


# Numerik 1

Lennart  Diwer

October 2022 - Februar 2023

# Inhaltsverzeichnis

<b>1</b>	<b>Grundlegende Konzepte der Numerik</b>	<b>2</b>
1.1	Zahlendarstellung und Rundungsfehler . . . . .	3
1.2	Kondition und Stabilität . . . . .	4
1.3	Landau-Symbole, Genauigkeit und Komplexität . . . . .	5
1.4	Differentielle Fehleranalyse: . . . . .	6
<b>2</b>	<b>Interpolation</b>	<b>9</b>
2.1	Polynominterpolation . . . . .	9
2.1.1	Lagrange-Interpolation . . . . .	11
2.1.2	Newton Darstellung . . . . .	12
2.1.3	Auswertung von Polynomen . . . . .	15
2.1.4	Interpolationsfehler bei der Interpolation einer gegebenen Funktion . . . . .	16
2.1.5	Hermite-Interpolation . . . . .	20
2.1.6	Spline Interpolation . . . . .	23
2.2	Trigonometrische Interpolation . . . . .	29
2.2.1	Zum Hintergrund . . . . .	29
2.2.2	Fourier-Reihen . . . . .	30
2.2.3	Diskrete Fourier-Transformation . . . . .	30
<b>3</b>	<b>Numerische Integration</b>	<b>32</b>
3.1	Numerische Integration . . . . .	32
3.1.1	Interpolatorische Quadraturformel . . . . .	32
3.1.2	Gauß-Quadraturformeln . . . . .	36
3.1.3	Richardson-Extrapolation . . . . .	41
<b>4</b>	<b>Numerische Lösung Linearer Gleichungssysteme</b>	<b>45</b>
4.1	Direkte Verfahren . . . . .	45
4.2	LR-Zerlegung einer Matrix . . . . .	46
4.2.1	LR-Zerlegung von Bandmatrizen . . . . .	49
4.2.2	Cholesky-Zerlegung . . . . .	50
4.2.3	”Lösung” nicht regulärer Systeme . . . . .	52

# Kapitel 1

## Grundlegende Konzepte der Numerik

Eine "Mathematische Aufgabe" besteht abstrakt aus der Auswertung einer Abbildung

$$\phi : X \rightarrow Y \quad \text{in einem } x \in X \quad \text{mit geeigneten Räumen } X, Y$$

Beispiele

- Berechnung eines Integrals:  $\int_a^b f(x)dx$ :

$$\phi_f((a, b), f) : X \times L^1 \rightarrow \mathbb{R}$$

- Lösung einer DGL

Objekte und Auswertungen können meist nur näherungsweise dargestellt werden, da z.B. nicht jede reelle Zahl auf dem Computer exakt dargestellt werden kann.

- Durch nicht exakte Darstellung entstehen Rundungsfehler
- Durch vereinfachte Beschreibung komplexer Vorgänge können auch Modellfehler entstehen
- Durch ungenaue Messungen können Datenfehler entstehen

Die Numerik befasst sich unter anderem mit folgenden Fragestellungen:

**Algorithmik:** Angabe von Algorithmen bzw. Berechnungsverfahren zur näherungsweisen Lösung von math. Aufgaben

**Konditionierung und Stabilität:** Einfluss von Störungen(Fehlern) auf das Ergebnis der math. Aufgabe oder Berechnung

**Konvergenz:** Abschätzung des Fehlers zwischen berechneter und exakter Lösung

**Komplexität:** Aufwand des numerischen Verfahrens

## 1.1 Zahlendarstellung und Rundungsfehler

Computer können Zahlen nur mit endlich vielen Ziffern darstellen, damit sind nicht alle reellen (komplexen) Zahlen exakt darstellbar. Manche Programme können Ganzzahlen mit beliebig vielen Stellen oder Gleitkommazahlen mit beliebig vielen Stellen darstellen (endlich viele, auch begrenzt durch Speicherplatz). Rechnungen damit werden dann jedoch sehr langsam. Meist ist die Anzahl der Stellen also begrenzt, weil nur eine gewisse Anzahl an Bits/Bytes für die Darstellung einer Zahl reserviert ist. 1 Byte = 8 Bits, kann Ganzzahlen zwischen 0 und 255, bzw zwischen -128 bis +127, darstellen.

$\underbrace{\pm}_{1\text{-Bit}} m * b^e, \quad b = 2 \text{ Basis } m = 1. \underbrace{m_1 \dots m_{52}}_{52\text{-Bits}}, \quad e = \underbrace{c}_{11\text{-Bits}} - 1023 \text{ (double-precision)}$

Menge aller Gleitkommazahlen =:  $A$  (endliche Menge)

$D := [x_{\min}, x_{\max}] \cup 0 \cup [x_{\text{posmin}}, x_{\max}]$  ist der "darstellbare Zahlenbereich"

Rundung bildet  $D$  auf  $A$  ab  $rd : D \rightarrow A$ , sodass  $|x - rd(x)| = \min_{y \in A} |x - y|$

Rundung zur nächstliegenden Zahl.

IEEE : bei gleichweit entfernten Gleitkommazahlen nehme die, wo  $m_{52} = 0$  ist.

Für eine Zahl  $x = \pm m * 2^e$  mit  $m \in [1, 2)$  ist der absolute Rundungsfehler:

$$|x - rd(x)| \leq \frac{1}{2} * 2^{-52} * 2^e$$

der relative Rundungsfehler

$$\frac{|x - rd(x)|}{|x|} \leq \frac{1}{2} * 2^{-52}$$

ist unabhängig von der Größe von  $x$ .

Mit der "Maschinengenauigkeit" einer Gleitkommadarstellung bezeichnet man den Abstand zwischen 1 und der nächst größeren Gleitkommazahl. bei doppelt genauer Darstellung :

$$\text{"Epsilon" "Eps", "eps"} := 2^{-52} \approx 2,22 * 10^{-16}$$

Es gilt immer :  $rd(x) = rd(x(1 + \varepsilon))$  für alle  $|\varepsilon| \leq \frac{\text{eps}}{2}$

Wichtig wird das Runden insbesondere auch bei den arithmetischen Operationen  $+, -, *, \div$

Diese werden in Computern durch Maschinenoperationen ersetzt ( $\oplus \ominus \otimes \oslash$ ) bei denen das Ergebnis wieder eine Maschinenzahl ist.

Für jede Operation  $*$   $\in \{+, -, \cdot, \div\}$  und  $y, x \in A$  gilt :

$$x \otimes y \in A, \quad x \otimes y = (x * y)(1 + \varepsilon) \text{ mit } |\varepsilon| \leq \frac{\text{eps}}{2}$$

Im allgemeinen gelten die typischen Geseze nicht, also:

- i)  $(x \otimes y) \oplus z$  ist nicht assoziativ
- ii)  $(x \otimes y) \otimes z$  ist nicht distributiv
- iii)  $x \oplus y = x$  falls  $|y| \leq \frac{|x|}{2} \text{eps}$

mit iii) kann man eps durch ausprobieren berechnen. (noch was von foto abschreiben)

## 1.2 Kondition und Stabilität

Einfluss von Störungen oder Fehlern auf das Ergebnis einer mathematischen Aufgabe oder eines Berechnungsverfahrens.

**Beispiel 1.2.1.** Kleine Unterschiede von Werten können evtl. auf dem Rechner/ in der gewählten Zahlendarstellung gar nicht unterschieden werden. (Berechnungsverfahren)

**Beispiel 1.2.2.** Mathematische Aufgabe: Beispiel für lineares Gleichungssystem:  $x : Ax = b$  mit  $A = \begin{pmatrix} 1,2969 & 0,8648 \\ 0,2161 & 0,1441 \end{pmatrix}$  für  $b = \begin{pmatrix} 0,8642 \\ 0,1220 \end{pmatrix}$  ist die Lösung  $x = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$ . Für  $b = \begin{pmatrix} 0,86419999 \\ 0,12200001 \end{pmatrix}$  ist die Lösung  $x = \begin{pmatrix} 0,9911 \\ -0,487 \end{pmatrix}$

**Definition 1.2.3.** Eine mathematische Aufgabe heißt "schlecht konditioniert" wenn kleine Änderungen in den Daten große relative Fehler verursachen. Andernfalls heißt die Aufgabe "gut konditioniert"

**Bemerkung 1.2.4.** Eine gute Konditionierung deiner math. Aufgabe ist notwendig, um das Problem numerisch sinnvoll lösen zu können, da Rundungsfehler sonst große Fehler verursachen können.

Sei  $\phi : X \rightarrow Y$  eine mathematische Aufgabe

**Definition 1.2.5.** Ein Verfahren oder Algorithmus zur (näherungsweisen) Lösung der math. Aufgabe  $\phi$  ist eine Abbildung  $\tilde{\phi} : X \rightarrow Y$ ,

die durch Hintereinanderschaltung endlich vieler (oder abzählbar unendlich vieler) elementarer, möglicherweise Rundungsfehlerbehafteter, Rechenoperation

$$\phi^{(k)}, \quad k = 1, 2, 3, \dots$$

definiert ist, also

$$\tilde{\phi} = \dots \circ \phi^{(3)} \circ \phi^{(2)} \circ \phi^{(2)} \circ \phi^{(1)}$$

**Bemerkung 1.2.6.** typischerweise gibt es verschiedene Algorithmen für die gleiche math. Aufgabe  $\phi$ . Von einem "guten" Algorithmus erwartet man, dass die im Verlauf des Algorithmus akkumulierten Fehler den durch die Kondition der math. Aufgabe unvermeidbaren Fehler nicht wesentlich übersteigen.

**Definition 1.2.7.** Ein Algorithmus  $\tilde{\phi}$  heißt "instabil", wenn es eine Störung  $\tilde{x}$  von  $x$  gibt so dass der durch den Rundungsfehler und Störungen verursachte relative Fehler erheblich größer ist als der nur durch die Störung verursachte Fehler, d.h. falls  $\phi(x) \neq 0$  und  $\frac{|\tilde{\phi}(\tilde{x}) - \phi(x)|}{|\phi(x)|} \gg \frac{|\phi(\tilde{x}) - \phi(x)|}{|\phi(x)|}$ . Der Algorithmus heißt stabil, falls er nicht instabil ist. (ggf. also "bei  $x$ " oder "für kleine  $|x|$ ", große  $|x|$ , o.ä.)

