}_{\mathbf{L}^{\mathrm{T}}}\,\mathbf{y}$$

Lagrange-Polynome: $L_i(u) = \dfrac{(u-u_0)\cdots(u-u_{i-1})(u-u_{i+1})\cdots(u-u_{n-1})}{(u_i-u_0)\cdots(u_i-u_{i-1})(u_i-u_{i+1})\cdots(u_i-u_{n-1})}$

2.2.2 Interpolation durch Lagrange-Polynome

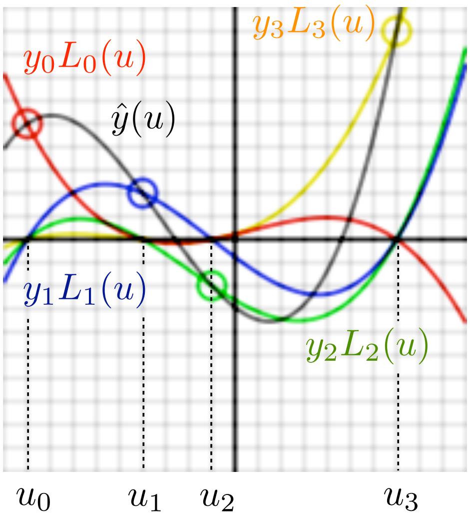
- Lagrange-Polynome:

$$L_i(u) = \frac{(u-u_0)\cdots(u-u_{i-1})(u-u_{i+1})\cdots(u-u_{n-1})}{(u_i-u_0)\cdots(u_i-u_{i-1})(u_i-u_{i+1})\cdots(u_i-u_{n-1})}$$

- Eigenschaft: $\boxed{L_i(u_j) = \delta_{ij}}$

➡ Die Stützstellen eines Lagrange-Polynoms werden exakt interpoliert:

$$\boxed{\hat{y}(u_j) = \sum_{i=0}^{n-1} y_i\, L_i(u_j) = y_j}$$



Quelle: Wikipedia

2.2.2 Interpolation durch Lagrange-Polynome

Beispiel 2.1: Lagrange-Interpolation mit drei Stützstellen

- Kennlinie sei durch drei **äquidistante** Stützstellen u_a, u_m, u_e gegeben ($h = u_\mathrm{a} - u_\mathrm{m}$ beschreibe den Abstand zwischen benachbarten Stützstellen)
- Die **Lagrange-Polynome** sind damit:

$$L_0(u) = \frac{(u-u_\mathrm{m})\,(u-u_\mathrm{e})}{(u_\mathrm{a}-u_\mathrm{m})\,(u_\mathrm{a}-u_\mathrm{e})} = \frac{1}{2h^2}\,(u-u_\mathrm{m})\,(u-u_\mathrm{e})$$

$$L_1(u) = \frac{(u-u_\mathrm{a})\,(u-u_\mathrm{e})}{(u_\mathrm{m}-u_\mathrm{a})\,(u_\mathrm{m}-u_\mathrm{e})} = -\frac{1}{h^2}\,(u-u_\mathrm{a})\,(u-u_\mathrm{e})$$

$$L_2(u) = \frac{(u-u_\mathrm{a})\,(u-u_\mathrm{m})}{(u_\mathrm{e}-u_\mathrm{a})\,(u_\mathrm{e}-u_\mathrm{m})} = \frac{1}{2h^2}\,(u-u_\mathrm{a})\,(u-u_\mathrm{m})$$

- In die Interpolationsgleichung eingesetzt:

$$\boxed{\hat{y} = \sum_{i=0}^{2} y_i L_i\,(u)} = \frac{y_\mathrm{a}}{2h^2}(u-u_\mathrm{m})(u-u_\mathrm{e}) - \frac{y_\mathrm{m}}{h^2}(u-u_\mathrm{a})(u-u_\mathrm{e}) + \frac{y_\mathrm{e}}{2h^2}(u-u_\mathrm{a})(u-u_\mathrm{m})$$

2.2.2 Interpolation durch Lagrange-Polynome

$$\hat{y} = \frac{y_\mathrm{a}}{2h^2}(u-u_\mathrm{m})(u-u_\mathrm{e}) - \frac{y_\mathrm{m}}{h^2}(u-u_\mathrm{a})(u-u_\mathrm{e}) + \frac{y_\mathrm{e}}{2h^2}(u-u_\mathrm{a})(u-u_\mathrm{m})$$