**Beispiel 1.2.8.** math. Aufgabe:

$$\phi(x) = \frac{1}{x(x+1)}, \quad x \in \mathbb{R} \setminus \{0, -1\}$$

Es gilt:

$$\frac{1}{x(x+1)} = \frac{1}{x} - \frac{1}{x+1}$$

Zwei mögliche Verfahren:

$$\tilde{\phi}_1(x) = \frac{1}{x(x+1)}$$

ist stabil für  $x \gg 1$

$$\tilde{\phi}_2(x) = \left( \frac{1}{x} \right) - \left( \frac{1}{(x+1)} \right)$$

ist instabil für  $x \gg 1$  wegen Auslöschung der Differenzbildung.

## 1.3 Landau-Symbole, Genauigkeit und Komplexität

**Beispiel 1.3.1.**

- (a)  $A \in \mathcal{M}_n(\mathbb{R})$ ,  $n$ -Vektor  $x \in \mathbb{R}^n$ ,  $Ax = b \in \mathbb{R}^n$ ,  $b_i = \sum_{j=1}^n A_{ij}x_j$   
 Rechnung von  $Ax$ :  $n^2$  Matrixmultiplikationen,  $n(n-1)$  Additionen nötig.  
 $\sim$  Rechenaufwand etwa quadratisch in der Dimension des Gleichungssystems. (bei voll besetzter Matrix)

(b) Genauigkeit der Differenzenquotienten zur Approximation der Ableitung:

$$\left| n'(x) \frac{n(x+h) - n(x)}{h} \right| \leq h \frac{1}{2} \max_{[x, x+h]} |n''|$$

Fehler gleich Größenordnung wie Abstand  $h$ .

**Definition 1.3.2.** Seien  $D \subset \mathbb{R}^n$ ,  $f, g : D \rightarrow \mathbb{R}$ , und  $x, x_0 \in D$   
Man sagt:

- i) Die Funktion  $f$  wächst für  $x \rightarrow x_0$  langsamer als  $g$ , geschrieben als:  
 $f = o(g)$  "f ist klein-o von g"
- ii) Die Funktion "f wächst für  $x \rightarrow x_0$  nicht wesentlich schneller als  $g$ ",  
geschrieben als  $f = \mathcal{O}(g)$ , "f ist groß-O von g" wenn  $\exists c > 0 \exists \varepsilon > 0 :$   
 $|f(x)| \leq c|g(x)| \forall x \in B_\varepsilon(x_0) |x - x_0| \leq \varepsilon$
- iii) analoge Definition für  $x \rightarrow \pm\infty$

**Bemerkung 1.3.3.** "Konditionierung schlecht" bzw. "Konditionierung instabil" ist nicht genau definiert.

Was einfacher ist: Aufgabenstellung bzw. Verfahren: Falls Fehlerverstärkung bei verfahren kleiner als bei anderen, dann ist das erste Verfahren "stabiler", bzw. eine aufgabe "besser konditioniert"

## 1.4 Differentielle Fehleranalyse:

Math. Aufgaben  $\phi : X \rightarrow Y$ . Ist  $\phi$  (unendlich) differenzierbar, dann kann die Kondition auch mit Hilfe der Ableitungen von  $\phi$  bestimmt/berechnet werden:  
Sei  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  eine Abbildung,  $x \in \mathbb{R}^m$ ,  $x = (x_1, \dots, x_m)^T$

$$\phi(x) = (\phi_1(x_1, \dots, x_m), \dots, \phi_m(x_1, \dots, x_m))^T$$

Die  $\phi_i$  seien alle zweimal stetig differenzierbar (partiell).  
Damit gilt dann:

$$\begin{aligned} \Delta y_i &= \phi_i(x + \Delta x) - \phi_i(x) \\ &= \sum_{j=1}^m \frac{\partial \phi_i(x)}{\partial x_j} \Delta x_j + R_i(x, \Delta x) \quad (\text{Taylor}) \\ &= \sum_{j=1}^m \frac{\partial \phi_i(x)}{\partial x_j} \Delta x_j + R_i(x, \Delta x) + \mathcal{O}(|\Delta x|^2) \end{aligned}$$

Dann folgt für den relativen Fehler:  $\frac{\Delta y}{|y|} = \frac{\Delta y}{|\phi(x)|}$

$$\frac{\Delta y}{y_i} = \sum_{j=1}^m \frac{\partial \phi_j(x)}{\partial x} \frac{x_j}{\phi_i(x)} \frac{\Delta x_j}{x_j} \quad \text{Für } x_j \neq 0, y_i \neq 0$$

$\frac{\Delta y}{y_i}$  Ist der relative Aufgabenfehler

$$\frac{\partial \phi_j(x)}{\partial x} =: K_{ij}(x)$$

$\frac{\Delta x_j}{x_j}$  ist der relative Datenfehler

**Definition 1.4.1.** Die  $K_{ij}(x), i = 1, \dots, n, j = 1, \dots, m$  heißen "relative Konditionszahlen" von  $\phi$  in  $x$  sie sind ein Maß dafür, wie sich kleine relative Fehler in den Eingangsdaten im ergebnis auswirken.

Die Aufgabe:  $y = \phi(x)$  aus  $x$  zu berechnen, ist schlecht konditioniert, wenn es ein  $i, j$  gibt mit  $|K_{ij}(x)| \gg 1$ . Ansonsten ist  $\phi$  gut konditioniert.

**Beispiel 1.4.2.** Grundoperation Addition:  $\phi(x_1, x_2) = x_1 + x_2$

$$K_1 = \frac{\partial \phi}{\partial x_1}(x) \frac{x_1}{\phi(x)} = 1 * \frac{x_1}{x_1 + x_2} = \frac{1}{1 + \frac{x_2}{x_1}}$$

$$K_2 = \frac{\partial \phi}{\partial x_2}(x) \frac{x_2}{\phi(x)} = 1 * \frac{x_2}{x_1 + x_2} = \frac{1}{1 + \frac{x_1}{x_2}}$$

Für  $\frac{x_1}{x_2} \approx -1$  werden die  $K_i$  sehr groß, dort ist die Addition schlecht konditioniert.

Das entspricht  $x_1 \approx -x_2$ , entspricht Subtraktion von 2 Zahlen, die fast gleich groß sind.

Bei Gleitkommazahlen: Übereinstimmung in den vorderen Mantissenstellen, dadurch Genauigkeit des Resultats geringer als der Daten.

**Definition 1.4.3.** Unter "Auslöschung" versteht man den Verlust an wesentlichen Dezimalstellen bei der Subtraktion von Zahlen gleichen Vorzeichens. Dies kann zu relativ großen Fehlern führen, falls eine oder beide Zahlen von operationen gerundet ( $\Delta x \neq 0$ ) werden.

**Bemerkung 1.4.4.** Diese differenzielle Fehleranalyse kann analog für einen komplexen Algorithmus  $\tilde{\phi} = \phi^{(n-1)} \circ \dots \circ \phi^{(1)}$ , bestehend aus einfachen Rechenoperationen  $\phi^{(i)}$ , durchgeführt werden, Kettenregel führt auf Ableitung der Hintereinanderschaltungen. Für komplexe Algorithmen aber nicht sehr sinnvoll durchzuführen. Man kann stattdessen versuchen statistische Methoden anzuwenden, in denen z.B. Rundungsfehler durch zufallsvariablen modelliert werden, um damit Wechselwirkungen abschätzen zu können.



**Beispiel 1.4.5** (Rekursive Berechnung von Integralen).

Aufgabe: Es sollen die folgenden Untegrale berechnet werden:

$$I_1 := \frac{1}{e} \int_0^1 x^n e^x dx, \quad n = 0, 1, 2, \dots$$

Berechnung mit integrationsformeln/numerische Integration: Später in Vorlesung.

mit partieller integration sieht man, dass

$$I_n = 1 - nI_{n-1}$$

eine Lösung ist. Für

$$n = 0 \Rightarrow I_0 = \frac{e-1}{e} \approx 0,632\dots$$

Numerische Berechnung:  $I_0 = 0,632\dots$

$$I_5 = 0,1455\dots$$

$$I_{10} = 0,0838\dots$$

$$I_{15} = 0,059\dots$$

$$I_{20} = -30, \dots$$

$$I_{21} = 635,04$$

$$I_{22} = -13970, \dots$$

Man sieht leicht:

$$I_n > 0, \quad I_n \leq \int_0^1 x^n dx = \frac{1}{n+1}$$

Warum diese Fehler?

In jedem Schritt der Rekursion wird der Fehler aus dem letzten schritt mit Faktor  $-n$  multipliziert. Nach  $n$  schritten mit gesamtfaktor  $(-n)^n * n!$

Die Fakultät wird schnell groß!

# Kapitel 2

## Interpolation

Aufgabe der Interpolation ist es, diskrete Datenwerte durch eine kontinuierliche Funktion darzustellen.

BILD!

Dabei sollen die Datenpunkte  $(x_i, y_i), i = 1, \dots, n$  exakt durch eine "interpolierende" Funktion  $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$  mit  $f(x_i) = y_i$  dargestellt werden. Voraussetzung:  $x_i$  paarweise verschieden!

Idee dahinter: Datenpunkte sind nur Punktauswertungen einer "glatten" Funktion, die durch  $f$  approximiert werden soll. Nach der interpolierenden Funktion  $f$  wird üblicherweise in einem "einfachen" Funktionenraum gesucht. Z.B. Polynome Trigonometrische Funktionen,..., evntuell nur stückweise definierte aber insgesamt glatte Funktionen.

Zusätzlich zu Funktionen  $f(x_i) = y_i$  können auch evntuell Ableitungen  $f'(x_i) = z_i$  vorgegeben sein.