- Für $u_\mathrm{a} = 0,\ y_\mathrm{a} = 0$ sowie $u_\mathrm{m} = h,\ u_\mathrm{e} = 2h$ folgt die Kennlinie:

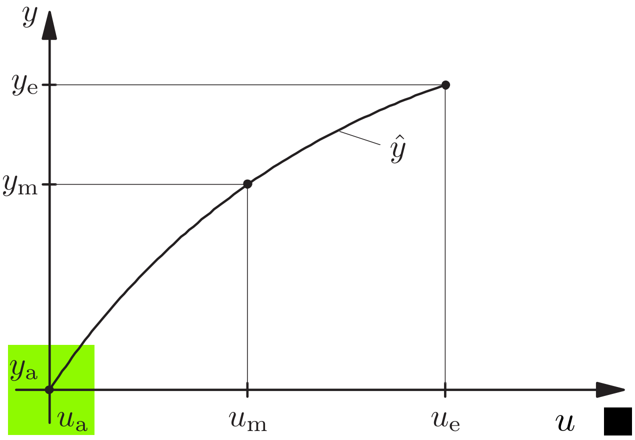
$$\hat{y} = u\left(\frac{2\,y_\mathrm{m}}{h} - \frac{y_\mathrm{e}}{2\,h}\right) + u^2\left(\frac{y_\mathrm{e}}{2\,h^2} - \frac{y_\mathrm{m}}{h^2}\right)$$ (Parabel durch den Ursprung)

- Liegen die drei Stützstellen auf einer Geraden,

$$y_\mathrm{m} = \frac{y_\mathrm{e}}{2}\,,$$

so wird die Kennlinie **linear**:

$$\hat{y}_\mathrm{lin} = u \cdot \frac{y_\mathrm{e}}{2h}$$



2.2.3 Interpolation durch Newton-Polynome

Warum Newton-Interpolation?

- geringere Komplexität der Basispolynome
- leichtere **Erweiterbarkeit** beim Hinzufügen von Stützstellen

Ansatz:

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

➡ Rekursive Berechnung der Koeffizienten a_i aus den Interpolationsbedingungen in den Stützstellen:

$$\begin{aligned}
y_0 &= a_0\\
y_1 &= a_0 + a_1\,(u_1-u_0)\\
&\vdots\\
y_{n-1} &= a_0 + a_1\,(u_{n-1}-u_0) + a_2\,(u_{n-1}-u_0)\,(u_{n-1}-u_1) + \cdots\\
&\quad + a_{n-1}\,(u_{n-1}-u_0)\,(u_{n-1}-u_1)\cdots(u_{n-1}-u_{n-2})
\end{aligned}$$

Newton- und **Lagrange**-Interpolation liefern das **gleiche** Polynom

2.2.3 Interpolation durch Newton-Polynome

- Zur einfachen Berechnung der Koeffizienten bei **äquidistanten Stützstellen** werden **Differenzen** von Funktionen eingeführt

- Erste Differenzen: $\Delta y_0 = y_1 - y_0,\ \Delta y_1 = y_2 - y_1,\ \Delta y_2 = y_3 - y_2, \cdots$

- Zweite Differenzen:
$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = y_2 - 2y_1 + y_0$$
$$\Delta^2 y_1 = \Delta y_2 - \Delta y_1 = y_3 - 2y_2 + y_1$$

- Höhere Differenzen: $\Delta^j y_i = \Delta^{j-1}(\Delta y_i) = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i$

- Für das Rechnen mit Differenzenoperatoren gilt:
$$y_i = (1 + \Delta)^i\ y_0$$
Beweis mittels vollständiger Induktion [PK12].