Das ganze funktioniert ähnlich auch bei Daten und Funktionen über mehrdimensionalen Gebieten, z.B. Rekonstruktion von 2D- Flächen in 3D (Computergraik, CAD)

### 2.1 Polynominterpolation

Ein einfacher Ansatz: Interpolation durch Polynome. Ein Polynom vom Grad  $n$  ist hier eine Funktion

$$p : \mathbb{R} \rightarrow \mathbb{R}, p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

mit reellen Koeffizienten

$$a_0, \dots, a_n \in \mathbb{R}, a_n \neq 0 \rightsquigarrow \text{Grad } n$$

Es bildet

$$\mathbb{P}_n := \{p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \mid a_i \in \mathbb{R}, i = 0, \dots, n\}$$

Die Menge der Polynome vom Grad  $\leq n$

$\mathbb{P}_n$  bildet einen  $\mathbb{R}$  Vektorraum,

$$(p+q)(x) := p(x) + q(x), (\alpha p)(x) := \alpha p(x) \quad \forall \alpha \in \mathbb{R}, p, q \in \mathbb{P}_n$$

Die "Monome"  $\{1, x, x^2, \dots, x^n\}$  bilden eine Basis von  $\mathbb{P}_n$ , mit  $\dim(\mathbb{P}_n) = n+1$ .

**Definition 2.1.1.** Die Aufgabe der Polynominterpolation besteht darin, zu  $n+1$  paarweise verschiedenen Punkten  $x_i, i = 0, \dots, n$  ("Stützstellen", "Knoten") und gegebenen Knotenwerten  $y_i, i = 0, \dots, n$  ein Polynom  $p \in \mathbb{P}$  zu bestimmen, mit der Eigenschaft:

$$p(x_i) = y_i, \quad i = 0, \dots, n$$

**Satz 2.1.2.** Die Aufgabe der Polynominterpolation ist eindeutig lösbar, d.h. es gibt genau ein  $p \in \mathbb{P}_n$ , das die Bedingung erfüllt.

**Beweis**

(a) Eindeutigkeit der Lösung:

angenommen  $p \in \mathbb{P}_n$  und  $q \in \mathbb{P}_n$  seien zwei Lösungen,  $p(x_i) = y_i = q(x_i)$ .

Für die Differenz  $p - q \in \mathbb{P}_n$  gilt dann:

$p - q$  hat  $n + 1$  Nullstellen in den  $x_i, i = 0, \dots, n$ , aber ein  $\tilde{p} \in \mathbb{P}_n$  kann höchstens  $n$  verschiedene Nullstellen haben, oder es gilt  $\tilde{p} \equiv 0$  (z.B. über Satz von Rolle). Also ist  $p \equiv q$ , es gibt demnach höchstens eine Lösung in  $\mathbb{P}_n$

(b) Existenz einer Lösung:

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n, \quad p(x_i) = y_i, \quad i = 0, \dots, n$$

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Bedingungen führen auf lineares Gleichungssystem und Matrix  $V_n$  (Vandermonde-Matrix)

Man kann zeigen:

$$\det(V_n) = \prod_{i=0}^n \prod_{j=i+1}^n (x_j - x_i) \neq 0$$

falls alle  $x_i$  paarweise verschieden sind. Also ist das lineare Gleichungssystem eindeutig lösbar, wenn  $\det(V_n) \neq 0$  bzw. wenn die Stützstellen paarweise verschieden sind. Zu beliebiger rechter Seite  $(y_0, \dots, y_n)$  gibt es also Koeffizienten  $(a_0, \dots, a_n)$  für ein interpolation Polynom.

□

Das Lineare Gleichungssystem liefert im Prinzip auch eine Berechnungsmethode, ist aber schlecht konditioniert, und die Lösung ist relativ aufwändig.

### 2.1.1 Lagrange-Interpolation

#### Idee

Wähle Basispolynome für  $\mathbb{P}_n$  angepasst an die Stützstellen  $x_i$ , so dass das Interpolationspolynom damit aus den Werten  $y_i$  leicht bestimmt werden kann. Man kann recht einfach Polynome  $L_i^{(n)} \in \mathbb{P}_n$  konstruieren, die in genau einem Stützpunkt  $x_i = 1$  sind und in allen anderen Stützpunkten  $= 0$ .

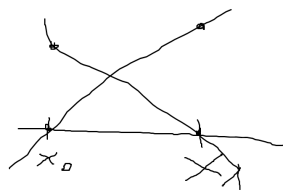
$$L_i^{(n)}(x_j) = \begin{pmatrix} 1 & i = j \\ 0 & \text{sonst} \end{pmatrix} = \delta_{ij}$$

$n$  Nullstellen

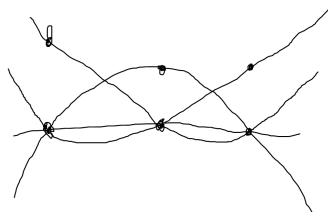
$$x_j, j \neq i \implies L_i^{(n)} = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

Diese  $L_i^{(n)}, i = 0, \dots, n$  heißen "Lagrange-Basispolynome" zu Stützstellen  $x_0, \dots, x_n$ . Linear unabhängig: leicht zu sehen: Alle  $L_j^{(n)}(x_i) = 0$ . Anzahl ist gleich  $\dim \mathbb{P}_n$

**Beispiel 2.1.3.**  $n = 1$   $x_0, x_1$



$n = 2$



**Bemerkung 2.1.4.** auch in höheren Dimensionen,  $\leadsto$  Numerik partieller Differentialgleichungen "Finite Elemente Methode"

**Definition 2.1.5.** Das Polynom  $p(x) := \sum_{i=0}^n y_i \cdot L_i^{(n)}(x)$  heißt Lagrange-Interpolationpolynom zu den Stützstellen  $x_0, \dots, x_n$  und Daten  $y_0, \dots, y_n$ .

**Vorteil:**

Für festes  $n \in \mathbb{N}$  und Stützstellen  $(x_0, \dots, x_n)$  relativ leicht zu berechnen, natürliche Darstellung des Interpolationpolynoms mit einfachen Koeffizienten

**Nachteil:**

- Für viele bzw eng beieinander liegende Stützstellen ist die Formel für  $L_i^{(n)}$  relativ schlecht konditioniert (Auslöschungseffekte)
- Will man weitere Stützstellen und Daten dazunehmen, muss komplett neu gerechnet werden

## 2.1.2 Newton Darstellung

Alternative Darstellung, auch an die Stützstellen angepasst. Eine Polynombasis, die auch zu den Stützstellen passt, aber leicht erweiterbar ist: "Newton-Basis"

$$N_0(x) := 1 \text{ konstant}$$

$$N_i(x) := \prod_{j=0}^{i-1} (x - x_j), \quad i = 1, \dots, n$$

$\text{Grad}(N_i) = i \implies (N_i)$  linear unabhängig,  $\text{span} \{N_0, \dots, N_i\} = \mathbb{P}_i$ , (Basis von  $\mathbb{P}_i$ )

Damit kann auch das Interpolationspolynom  $p$ ,  $p(x_i) = y_i$ ,  $i = 0, \dots, n$  in dieser Basis dargestellt werden,  $p = \sum_{i=0}^n a_i N_i$   
Koeffizienten  $a_i$

$$\begin{aligned} a_0 &= y_0 \\ y_1 &= p(x_1) = a_0 + a_1(x_1 - x_0) + 0 \implies a_1 = \frac{y_1 - a_0}{x_1 - x_0} \\ y_n &= p(x_n) = a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) + \dots + a_n(x_n - x_0) \dots (x_n - x_{n-1}) \\ \implies a_n &= \frac{y_n - \sum_{i=0}^{n-1} a_i \prod_{j<i} x_n - x_j}{\prod_{j=0}^{n-1} x_n - x_j} \end{aligned}$$

**Vorteil:**

- Man kann beliebig zusätzliche Stützstellen dazunehmen, ohne bisherige Berechnung zu verwerfen
- Reihenfolge/Ordnung der Stützstellen beliebig

Einfacher Algorithmus zur Berechnung der Koeffizienten:

**Satz 2.1.6** (Newton Darstellung mit dividierten Differenzen). Das Interpolationspolynom zu den Punkten  $(x_i, y_i)$ ,  $i = 0, \dots, n$  lässt sich bzgl. der Newton-Basis darstellen als

$$p(x) = \sum_{i=0}^n y[x_0, \dots, x_i] N_i(x)$$

Dabei bezeichnen  $y[x_0, \dots, x_i]$  die zu  $(x_j, y_j)$  gehörenden "dividierte Differenzen", rekursiv definiert als

$$y[x_i] := y_i, i = 0, \dots, n$$

$$y[x_i, x_{i+1}, \dots, x_{i+k}] := \frac{y[x_{i+1}, \dots, x_{i+k}] - y[x_i, \dots, x_{i+k-1}]}{x_{i+1} - x_i}, \quad i = 0, \dots, n, \quad k = 1, \dots, n-i$$

**Beweis**

Zu  $i, n$  sei  $P_{i,i+n}$  das Polynom, das  $(x_i, y_i) \dots (x_{i+n}, y_{i+n})$  interpoliert.  $\implies p = P_{0,n}$  ist gesucht.

*Behauptung:*  $P_{i,i+k}(x) = y[x_i] + y[x_i, x_{i+1}](x - x_i) + \dots + y[x_i, x_{i+k}](x - x_i) \dots (x - x_{i+k-1})$

Per Induktion über  $k$ :  $k = 0$ :  $P_{i,i}(x) = y_i = y[x_i]$

angenommen, es gilt für  $k - 1$ .

Es ist

$$P_{i,i+k} = P_{i,i+k-1} + a(x - x_i) \dots (x - x_{i+k-1})$$

mit  $a \in \mathbb{R}$

z.z.:

$$a = y[x_i, \dots, x_{i+k}]$$

.  $a$  ist der Koeffizient von  $x^k$  in  $P_{i,i+k}$

Nach Induktionsannahme gilt:

$$P_{i,i+k-1} = \dots + y[x_i, \dots, x_{i+k-1}] \cdot x^{k-1}$$

und

$$P_{i+1,i+k} = \dots + y[x_{i+1}, \dots, x_{i+k}] \cdot x^{k-1}$$

**Bild**

$$P_{i,i+k} = \frac{(x - x_{i+k}) \cdot P_{i,i+k-1} - (x - x_i) P_{i+1,i+k}}{x_i - x_{i+k}}$$

Der Koeffizient der höchsten Potenz  $x^k$  in  $P_{i,i+k}$  ist gerade

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

□

**Bemerkung 2.1.7.** Das Polynom  $P_{i,i+k}$  und die  $y[x_i, \dots, x_{i+k}]$  sind unabhängig von der Reihenfolge der Punkte, also invariant gegenüber Permutation.

**Frage** (Berechnung der dividierten Differenzen?).

$x_0$   $y[x_0]$   
 $x_1$   $y[x_1]$   $y[x_0, x_1]$   
 $x_2$   $y[x_2]$   $y[x_1, x_2]$   $y[x_0, x_1, x_2]$   
 $\vdots$   $\vdots$   $\vdots$   $\vdots$   
 $x_k$   $y[x_k]$   $\dots$   $y[x_0, x_1, \dots, x_k]$

Algorithmus: berechne  $P_{i,j}$ :

$$\begin{aligned} &\text{Für } i = 0, \dots, n : P_{i,0} := y_i \\ &\text{Für } K := 1, \dots, n, \ i = 0, \dots, n - k : P_{i,k} := \frac{P_{i+1,k-1} - P_{i,k-1}}{x_{i+k} - x_i} \end{aligned}$$

### 2.1.3 Auswertung von Polynomen

Gegenüber der Auswertung von  $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$

$$\#multiplikationen : n + 1 + 2 + \dots + (n - 1) = \frac{n(n+1)}{2}$$

Ist die Auswertung der alternativen Darstellung

$$p(x) = a_0 + x(a_1 + x(a_2 + x(\dots(a_n - 1 + xa_n)\dots)))$$

nach dem "Horner-Schema"

$$\begin{cases} b_n := a_n, \\ b_k := a_k + xb_{k+1}, \end{cases} \quad \begin{aligned} &\Rightarrow p(x) = b_0 \\ &k = n - 1, \dots, 0 \end{aligned} \quad \#multiplikationen : n$$

1. deutlich effizienter
2. oft auch stabiler/besser konditioniert als die Berechnung aller Potenzen

Für die Newton Darstellung:

$$N_i = \prod_{j=0}^{i-1} (x - x_j) = N_{i-1} \cdot (x - x_{i-1})$$

damit gilt für ein Polynom P die alternative Darstellung

$$\begin{aligned} p(x) &= \sum_{i=0}^n y[x_0, \dots, x_i] N_i(x) \\ &= y[x_0] + (x - x_0)(y[x_0, x_1] + (x - x_1)(y[x_0, x_1, x_2] + (x - x_2)(\dots + (x - x_{n-1})(y[x_0, \dots, x_n]))) \end{aligned}$$

und ein entsprechendes verallgemeinertes Horner-Schema:

$$\begin{cases} b_n &:= y[x_0, \dots, x_n] \\ b_k &:= y[x_0, \dots, x_n] + (x - x_k) \cdot b_{k+1}, \end{cases} \quad k = n - 1, \dots, 0$$

Also  $P(x) = b_0$ . Dabei ist die Anzahl der Multiplikationen =  $n$ , somit ist der Aufwand deutlich geringer. Sind die Stützstellen aufsteigend nach dem Abstand zu  $x$  sortiert, dann ist das Horner-Schema relativ gut konditioniert.



### 2.1.4 Interpolationsfehler bei der Interpolation einer gegebenen Funktion

$y_i$ ,  $i = 0, \dots, n$  nicht (willkürlich) vorgegeben, sondern Funktionswerte einer Funktion  $f : I \rightarrow \mathbb{R}$ , mit  $I \subset \mathbb{R}$ , alle  $x_i \in I$ ,  $i = 0, \dots, n$  also  $y_i = f(x_i)$ .

**Frage.** Wie groß ist der Unterschied zwischen  $p$  und  $f$ ?

Der Unterschied kann mit einem ähnlichen Ausdruck wie Taylor-Restglied abgeschätzt werden:

**Satz 2.1.8.** Sei

$$\left[ \min_{i=0, \dots, n} x_i, \max_{i=0, \dots, n} x_i \right] \subseteq I \text{ und } f \in C^{n+1}(I)$$

Dann gibt es zu jedem  $x \in I$  ein  $\xi_x \in \left[ \min_i(x_i, x), \max_i(x_i, x) \right]$

mit  $\tilde{I} := \left[ \min_i(x_i, x), \max_i(x_i, x) \right]$  sodass:

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{j=0}^n (x - x_j)$$

**Beweis** Für  $x = x_i$ ,  $i \in \{0, \dots, n\}$  ist nichts zu zeigen, da dann

$$f(x) = p(x), \quad \prod (x - x_j) = 0$$

Wir machen einen Ansatz:

$$g(t) := \prod_{j=0}^n (t - x_j)$$

und

$$c(x) := \frac{f(x) - p(x)}{g(x)} \quad (\text{eine Konstante abh. von } x \text{ bzgl. } t)$$

Setze

$$F(t) := f(t) - p(t) - c(x) \cdot g(t)$$

Es ist  $F(x) = 0$ , und

$$F(x_i) = \underbrace{f(x_i) - p(x_i)}_{=0} - c(x) \cdot \underbrace{g(x_i)}_{=0}$$

$$\begin{aligned}
&\implies F \text{ hat mindestens } n+2 \text{ Nullstellen in } \tilde{I} \\
&\implies F' \text{ hat mindestens } n+1 \text{ Nullstellen in } \tilde{I} \\
&\implies F'' \text{ hat mindestens } n \text{ Nullstellen in } \tilde{I} \\
&\vdots \\
&\implies F^{(n+1)} \text{ hat mindestens 1 Nullstelle in } \tilde{I} =: \xi_x
\end{aligned}$$

und es ist

$$\begin{aligned}
0 = F^{(n+1)}(\xi_x) &= f^{(n+1)}(\xi_x) - \underbrace{P^{(n+1)}(\xi_x)}_{=0, \text{ da } p \in \mathbb{P}_n} - c(x) \cdot \underbrace{g^{(n+1)}(\xi_x)}_{=(n+1)!} \\
\frac{f(x) - p(x)}{g(x)} \cdot (n+1)! &= f^{(n+1)}(\xi_x) \implies f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \underbrace{g(x)}_{=\prod (x-x_j)}
\end{aligned}$$

□

**Korollar 2.1.9.** Ist  $f \in C^\infty(I)$  und es gebe ein  $M < \infty$  so, dass

$$|f^{(n)}(x)| \leq M \quad \forall n \in \mathbb{N} \text{ und } x \in I$$

dann konvergiert die Folge der Interpolationspolynome  $p_n \in \mathbb{P}_n$  zu  $f$  mit beliebigen disjunkten Stützpunkten  $x_0, \dots, x_n \in I$  auf  $I$  gleichmäßig gegen  $f$

**Beweis**  $\forall x \in [a, b] : |f(x) - p(x)| \leq \frac{1}{(n+1)!} \cdot M \cdot (b-a)^{n+1} \rightarrow 0$  für  $n \rightarrow \infty$  □

**Bemerkung 2.1.10.** Dies gilt leider nicht für beliebige (auch beliebig glatte) Funktionen

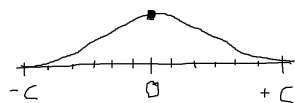
**Bemerkung 2.1.11.** Satz von Weierstraß sagt aus, dass wir jede stetige Funktion  $f \in C^0(I)$  beliebig gut gleichmäßig durch Polynome approximieren können. Dies gilt leider nicht für die (Lagrange)-Interpolationspolynome

**Beispiel 2.1.12** (Das Runge-Beispiel).

$$f(x) = \frac{1}{1-x^2} \in C^\infty(\mathbb{R}).$$

Interpolation auf  $[-c, +c]$  mit äquidistanten Stützstellen

$$x_i = -c + \frac{2c}{n}i, \quad i = 0, \dots, n$$



Man kann zeigen: ist

$$c \leq \frac{e}{2} \implies \|f - p_n\|_\infty \rightarrow 0 \text{ für } n \rightarrow \infty$$

$$c > \frac{e}{2} \implies \|f - p_n\|_\infty \rightarrow \infty \text{ für } n \rightarrow \infty$$

warum?

$$\|f - p\|_\infty := \max_{x \in [-c, +c]} |f(x) - p(x)|$$

$$|f^{(n)}(x)| \sim 2^n \cdot n! \cdot \mathcal{O}(|x|^{-2-n})$$

**Beispiel 2.1.13.**

$$f(x) = |x|$$

$f$  ist nur in  $C^0([-1, +1])$ . Äquidistante Stützstellen:

$$x_i = -1 + \frac{2}{n} \cdot i, \quad i = 0, \dots, n$$

Man kann zeigen:

$$x \neq x_i, \text{ z.B. } x \text{ irrational, dann } \lim_{n \rightarrow \infty} P(x) \neq f(x)$$

**Verhalten gegenüber Störungen in den Daten?**

**Beispiel 2.1.14.**

$$I = [-1, 1], \quad x_i = -1 + \frac{2}{n} \cdot i, \quad i = 0, \dots, n, \quad n \text{ gerade } (x_{n/2} = 0)$$

Wir betrachten Daten

$$y_i = \begin{cases} \varepsilon & i = \frac{n}{2} \\ 0 & \text{sonst} \end{cases}$$



Interpolierende ist

$$p(x) = \varepsilon \cdot L_{n/2}^{(n)}(x) = \varepsilon \cdot \frac{\prod_{j \neq \frac{n}{2}} (x - x_j)}{\prod (0 - x_i)}$$

Die Faktoren im Produkt sind teilweise  $> 1$ ,

**Frage.** Kann man bei der Interpolation einer Funktion dem Interpolationsfehler  $f - p$  durch geschickte Wahl der Stützstellen kleiner machen?

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{j=0}^n (x - x_j)$$

$$W(x) := \prod_{j=0}^n (x - x_j)$$

Verändern kann man nur den Term

$$W(x) = \prod_{j=0}^n (x - x_j).$$

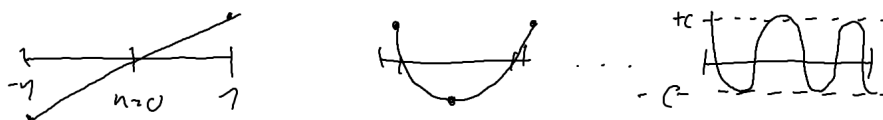
Man kann zeigen, es gilt spezielle paarweise verschiedene Stützstellen, die  $|W(x)|$  gleichmäßig minimieren.