2.2.3 Interpolation durch Newton-Polynome

- Durch Ausmultiplizieren von $y_i = (1 + \Delta)^i\ y_0$ mit der allgemeinen binomischen Formel erhält man:

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

- Aus dem **Interpolationsansatz** auf Folie 22 folgt bei **konstantem Stützstellenabstand** h:

$$\begin{aligned}
y_0 &= a_0 \\
y_1 &= a_0 + a_1\, h \\
&\vdots \\
y_i &= a_0 + a_1\, i\, h + \cdots + a_j\, i(i-1)\cdots(i-j+1)\, h^j + \cdots + a_i\, i!\, h^i \\
&\vdots \\
y_{n-1} &= a_0 + a_1\,(n-1)\,h + a_2\,(n-1)(n-2)h^2 + \cdots + a_n\,(n-1)!\,h^n
\end{aligned}$$

- Durch Vergleich der Koeffizienten vom Grad j erhält man:

$$a_j \cdot i(i-1)\cdots(i-j+1) \cdot h^j = i(i-1)\cdots(i-j+1) \cdot \Delta^j y_0 / j!$$

- Daraus folgt für die Koeffizienten des Newton-Polynoms: $\boxed{a_j = \dfrac{\Delta^j y_0}{j!\, h^j}}$

➡ **Newton'sche Interpolationsformel** für **äquidistante** Stützstellen:

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

- Die Differenzen höherer Ordnung lassen sich auf einfache Weise aus dem **Differenzenschema** durch fortlaufende Subtraktion bestimmen:

$$\begin{array}{c|ccccc}
u & y & \Delta y & \Delta^2 y & \Delta^3 y & \Delta^4 y \\
\hline
0 & y_0 & & & & \\
 & & \rangle\, \Delta y_0 & & & \\
h & y_1 & & \rangle\, \Delta^2 y_0 & & \\
 & & \rangle\, \Delta y_1 & & \rangle\, \Delta^3 y_0 & \\
2h & y_2 & & \rangle\, \Delta^2 y_1 & & \rangle\, \Delta^4 y_0 \\
 & & \rangle\, \Delta y_2 & & \rangle\, \Delta^3 y_1 & \\
3h & y_3 & & \rangle\, \Delta^2 y_2 & & \\
 & & \rangle\, \Delta y_3 & & & \\
4h & y_4 & & & &
\end{array}$$

Beispiel 2.2: Newton-Interpolation mit drei Stützstellen

- Messkennlinie durch drei **äquidistante** Stützstellen u_a, u_m, u_e gegeben:

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

- Differenzenschema:

$$\begin{array}{c|ccc}
u & y & \Delta y & \Delta^2 y \\
\hline
0 & y_a & & \\
 & & \rangle\, y_m - y_a & \\
h & y_m & & \rangle\, y_e - 2y_m + y_a \\
 & & \rangle\, y_e - y_m & \\
2h & y_e & &
\end{array}$$

- Interpolationsfunktion: $\boxed{\hat{y} = \dfrac{y_m}{h} u + \dfrac{y_e - 2y_m}{2h^2} u\,(u-h)}$

➡ Ergebnis stimmt mit dem aus Bsp. 2.1 überein! ∎

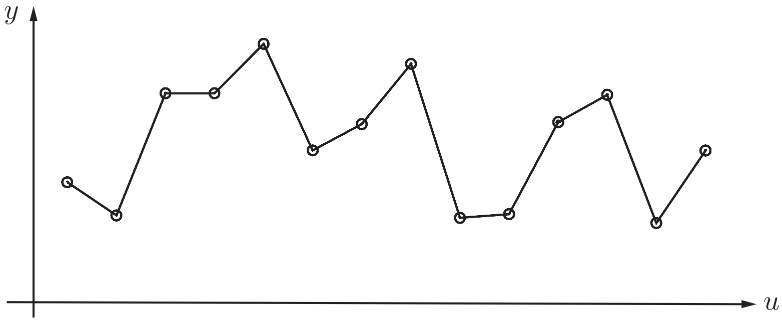
- Nachteil der **klassischen** Polynominterpolation:

  > Bei **vielen Stützstellen** erhält man Interpolationspolynome **hohen Grades** (→ stark oszillierendes Verhalten)

- **Abhilfe**: Den Teilintervallen zwischen Stützstellen werden Polynome **niedrigen** Grades zugeordnet

Einfachstes Verfahren: Lineare Interpolation

- Bei n Messpunkten besteht die Interpolationsfunktion aus $n-1$ Geradenstücken in den Intervallen $[u_i, u_{i+1}], \quad i \in \{0, \ldots, n-2\}$