**Satz 2.1.15.**

$$\min_{x_0, \dots, x_n \in I} \max_{x \in I} \left| \prod_{j=0}^n (x - x_j) \right| = \max_{x \in I} \left| \prod_{j=0}^n (x - t_j) \right|$$

mit  $x_0, \dots, x_n$  paarweise verschieden und mit den  $t_j \in [-1, +1]$  gerade die Nullstellen des "Tschebyscheff- Polynoms"  $T_{n+1}$

**Bemerkung 2.1.16** (Optimales  $w \in \mathbb{P}_{n+1}$ ?).



Diese Tschebyscheff-Polynome sind für  $k \in \mathbb{N}_0$  definiert als

$$T_k(x) := \cos(k \cdot \arccos x)$$

mit Nullstellen

$$t_j = \cos\left(\frac{2j+1}{2k} \cdot \pi\right), \quad j = 0, \dots, k-1$$

Es ist nicht direkt klar, dass  $T_k$  überhaupt ein Polynom ist (außer für  $k = 0, 1$ ):

$$\begin{cases} T_0 &= \cos(0 \cdot \arccos(x)) = \cos(0) = 1 \\ T_1 &= \cos(1 \cdot \arccos(x)) = x \end{cases}$$

Es gibt folgende 3-Term-Rekursion:

$$T_0(x) \equiv 1, \quad T_1(x) = x, \quad T_k(x) = 2x \cdot T_{k-1}(x) - T_{k-2}(x), \quad k \geq 2$$

Man sieht für das Runge-Beispiel:

Auch bei Verwendung der Tschebyscheff-Knoten können für relativ großes  $n$ , große Fehler am Rand des Intervalles auftreten. Abhilfe gelingt bei diesem Beispiel die Verwendung der "Tschebyscheff-Knoten zweiter Art", das sind gerade die Extremstellen des Tschebyscheff-Polynoms:

$$\tilde{t}_j = \cos\left(\frac{j}{n} \cdot \pi\right), \quad j = 0, \dots, n$$

**Bemerkung 2.1.17** (Kondition/Verhalten bei Störungen:). Sowohl bei Tschebyscheff-Knoten erster oder zweiter Art bleibt die Auswirkung einer Störung bei  $x = 0$  auf dem gesamten Intervall  $[-1, 1]$  beschränkt, im Gegensatz zu äquidistanten Knoten.

### 2.1.5 Hermite-Interpolation

Nicht nur Funktionswerte vorgegeben, sondern ggf. auch Ableitungen, evtl. auch höhere.

**Definition 2.1.18.** Die Hermite-Interpolationsaufgabe:

Gegeben:

Stützstellen  $x_i$ ,  $i = 0, \dots, m$  paarweise verschieden.

Werte  $y_i^{(k)}$ ,  $i = 0, \dots, m$   $k = 0, \dots, \mu_i \geq 0$

Gesucht: Polynom  $p \in \mathbb{P}_n$ ,  $n \sum_{i=0}^m \mu_i$  mit  $p^{(k)}(x_i) = y_i^{(k)}$ ,  $i = 0, \dots, m$   $k = 0, \dots, \mu_i$

Die  $x_i$  werden manchmal auch als  $\mu_i$ -fache Stützstellen bezeichnet.

**Satz 2.1.19.** Die Hermite Interpolationsaufgabe ist eindeutig lösbar

**Beweis** analog zur Lagrange-Interpolation, ähnliche (nicht im Sinne der Äquivalenzrelation) Matrix zur Vandermonde Matrix

$$\begin{aligned} p(x) &= a_0 + a_1x + \dots + a_nx^n \\ p'(x) &= a_1 + 2a_2x + 3a_3x^2 + \dots + na_nx^{n-1} \\ &\vdots \end{aligned}$$

sind z.B.  $p(x_0) = b_0$ ,  $p'(x_0) = c_0$ , so gilt für die Matrix

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 0 & 1 & 2x_0 & 3x_0^2 & \dots & nx_0^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ c_0 \\ \vdots \\ \vdots \end{pmatrix}$$

□

**Bemerkung 2.1.20.** Eine ähnliche Aufgabe bei der in Stützstellen eventuell nur einige Ableitungen vorgegeben sind, z.B.

$$p \in \mathbb{P}_2 : p(x_0) = y_0, p''(x_1) = y_1^{(2)}, p''(x_2) = y_2^{(2)}$$

ist im allgemeinen nicht oder nicht eindeutig lösbar

Ableitung ist Grenzwert der Differenzenquotienten

$$\frac{f(x+h) - f(x)}{h} \quad \text{für } h \rightarrow 0$$

Ähnlich kann man den Grenzwert der dividierten Differenzen anschauen:

Angenommen, die gegebenen Werte  $y_i$  sind Werte einer differenzierbaren Funktion  $f(x_i) = y_i$ , dann ist die erste dividierte Differenz gerade

$$f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}$$

ist gerade der Differenzenquotient zu  $f_1$

Für  $x_j \rightarrow x_i$  konvergiert dann  $f[x_j, x_i] \rightarrow f'(x_i) =: f[x_i, x_i]$

So können Hermite Stützstellen mit vorgegebenen Ableitungen als mehrfache Stützstellen aufgefasst werden:

Damit können wir das Hermite-Interpolationsverfahren wieder in Newton-Darstellung schreiben, wenn man die dividierten Differenzen verallgemeinert:

- Stützstellen mit (höheren) Ableitungen, also  $\mu_i > 0$  werden entsprechend dupliziert, statt  $x_i$  wird Folge

$$\underbrace{x_i, \dots, x_i}_{\mu_i+1\text{-mal}} \quad i = 0, \dots, m$$

eingefügt, damit bekommt man eine Folge von Stützstellen  $\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_n$ .

Für diese werden nun die modifizierten dividierten Differenzen definiert durch

$$y[\tilde{x}_i] := y_i^{(0)}, \quad y[\tilde{x}_1, \dots, \tilde{x}_{i+k}] := \begin{cases} y_i^{(k)} \cdot \frac{1}{k!} & \tilde{x}_i = \tilde{x}_{i+k} \\ \frac{y[\tilde{x}_{i+1}, \dots, \tilde{x}_{i+k}] - y[\tilde{x}_i, \dots, \tilde{x}_{i+k-1}]}{\tilde{x}_{i+k} - \tilde{x}_i} & \tilde{x}_i \neq \tilde{x}_{i+k} \end{cases}$$

$i$  ist der Original-Index zu  $\tilde{x}_i = x_i$

**Satz 2.1.21.** Damit hat das Hermite-Interpolationspolynom die Darstellung

$$p = \sum_{i=0}^n y[\tilde{x}_0, \dots, \tilde{x}_i] \prod_{j=0}^{i-1} (x - \tilde{x}_j)$$

A8: Stückweise Hermite-Interpolation: Werte  $y_i^{(0)}, y_i^{(1)}$  auf  $I_i = [x_{i-1}, x_i]$ ,  $i = 1, \dots, m$ :

**Beispiel 2.1.22.** Gesucht ist ein  $p$  mit  $p \in \mathbb{P}_4$ ,  $\begin{cases} p(0) = -1, & p'(0) = -2 \\ p(1) = 0, & p'(1) = 10, & p''(1) = 40 \end{cases}$

$m = 1, \mu_0 = 1, \mu_1 = 2$

Direkte Differenzen dazu?

$$\begin{array}{l|l} \tilde{x}_0 = 0 & -1 \quad -2 \quad 3 \quad 6 \quad 5 \\ \tilde{x}_1 = 0 & -1 \quad 1 \quad 6 \quad 11 \\ \tilde{x}_2 = 1 & 0 \quad 10 \quad 20 \\ \tilde{x}_3 = 1 & 0 \quad 10 \quad 20 \\ \tilde{x}_4 = 1 & 0 \end{array}$$

$$\begin{aligned} \Rightarrow p(x) &= -1 - 2 \cdot (x-0) + 3 \cdot (x-0)(x-0) + 6 \cdot (x-0)(x-0)(x-1) + 5 \cdot (x-0)(x-0)(x-1)(x-1) \\ &= -1 - 2x + 3x^2 + 6x^2(x-1) + 5x^2(x-1)^2 \end{aligned}$$

Ähnlich wie beim Satz über den Fehler der Lagrange-Polynominterpolation kann man zeigen

**Satz 2.1.23.** Ist

$$I := \left\{ \min_i(x_i), \max_i(x_i) \right\} \quad \text{und} \quad f \in C^{(n+1)}(I)$$

Dann gibt es zu jedem  $x \in I$  ein  $\xi \in I$  so, dass für das Hermite-Interpolationspolynom  $p$  zu  $f$  gilt

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \cdot \prod_{i=0}^m (x - x_i)^{\mu_i+1}$$

Mit der Vorgabe der Ableitungen kann man typischerweise erreichen, dass die Oszillationen des interpolierenden Polynoms zwischen den Stützstellen kleiner werden. Eine andere Möglichkeit/Ansatz dafür:

### 2.1.6 Spline Interpolation

Bei Polynom-Interpolation erhöht sich der Grad des Polynoms mit Anzahl der Stützstellen/Vorgaben, dies kann dann mit höheren  $x$ -Potenzen zu Oszillationen führen, die nicht gewünscht sind. Ein anderer Ansatz: Verwende nicht global (auf  $\mathbb{R}$  bzw.  $I$ ) definierte Polynome, sondern nur stückweise auf jedem Intervall (z.B.  $[x_{i-1}, x_i]$ , wenn sortiert) und verwende zusätzliche Übergangsbedingungen: Für  $x_0 < x_1 < \dots < x_m = b$  und  $I_i := [x_{i-1}, x_i]$ ,  $i = 1, \dots, n$  ist der entsprechende Funktionsraum dann:

$$S_{\text{li}}^{(k,r)}[a, b] := \left\{ s \in C^{(r)}[a, b] : s|_{I_i} \in \mathbb{P}_k, i = 1, \dots, n \right\}$$

( $r$ -mal stetig diffbar, lokale Polynome vom Grad  $\leq k$ )

**Beispiel 2.1.24.** Stückweise lineare Interpolation:

$$S_{\text{li}}^{(1,0)}[a, b] = \left\{ s \in C^{(0)}[a, b] : s|_{I_i} \in \mathbb{P}_1 \right\}$$

Sind Werte  $y_i$  an den Stützstellen  $x_i$  vorgegeben, dann ist durch die Vorgabe der Werte in den Endpunkten jedes Teilintervalls  $I_i$  genau ein Polynom  $p_i \in \mathbb{P}_1$  festgelegt. Stetigkeit über die Intervallgrenzen ergibt sich dadurch, dass sowohl  $p_i(x_i)$  und  $p_{i+1}(x_i)$  den gleichen Wert  $y_i$  haben. Der Graph der Funktion  $s$  ist gerade der Polygonzug mit Eckpunkten  $(x_i, y_i)$ ,  $i = 0, \dots, n$ . Auf  $I_i$  ist  $s|_{I_i}$  gerade das Interpolationspolynom zu  $(x_{i-1}, y_{i-1}), (x_i, y_i)$ . Also haben wir die Fehlerabschätzung für Interpolation einer Funktion  $f \in C^2[x_{i-1}, x_i]$ :

$$f(x) - p(x) = \frac{1}{2} f''(\xi_x) \cdot \prod_{j=0}^1 \underbrace{(x - x_{i-1+j})}_{\leq h_i}$$

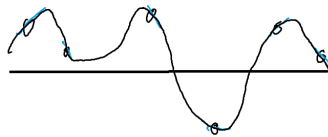
mit  $h := \max_{i=1, \dots, n} |h_i|$  folgt dann

**Korollar 2.1.25.** Ist  $f \in C^2[a, b]$  und  $s$  der interpolierende, stückweise lineare Spline, dann gilt:

$$\max_{x \in [a, b]} |f(x) - s(x)| \leq \frac{1}{2} \cdot \max_{x \in [a, b]} |f''(x)| \cdot h^2$$

**Beispiel 2.1.26.**

$$s \in S_h^{(3,1)}[a, b] : s(x_i) = y_i^{(0)}, s'(x_i) = y_i^{(1)}, i = 0, \dots, n$$





$s$  gegeben durch die stückweise Hermite Interpolation auf jedem der Teilintervalle. Fehlerabschätzung  $|f(x) - s(x)| \leq \dots$  siehe Übung A8, für  $f \in C^4[a, b]$ .

”Kubische Splines” : Im gegensatz zu den Hermite-Splines, sollen hier nur die Werte  $s(x_i) = y_i$ ,  $i = 0, \dots, n$  vorgegeben werden, dafür soll der Spline insgesamt glatter sein:

$$s \in S_h^{(3,2)}[a, b] = \{s \in C^2[a, b] : s|_{I_i} \in \mathbb{P}_3, i = 1, \dots, n\}$$

**Frage** (Kann das überhaupt Funktionieren?).

$S|_{I_i} \in \mathbb{P}_3$ ,  $s'|_{I_i} \in \mathbb{P}_2$ ,  $s''|_{I_i} \in \mathbb{P}_1$ ,  $s'''|_{I_i} \in \mathbb{P}_0$ , konstant. Es gilt

$$s \in C^2[a, b] \implies s'' \text{ ist Polygonzug auf } [a, b]$$

Wieviel Bedingungen ergeben sich? Wieviel Freiheitsgrade gibt es?

$n$ Teilintervalle $I_i$ , $s _{I_i} = P_i \in \mathbb{P}_3 \rightarrow$	$4 \cdot n$ Koeffizienten ”frei”
$s(x_i^-) = y_i$ , $i = 0 \dots, n$	$n + 1$ Bedingungen
$s(x_i^-) = s(x_i^+)$ , $i = 1 \dots, n - 1$	$n - 1$
$s'(x_i^-) = s'(x_i^+)$ ,	$n - 1$
$s''(x_i^-) = s''(x_i^+)$ ,	$n - 1$ $4n - 2$ Bedingungen

Wir haben also weniger Bedingungen als Koeffizienten, die Bedingungen sollten demnach zu erfüllen sein. Um die Eindeutigkeit zu bekommen sind eventuell noch 2 Bedingungen zusätzlich zu stellen, z.B.:

Steigung in $a, b$ :	$s'(x_0), s'(x_n)$
Krümmung in $a, b$ :	$s''(x_0), s''(x_n)$
Periodizität :	$s(x_0) = s(x_n), s'(x_0) = s'(x_n),$ $s''(x_0) = s''(x_n), y_0 = y_n$

Das Wort ”Spline” bezeichnet (englisch) eine dünne, biegsame Latte, z.B. Konstruktion der Form eines Schiffsrumpfs. Latte versucht (unter der Vorgabe der festen Punkte) ihre elastische Energie zu minimieren. Das entspricht der Minimierung der Gesamtkrümmung. Krümmung eines Graphen  $(x, f(x))$ :

$$\kappa(x, f(x)) = \frac{f''(x)}{\sqrt{1 + |f'(x)|^2}^{\frac{3}{2}}}$$

Für die Gesamtenergie gilt

$$E(f) := \int_a^b |\kappa(x, f(x))|^2 \cdot \sqrt{1 + f'^2} dx \approx \int_a^b |f''(x)|^2 dx \text{ (nach Linearisierung)}$$

Versucht man, unter allen glatten Funktionen, die in den  $x_i$  interpolieren, diese Energie zu minimieren, bekommt man gerade das kubische Spline

**Satz 2.1.27.**  $a = x_0 < x_1 < \dots < x_n = b, y_i \in \mathbb{R}, i = 0, \dots, n$   
 $A := \{f \in C^2[a, b] : f(x_i) = y_i, i = 0, \dots, n\}$  Dann gibt es genau eine Lösung  $s \in A$  mit

$$E(s) \leq E(f) \quad \forall f \in A$$

mit

$$E(f) := \int_a^b (f''(x))^2 dx$$

und dieses  $s \in S_h^{(3,2)}[a, b]$  mit  $s''(a) = s''(b) = 0$ ,  $s$  ist ein "natürlicher Spline"

**Beweis** Seien  $f, s \in A$ , und  $s \in S_h^{(3,2)}[a, b]$ . Zu zeigen:

$$E(s) \leq E(f)$$

Wir betrachten  $f - s$ :

$$\begin{aligned} E(f - s) &= \int_a^b (f''(x) - s''(x))^2 dx \\ &= \int_a^b (f'')^2 dx - 2 \int_a^b f'' s'' dx + \int_a^b (s'')^2 dx \\ \text{🏃} &= \int_a^b (f'')^2 dx - 2 \int_a^b (f'' - s'') s'' dx - \int_a^b (s'')^2 dx \end{aligned}$$

Mittlerer Term = 0  $?!?!?!?!?!?$

$$\begin{aligned} \int_a^b (f'' - s'') s'' dx &= \sum_{i=1}^n \int_{I_i} (f'' - s'') \underbrace{s''|_{I_i}}_{\in \mathbb{P}_3 \subset C^\infty} dx \\ &= \sum_{i=1}^n \left( [f' - s']_x \Big|_{x_i-1}^{x_i} - \underbrace{\int_{I_i} (f' - s') \underbrace{s'''}_{\text{konstant auf } I_i} dx}_{= s'''|_{I_i} \int_{I_i} (f-s)' dx} \right) \\ &= \left( (f' - s') \Big|_{\substack{s''=0}} \right) (x_n) - \left( (f' - s') \Big|_{\substack{s''=0}} \right) = 0, \\ &\Rightarrow \text{🏃} \Rightarrow 0 \leq E(f) - E(s) \Rightarrow E(s) \leq E(f) \end{aligned}$$

Zur Eindeutigkeit:

angenommen,  $s, \tilde{s} \in S_h^{(3,2)}[a, b]$  beides Lösungen. Zu zeigen:  $s = \tilde{s}$  Damit ist

$$\begin{aligned} E(s - \tilde{s}) = E(s) - E(\tilde{s}) = 0 \text{ (wie oben)} &\Rightarrow \int_a^b (s'' - \tilde{s}'')^2(x) dx = 0 \\ &\Rightarrow (s - \tilde{s})''(x) \quad \forall x \in [a, b] \\ &\rightarrow (s - \tilde{s}) \in \mathbb{P}_1[a, b], \quad s(x) - \tilde{s}(x) = a_0 + a_1 x \\ &\quad s(x_i) = \tilde{s}(x_i), \quad i = 0, \dots, n \\ &\Rightarrow s(x) - \tilde{s}(x) \equiv 0 \end{aligned}$$

Zur Existenz:

mit linearer Algebra: alle Bedingungen sind lineare Gleichungen in den Koeffizienten  $(a_j(i), \quad i = 1, \dots, n, \quad j = 0, 1, 2, 3)$

$$|_{I_i} = a_0^{(i)} + a_1^{(i)} x + a_2^{(i)} x^2 + a_3^{(i)} x^3$$

Also haben wir  $4n$  Bedingungen (linear!) für  $4n$  Unbekannte und somit ein quadratisches LGS mit linearer Abb. A. Aus der Eindeutigkeit folgt, dass  $\ker(A) = \{0\}$  und  $\dim \text{Bild}(A) = 4n$ . Dementsprechend haben wir eine eindeutige Lösung für jede rechte Seite  $\square$

**Beweis** (Ein Konstruktiver Existenzbeweis)

Wie oben:

$$s|_{I_i} \in \mathbb{P}_3 \rightsquigarrow s''|_{I_i} \in \mathbb{P}_1, \quad s'' \text{ stetig} \rightsquigarrow s'' \text{ Polygonzug.}$$

Sei

$$\begin{aligned} M_i &:= s''(x_i), \quad i = 0, \dots, n \\ \implies s''|_{I_i}(x) &= \frac{M_{i-1}(x_i - x) + M_i(x - x_{i-1})}{x_i - x_{i-1}} \\ \implies s''|_{I_i}(x) &= \frac{1}{h_i}(M_{i-1}(x_i - x) + M_i(x - x_{i-1})) \\ \implies s'|_{I_i}(x) &= \frac{1}{h_i}(M_{i-1} \frac{(x_i - x)^2}{2} + M_i \frac{(x - x_{i-1})^2}{2}) + c_i, \quad c_i \in \mathbb{R} \\ \implies s|_{I_i}(x) &= \frac{1}{h_i}(M_{i-1} \frac{(x_i - x)^3}{6} + M_i \frac{(x - x_{i-1})^3}{6}) + c_i(x - x_{i-1}) + d_i, \quad d_i \in \mathbb{R} \end{aligned}$$