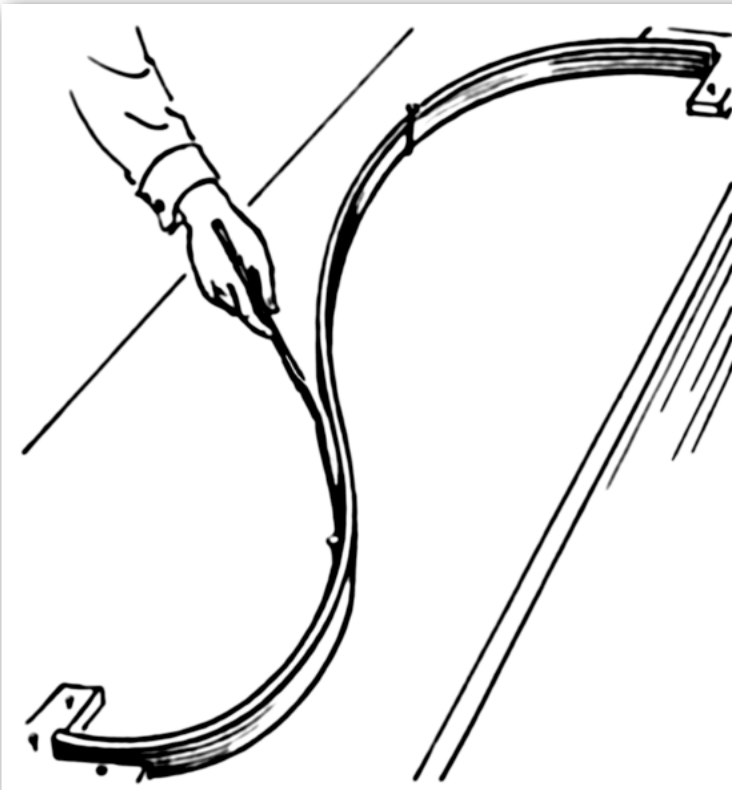


→ für Kennlinien unbrauchbar, da Ableitung („Empfindlichkeit") unstetig

2.2.4 Spline-Interpolation

Spline-Funktionen

- Inspiriert durch biegsame, im Schiffbau eingesetzte Lineale
- An den Stützstellen wird eine dünne Latte fixiert; es wirken **keine** weiteren äußeren Kräfte auf die Latte ein
- Die Latte verformt sich und es entsteht eine **glatte** Biegelinie durch alle Stützstellen mit **minimaler Biegeenergie** und **kleinen Krümmungen** – die Interpolierende $s(u)$



Quelle: Wikipedia

2.2.4 Spline-Interpolation

Karlsruher Institut für Technologie

Spline-Funktionen

- Vorteilhaft an den Spline-Funktionen ist, dass die Wendepunkte (Stellen maximaler Linearität) i. d. R. zwischen den Stützstellen liegen; bei der Polynominterpolation liegen sie dagegen nahe an den Stützstellen
- Die aufgrund der Biegekraft **in der Latte gespeicherte Energie** ist proportional zum Integral über das Quadrat der **„Krümmung“**:

$$\boxed{E = \frac{1}{2} \int\limits_{u_0}^{u_{n-1}} \left(s''(u)\right)^2 \, \mathrm{d}u}$$

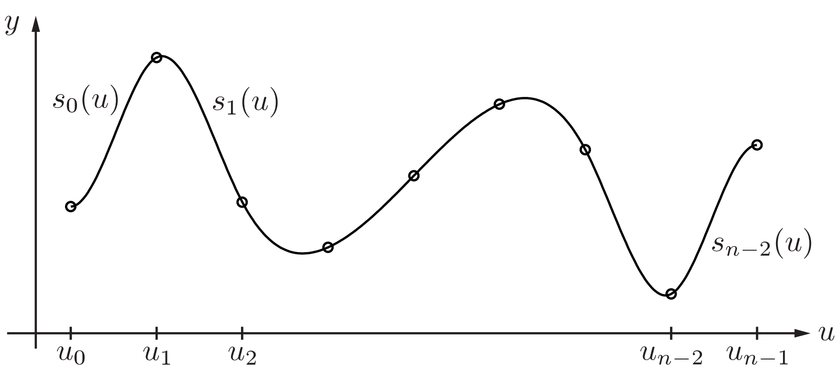
- Der stabile Arbeitspunkt stellt sich bei einer **minimalen** Energie ein:

$$\boxed{\int\limits_{u_0}^{u_{n-1}} \left(s''(u)\right)^2 \, \mathrm{d}u \quad \rightarrow \quad \min} \tag{2.84}$$