Stetigkeit in  $x_i, \quad i = 1, \dots, n-1 \implies$  Bedingungen für  $(\mu_i, c_i, d-i)$

$\square$

Stetigkeit von  $s'$  in  $x_i$  bedeutet

$$s'_i(x_i) = s'_{i+1}(x_i) : M_i \frac{h_i}{2} + c_i = -M_i \frac{h_{i+1}}{2} + c_{i+1} \quad \left( \text{🧑} \right)$$

Interpolation:

$$\begin{aligned} s(x_i) = y_i &\implies y_{i-1} = \frac{1}{6}h_i^2 \cdot M_{i-1} + d_i, \quad y_i = \frac{1}{6}h_i^2 \cdot M_i + c_i h_i + d_i \\ &\implies \frac{y_i - y_{i-1}}{h_i} = \frac{h_i}{6}(M_i - M_{i-1}) + c_i \\ &\implies c_i = y[x_{i-1}, x_i] - \frac{1}{6}h_i(M_i - M_{i-1}), \quad d_i = y_{i-1} - \frac{1}{6}h_i^2 M_{i-1} \end{aligned}$$

in  $\left(\begin{array}{c} \text{Person} \end{array}\right)$  einsetzen.

$$\begin{aligned} \frac{1}{2}h_i M_i - \frac{1}{6}h_i(M_i - M_{i-1}) + y[x_{i-1}, x_i] &= -\frac{1}{2}h_{i+1}M_i - \frac{1}{6}h_{i+1}(M_{i+1} - M_i) + y[x_i, x_{i+1}] \\ h_i M_{i-1} + 2(h_i + h_{i+1})M_i + h_{i+1}M_{i+1} &= 6(y[x_i, x_{i+1}] - y[x_{i-1}, x_i]) \end{aligned}$$

Setze  $\mu_i = \frac{h_i}{h_i + h_{i+1}}$  und  $\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}$

$$\mu_i M_{i+1} + 2M_i + \lambda_i M_{i+1} = y[x_{i-1}, x_i, x_{i+1}]$$

Dies ergibt ein lineares Gleichungssystem für die  $M_i, i = 1, \dots, n-1$  ( $M_0 = 0, M_n = 0$ , da natürlicher Spline)

$$\begin{pmatrix} 2 & J_1 & & \dots \\ \mu_2 & 2 & J_2 & \\ & \ddots & \ddots & J_{n-2} \\ & & \mu_{n-1} & 2 \end{pmatrix} \cdot \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} 6y[x_0, x_1, x_2] \\ 6y[x_1, x_2, x_3] \\ \vdots \\ 6y[x_{n-1}, x_n, x_{n+1}] \end{pmatrix}$$

Die Matrix des linearen Gleichungssystems ist "strikt diagonaldominant". Also gibt es eine eindeutige Lösung

**Definition 2.1.28.** Eine Matrix  $A \in \mathbb{R}^{m \times m}$  oder  $A \in \mathbb{C}^{m \times m}$  heißt

- "strikt diagonaldominant", wenn

$$\forall i = 1, \dots, m : |a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^m |a_{ij}|$$

- "schwach diagonaldominant", wenn

$$\forall i = 1, \dots, m : |a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^m |a_{ij}|$$

und für mindestens ein  $i$  es " $>$ " ist.

**Lemma 2.1.29.** Ist  $A$  strikt diagonaldominant, so gilt

$$\forall z \in \mathbb{R}^n(\mathbb{C}^n) : \|Az\|_{\max} \geq c \cdot \|z\|_{\max}$$

mit

$$c = \min_i \left( a_{ii} - \sum_{\substack{j=1 \\ j \neq i}} |a_{ij}| \right)$$

**Satz 2.1.30.**  $a = x_0 < x_1 < \dots < x_n = b$  Die Interpolationsprobleme mit kubischen Splines:

- i) Für "natürliche Splines"  $s(x_i) = f_i, i = 0, \dots, n, M_0 = M_n = 0$  bzw.  $s''(x_0) = s''(x_n) = 0$
- ii) Für "vollständige Splines"  $s(x_i) = f_i, i = 0, \dots, n, s'(x_0) = f'(x_0), s'(x_n) = f'(x_n)$
- iii) Für "periodische Splines"  $s(x_i) = f_i, i = 0, \dots, n$ , mit  $f_0 = f_n, s'(x_0) = s'(x_n), s''(x_0) = s''(x_n)$
- iv) Für "not-a-Knot-Splines"  $s(x_i) = f_i, i = 0, \dots, n, s'''$  stetig in  $x_1$  und  $x_{n-1} \Rightarrow s|_{I_1 \cup I_2} \in \mathbb{P}_3, S|_{I_{n-1} \cup I_n} \in \mathbb{P}_3$

sind stets eindeutig lösbar.

(kein neuer Beweis, ähnliche Beweise für ii), iii) iv))

**Frage** (Fehler bei der Splineinterpolation?).

**Erinerung.** Hermite Interpolation:

$$\|f - s\|_{\max} \leq \frac{1}{4!} h^4 \left\| f^{(4)} \right\|_{\max}$$

**Satz 2.1.31.**  $a = x_0 < x_1 < \dots < x_n = b$  und  $f \in C^4[a, b], h := \max_{i=1, \dots, n} (x_i - x_{i-1})$  Dann ist für den interpolierenden kubischen Spline

$$\|f - s\|_{\max[a, b]} \leq h^4 \left\| f^{(4)} \right\|_{\max[a, b]}$$

**Beweis**  $(f - s)(x_i) = 0$ . Dann ist auf jeden Teilintervall  $I_i$  das Polynom  $p_0 \equiv 0$  die lineare Interpolierende zu  $f - s$ . Die Interpolations-Fehlerabschätzung liefert

$$\|(f - s) - p_0\|_{\max I_i} = \|f - s\|_{\max I_i} \leq \frac{1}{2} h_i^2 \left\| (f - s)'' \right\|_{\max I_i} = \frac{1}{2} h_i^2 \|f'' - s''\|_{\max I_i}$$

Sei nun  $p_i \in \mathbb{P}_1$  das Polynom, das  $f''(x_{i-1})$  und  $f''(x_i)$  interpoliert. Dann ist

$$\begin{aligned} \|f'' - s''\|_{\max I_i} &\leq \|f'' - p_i\|_{\max I_i} + \|p_i - s''\|_{\max I_i} \\ &\leq \frac{1}{2} h_i^2 \|f^{(4)}\|_{\max I_i} + \max_{j=i-1, i} |f''(x_j) - s''(x_j)| \end{aligned}$$

Siehe oben:

Die Matrix  $A$  ist auf dem Tafelbild ... zu sehen!! ☺

$$A = \begin{pmatrix} 2 & \lambda_i \\ \mu_i & \ddots & \ddots \\ & \ddots & 2 \end{pmatrix} \implies \|Az\| \geq c \|z\|_{\max} \text{ mit } c = 1, \text{ also } \|z\|_{\max} \leq \|Az\|_{\max}$$

mit  $z_i = f''(x_i) - M_i$  folgt

$$\max_i |f''(x_i) - M_i| \leq \max_i |\mu_i f''(x_{i-1}) + 2f''(x_i) + \lambda_i f''(x_{i+1})|$$

□

## 2.2 Trigonometrische Interpolation

Ein Ansatz zur Interpolation von periodischen Signalen/Daten mit sin/cos-Funktionen z.B. bei akustischen Signalen.

### 2.2.1 Zum Hintergrund

”Fourier-Transformation”: auch für nicht periodische Funktionen:

$$f: \mathbb{R} \rightarrow \mathbb{C}: \mathcal{F}(f)(y) := \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) e^{-ixy} dx$$

mit  $i^2 = -1$

$$e^{a+ib} := e^a \cdot (\cos(b) + i \sin(b))$$

Dann ist

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} F(y) e^{-ixy} dy$$

z.B. für alle  $f \in L^1(\mathbb{R})$

### 2.2.2 Fourier-Reihen

Für periodische Funktionen, z.B.  $f: \mathbb{R} \rightarrow \mathbb{C}$   $2\pi$ -periodisch, also  $f(x + 2\pi) = f(x) \quad \forall x \in \mathbb{R}$  Damit

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx) + \sin(kx)$$

bzw.

$$f(x) = \sum_{k \in \mathbb{Z}} c_k \cdot e^{ikx}$$

mit

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \cdot e^{-iks} ds \in \mathbb{C}$$

Damit ist

$$a_k = c_k + c_{-k}, \quad b_k = i(c_k - c_{-k})$$

bzw.

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(s) \cdot \cos(ks) ds$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(s) \cdot \sin(ks) ds$$

Konvergenz der Fourierreihe für beliebige  $L^1$  oder  $L^2$ -Funktionen auf  $(-\pi, \pi)$ , bzw. stetige, periodische Funktionen auf  $[-\pi, \pi]$   
Abgebrochene Reihe/Partialsummen  $\rightarrow$  Approximation der Funktion  $f$ .