- Die gesuchte Spline-Funktion $s(u)$ minimiert die Energie unter der Voraussetzung, dass $s(u)$ **mindestens einmal stetig differenzierbar** ist

- Durch Variationsrechnung erhält man aus dem Energieminimierungsansatz (2.84) folgende **Eigenschaften der Spline-Interpolierenden**:

$s_i(u_i) = y_i\,, \qquad i \in \{0, \ldots, n-1\}$	Interpolationsbedingung
$s_i''(u_i + 0) = s_{i-1}''(u_i - 0)\,, \quad i \in \{1, \ldots, n-2\}$	stetige 2. Ableitung
$s_0''(u_0) = s_{n-2}''(u_{n-1})$	Randbed. für 2. Ableitung
$s''''(u) = 0\,, \qquad u \neq u_0, \ldots, u_{n-1}$	kubische Polynome



Berechnung der kubischen Spline-Interpolierenden:

Für jedes Intervall $[u_i\,,\,u_{i+1}]$ der Länge $h_i = u_{i+1} - u_i$ wird als Ansatz ein **Polynom 3. Grades** gewählt

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

- Funktionswerte und erste beiden Ableitungen an den Stützstellen:

$$\begin{aligned}
s_i(u_i) &= d_i &&= y_i \\
s_i(u_{i+1}) &= a_i\,h_i^3 + b_i\,h_i^2 + c_i\,h_i + d_i &&= y_{i+1} \\
s_i'(u_i) &= c_i \\
s_i'(u_{i+1}) &= 3\,a_i\,h_i^2 + 2\,b_i\,h_i + c_i \\
s_i''(u_i) &= 2b_i &&= y_i'' \\
s_i''(u_{i+1}) &= 6\,a_i\,h_i + 2\,b_i &&= y_{i+1}''
\end{aligned}$$

→ Auflösen nach den unbekannten Parametern a_i, b_i, c_i und d_i

2.2.4 Spline-Interpolation

➡ Alle **unbekannten Parameter** a_i, b_i, c_i und d_i können durch die gegebenen **Stützpunkte** y_i und y_{i+1} sowie die **noch unbekannten zweiten Ableitungen** y_i'' und y_{i+1}'' ausgedrückt werden:

$$\begin{aligned}
a_i &= \frac{1}{6h_i}\,(y_{i+1}'' - y_i'') \\
b_i &= \frac{1}{2}\,y_i'' \\
c_i &= \frac{1}{h_i}\,(y_{i+1} - y_i) - \frac{1}{6}\,h_i\,(y_{i+1}'' + 2y_i'') \\
d_i &= y_i
\end{aligned}$$

Berechnung der zweiten Ableitungen:

- Forderung nach **Stetigkeit** der ersten Ableitung an den Stützstellen:

$$\boxed{s_i'(u_{i+1}) = s_{i+1}'(u_{i+1})}$$

- Nach Einsetzen der Parameter a_i, b_i und c_i in die Ableitungen folgt:

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

$$s_{i+1}'(u_{i+1}) = c_{i+1} = \frac{1}{h_{i+1}}\,(y_{i+2} - y_{i+1}) - \frac{h_{i+1}}{6}\,(y_{i+2}'' + 2y_{i+1}'')$$

- Gleichsetzen beider Gleichungen und Ordnen nach den unbekannten zweiten Ableitungen ergibt:

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

- Unter Berücksichtigung von $y_0'' = y_{n-1}'' = 0$ erhält man $(n-2)$ lineare Gleichungen für die unbekannten zweiten Ableitungen $y_1'', y_2'', \ldots, y_{n-2}''$:

$$\begin{bmatrix} 2(h_0+h_1) & h_1 & 0 & & \cdots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & h_{n-3} & 2(h_{n-3}+h_{n-2}) \end{bmatrix} \begin{bmatrix} y_1'' \\ y_2'' \\ \vdots \\ y_{n-2}'' \end{bmatrix} =$$

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

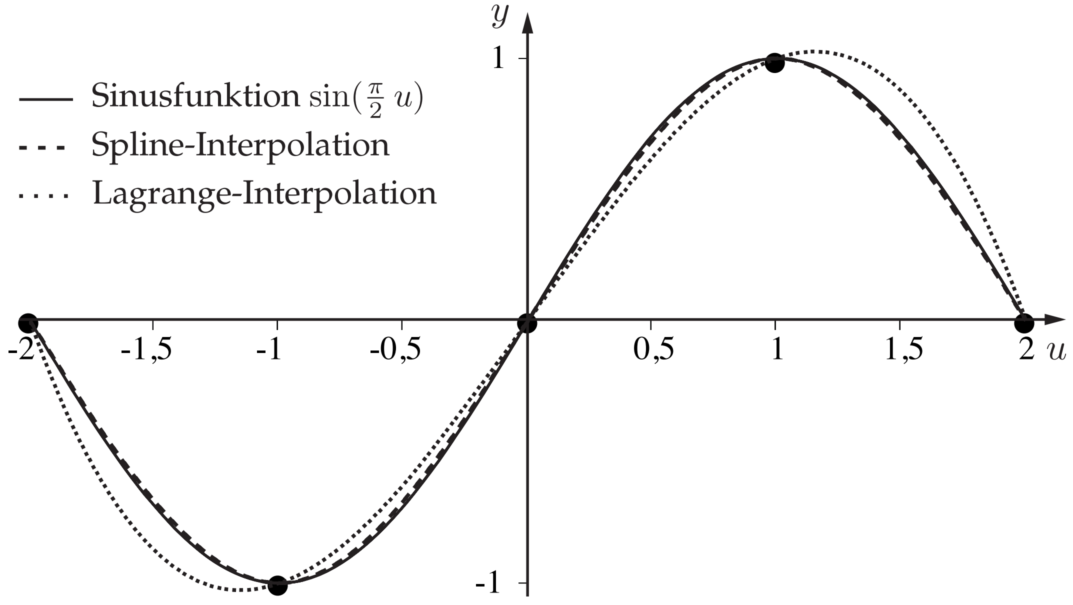
- Nach der Auflösung werden die a_i, b_i und c_i und daraus $s_i(x)$ bestimmt.

2.2.4 Spline-Interpolation

KIT Karlsruher Institut für Technologie

Beispiel 2.3: Interpolation einer Sinusfunktion

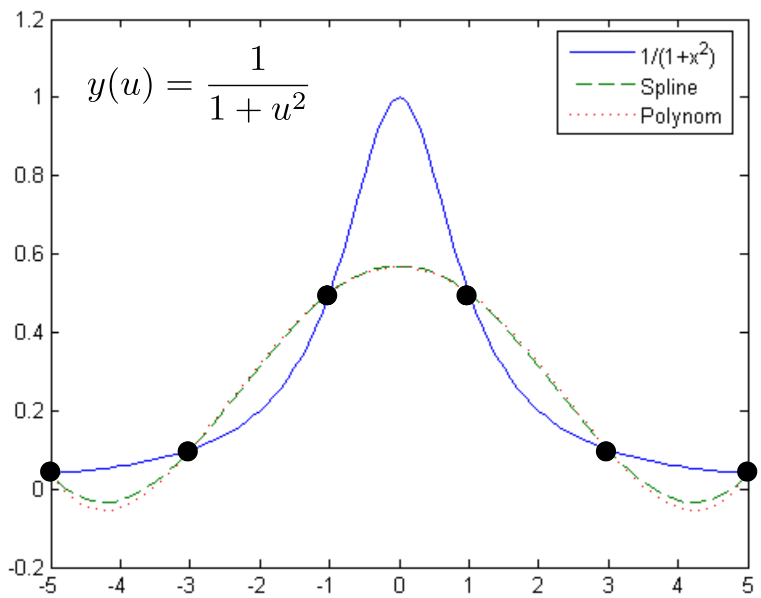
- Vergleich von Spline- und Lagrange-Interpolation



2.2.4 Spline-Interpolation

KIT Karlsruher Institut für Technologie

Beispiel: 6-Punkte-Interpolation

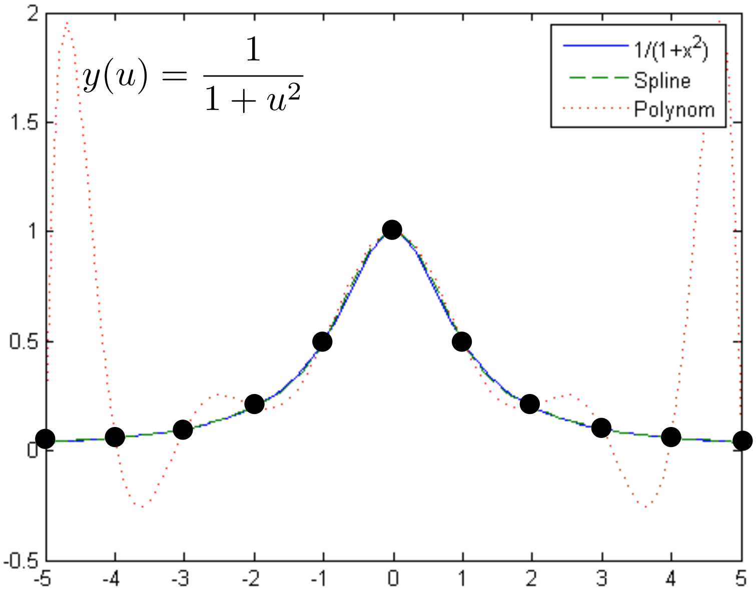


Schlechte Interpolation:

- zu wenige Stützstellen
- Spline-Interpolation ähnlich wie Polynominterpolation

2.2.4 Spline-Interpolation

Beispiel: 11-Punkte-Interpolation



Mehr Interpolationspunkte:

- Spline-Interpolation mit guten Ergebnissen
- Polynominterpolation mit starkem Überschwingen

2.2.5 Systemtheoretische Deutung der Interpolation

- **Annahme:** äquidistante Stützstellen

$$\hat{y}(u) = \Big[\underbrace{y(u)\sum_{n=-\infty}^{\infty}\delta(u - n\,\Delta u)}_{\text{Abtastsignal}}\Big] * \underbrace{i(u)}_{\text{Interpolationsfunktion}}$$

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

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

2.3 Kennfeldinterpolation

- Kennlinie $y = f(u, z)$ eines Systems sei eine nichtlineare Funktion der Messgröße u und einer messbaren systematischen Störgröße z

  → Kompensation der Störgröße z durch Interpolation
- **Voraussetzung:** Kennfeldwerte für genügend Stützstellen (u_i, z_i) bekannt

> Die Kennfeldinterpolation ist eine **2D-Interpolation**. Es werden die Werte y zwischen den äquidistanten Stützstellen $(u_i\,,\, z_j)$ interpoliert.

- Zur Herleitung der Interpolationsformel wird der Polynomansatz (2.37)

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

auf zwei Dimensionen erweitert:

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

- Alternativ darf bei **Polynomen 1. Grades** die Ausgangsgröße y an der Stelle (u_i, z_i) in eine Taylor-Reihe entwickelt werden, die nach den linearen Gliedern abgebrochen wird:

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

- Durch **Approximation** der Ableitungen mittels Differenzenquotienten folgt:

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

Diese Näherung entspricht einer **bilinearen Interpolation**.

Beispiel 2.4: Bilineare Interpolation

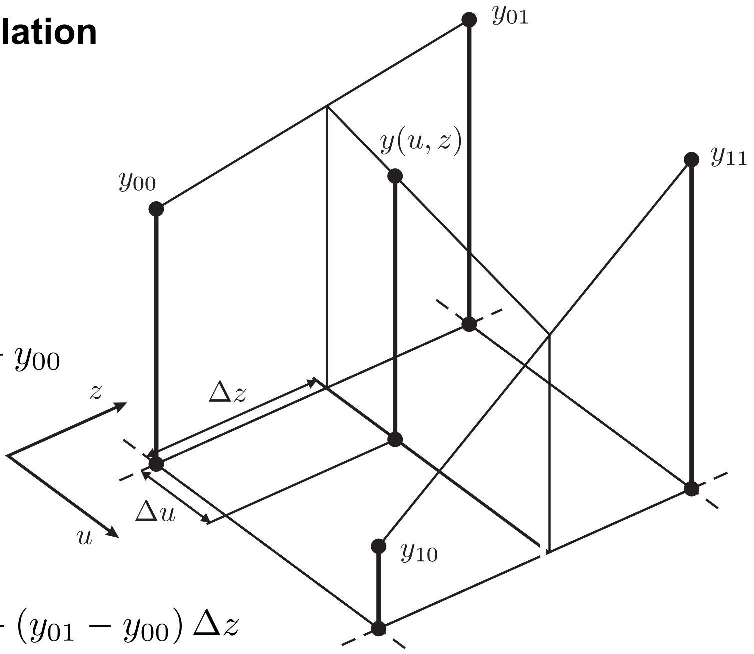
- Die Differenzen lassen sich wie folgt schreiben:

$$\Delta y(u_i) = y_{10} - y_{00}$$

$$\Delta y(z_j) = y_{01} - y_{00}$$

$$\Delta^2 y(u_i, z_j) = y_{11} - y_{10} - y_{01} + y_{00}$$

- Bei normierten Stützstellenweiten $\Delta u_i = \Delta z_j = 1$ folgt:



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

■

Beispiel: Druckmessung mit piezoresistiven Sensoren

- Temperaturabhängige Kennlinie:

$$p = p(T)$$

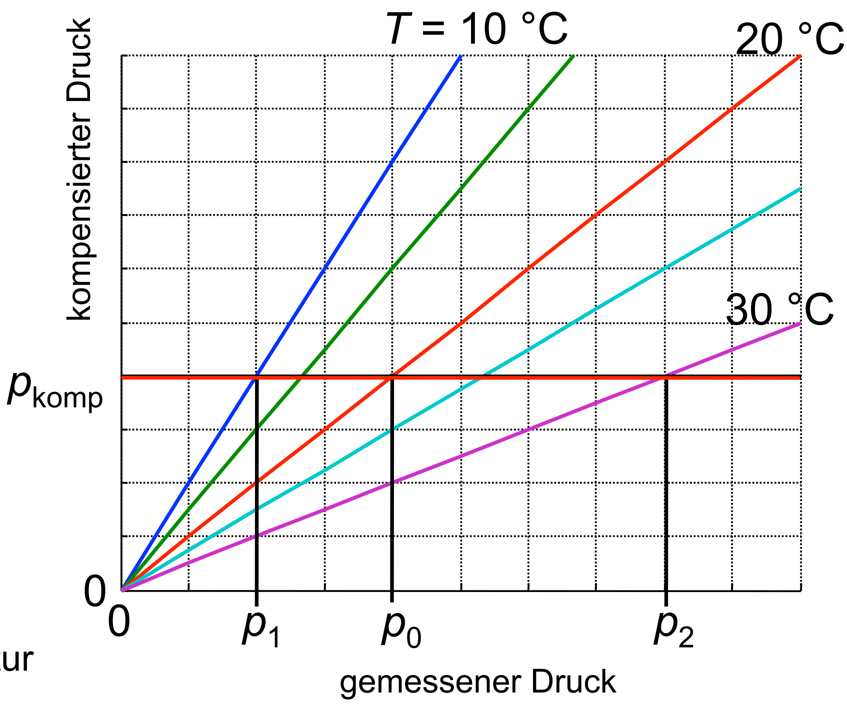
p_0: Richtiger Druck bei T = 20 °C

p_1: Gemessener Druck bei T = 10 °C

p_2: Gemessener Druck bei T = 30 °C

p_{komp} : Gewünschter Anzeigewert

T : Störgröße Temperatur



Kompensation der temperaturverfälschten Druckwerte

p_i : Gemessener Druck

T : Störgröße Temperatur

p_{komp}: Anzeigewert