### 2.2.3 Diskrete Fourier-Transformation

Gegeben Werte  $a_k, k = 0, \dots, n-1$  an Punkten  $x_k = k \frac{2\pi}{n}$  Mit den Koeffizienten

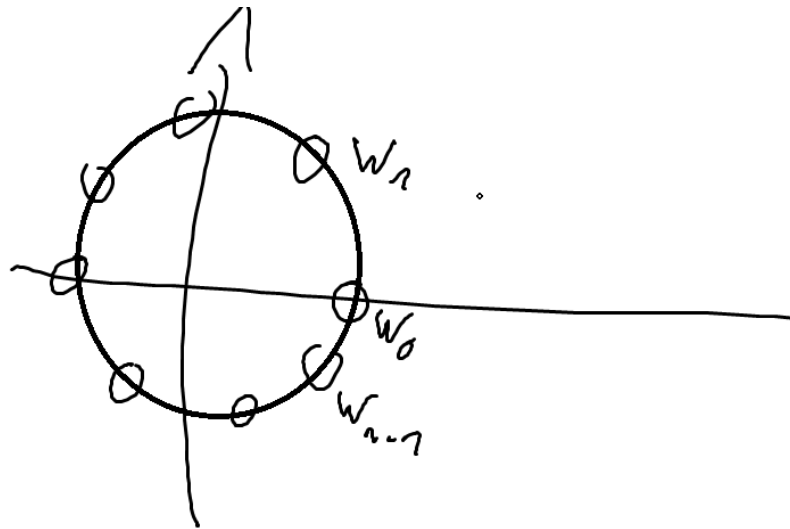
$$\hat{a}_j = \sum_{k=0}^{n-1} a_k \cdot e^{-i \cdot j \cdot k \cdot \frac{2\pi}{n}}$$

ist dann

$$a_k = \frac{1}{n} \sum_{j=0}^{n-1} \hat{a}_j \cdot e^{ijk \frac{2\pi}{n}} = \frac{1}{n} \sum_{j=0}^{n-1} \hat{a}_j \cdot \left( \cos\left(jk \frac{2\pi}{n}\right) + i \sin\left(jk \frac{2\pi}{n}\right) \right)$$

Diskrete Fourier Transformation lässt sich auf Polynom-Interpolation zurückführen mit

$$w := e^{ix}, \quad w_k := e^{ixk} = e^{i2\pi \frac{k}{n}}$$



Dann ist DFT gerade

$$t_n(x) = \sum_{j=0}^{n-1} \frac{\hat{a}_j}{n} \cdot e^{ijx} = \sum_{j=0}^{n-1} \frac{\hat{a}_j}{n} w^j = p(w), \quad \text{mit } p \in \mathbb{P}_{n-1}$$

Polynominterpolation wie in 2.1 - 2.3 geht ganz genau so für Komplexwertige Polynome  $p \in \mathbb{P}_n[\mathbb{C}]$ . Daraus folgt die Existenz und Eindeutigkeit eines interpolierenden Polynoms für Werte  $a_k$  an Punkten  $w_k$ ,  $k = 0, \dots, n-1$ . Berechnung der Fourierkoeffizienten  $\hat{a}_j$  nicht über Polynom-Methode, sondern entsprechend obiger Formel, bzw. über "Fast Fourier Transformation", mit Aufwand  $\mathcal{O}(n \cdot \log n)$  statt  $\mathcal{O}(n^2)$ .

Anwendung: z.B. MP3-Kompression von Audio-Dateien.



## Kapitel 3

# Numerische Integration

### 3.1 Numerische Integration

Berechnung von Integralen, z.B. zur Flächen- oder Volumenberechnung, aber auch notwendig in komplexeren Formeln/Algorithmen, z.B. Fourier-Integrale, Numerik partieller-Differentialgleichungen. Oft nicht (leicht) von Hand zu berechnen,  $\Rightarrow$  Algorithmen zur näherungsweisen Berechnung von Integralen. Viele typische "Quadraturformeln" haben für  $f \in C[a, b]$  die Form

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

d.h. Kombination von Punktauswertungen mit Stützstellen  $a \leq x_0 < x_1 \dots x_n \leq b$

**Erinnerung/Beispiel** (Rieman-Integral).

z.B. Rieman-Summe

$$I_h(f) := \sum_{i=1}^n f(x_{i-1}) \cdot (x_i - x_{i-1})$$

$f$  Rieman-Integrierbar  $\leadsto I_h(f) \rightarrow I(f)$  für  $h \rightarrow 0$ ,  $h := \max(x_i - x_{i-1})$

#### 3.1.1 Interpolatorische Quadraturformel

Kennt man eine Polynom-Interpolation von  $f$ , (oder Hermite-), kann man statt  $f$  einfach die Interpolierende integrieren. Integration über Polynome ist einfach. Zu  $a \leq x_0 < x_1 \dots < x_n \leq b$  sei  $P_n \in \mathbb{P}_n$  das interpolierende Polynom zu  $f$  mit  $P_n(x_i) = f(x_i), i = 0, \dots, n$  setze dann

$$I^{(n)}(f) := \int_a^b P_n(x) dx = \int_a^b \sum_{i=0}^n f(x_i) L_i^{(n)}(x) dx = \sum_{i=0}^n f(x_i) \int_a^b L_i^{(n)}(x) dx$$

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

Wie groß ist der Fehler  $I(f) - I^{(n)}$ ?

Mit der Formel für den Interpolationsfehler folgt:

**Satz 3.1.1.** Für die Lagrange-Quadraturformel  $I^{(n)}$  gilt, falls  $f \in C^{n+1}[a, b]$ :

$$I(f) - I^{(n)}(f) = \int_a^b f(x) - p_n(x) dx = \int_a^b \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{j=0}^n (x - x_j) dx$$

also

$$\left| I(f) - I^{(n)}(f) \right| \leq \frac{1}{(n+1)!} \cdot \max_{[a,b]} |f^{(n+1)}| \cdot \left| \int_a^b \prod_{j=0}^n (x - x_j) dx \right|$$

**Bemerkung 3.1.2.** man kann auch zeigen:

$$I(f) - I^{(n)}(f) = \int_a^b f[x_0, \dots, x_n, x] \prod_{j=0}^n (x - x_j) dx$$

Interpolatorische Integrationsformeln,  $I^{(n)}(f) := \int_a^b p_n(x) dx$ ,  $p_n \in \mathbb{P}$  des Interpolationspolynoms zu  $f$  in  $x_0, \dots, x_n \in [a, b]$  Nach Konstruktion: die interpolierende Quadraturformel ist "exakt" für beliebige Polynome  $p \in \mathbb{P}_n$ , wegen der Eindeutigkeit der Interpolationspolynome.

**Definition 3.1.3.** Eine Quadraturformel  $I^{(n)}$  wird (mindestens) "von der Ordnung m" genannt, falls durch sie alle Polynome vom Grad  $\leq m - n$  exakt integriert werden.

Damit sind die interpolatorischen Quadraturformeln  $I^{(n)}$  mindestens von der Ordnung  $n + 1$ .

**Beispiel 3.1.4** (Lagrange-Quadraturen mit  $n+1$  Stützstellen mit gleichen Abständen).

(a) "abgeschlossene Newton-Cotes-Formeln":

$a, b$  sind Stützstellen,  $x_i = a + ih$ ,  $i = 0, \dots, n$  mit  $h = \frac{b-a}{n}$

(b) "offene Newton-Cotes-Formeln":

$a, b$  sind keine Stützstellen,  $x_i = a + (i+1)h$ ,  $i = 0, \dots, n$ ,  $h = \frac{b-a}{n+2}$

Die ersten Newton-Cotes-Formeln sind:

**abgeschlossen:**

$$\begin{aligned}
 I^{(1)} &= \frac{b-a}{2} (f(a) + f(b)) && \text{"Trapezregel"} \\
 I^{(2)} &= \frac{b-a}{6} \left( f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) && \text{"Keplersche Fassregel"} \\
 I^{(3)} &= \frac{b-a}{8} (f(a) + 3f(a+h) + 3f(b-h) + f(b)) && \text{"}\frac{3}{8}\text{-Regel"}
 \end{aligned}$$

**offen:**

$$\begin{aligned}
 I^{(0)}(f) &:= (b-a) \cdot f\left(\frac{a+b}{2}\right) && \text{"Mittelpunktsregel"} \\
 I^{(1)}(f) &:= \frac{b-a}{2} (f(a+h) + f(b-h)) \\
 I^{(2)}(f) &:= \frac{b-a}{3} \left( 2f(a+h) - f\left(\frac{a+b}{2}\right) + 2f(b-h) \right)
 \end{aligned}$$

Mit den Interpolationsabschätzungen und Integral-Mittelwertsätzen zeigt man:

**Satz 3.1.5** (Quadraturfehler Newton-Cotes-Formeln).

i) Für die Trapezregel  $I^{(1)}$  mit  $f \in C^2[a, b]$  gilt:

$$\int_a^b f(x) \, dx - \frac{b-a}{2} (f(a) + f(b)) = -\frac{(b-a)^3}{12} f''(\xi) \text{ mit } \xi \in [a, b]$$

ii) Für die Simpson-Regel  $I^{(2)}$  mit  $f \in C^4[a, b]$  gilt:

$$\int_a^b f(x) \, dx - \frac{b-a}{2} (f(a) + 4f(\frac{a+b}{2}) + f(b)) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi) \text{ mit } \xi \in [a, b]$$

iii) Für die Mittelpunktformel  $I^{(0)}$  mit  $f \in C^2[a, b]$  gilt:

$$\int_a^b f(x) \, dx - (b-a)f\left(\frac{a+b}{2}\right) = -\frac{(b-a)^3}{24} f^{(2)}(\xi) \text{ mit } \xi \in [a, b]$$

**Beweis** zu i)

$$\begin{aligned}
 \int_a^b f(x) \, dx - \int_a^b p_n(x) \, dx &= \int_a^b f(x) - p_n(x) \, dx \\
 &= \int_a^b f''(\xi(x)) \cdot \frac{1}{2} \cdot (x-a)(x-b) \, dx \\
 &= f''(\xi) \frac{1}{2} \int_a^b (x-a)(x-b) \, dx \\
 &= \frac{1}{12} f''(\xi) (b-a)^3
 \end{aligned}$$

□

**Bemerkung 3.1.6.** zu iii) Mittelpunktsformel ist exakt nicht nur für  $p \in \mathbb{P}_0$  sondern sogar für alle  $p \in \mathbb{P}_n$

**Bemerkung 3.1.7.** Sind neben  $f$  auch die Ableitungen  $f'(x)$  bekannt, dann kann man auch eine Hermite-Interpolation zur Herleitung von Quadraturformeln nehmen, die Hermite- Interpolationsfehlerabschätzung überträgt sich dann auf die Quadraturfehlerabschätzung.

Um ein Integral besser zu approximieren, wird typischerweise nicht der Polynomgrad weiter erhöht, sondern eine Quadraturformel mit relativ geringen Grad auf Teilintervallen immer kleinerer Größe genutzt:

z.B.  $a = y_0 < y_1 < \dots < y_N = b$  mit Teilintervallen  $I_i = [y_{i-1}, y_i]$ :

$$\int_a^b f(x) dx = \sum_{i=1}^N \int_{I_i} f(x) dx \approx \underbrace{\sum_{i=1}^N \int_{I_i} \underbrace{I_{[y_{i-1}, y_i]}^{(n)} f}_{\text{Interpolierende}} dx}_{I_h^{(n)}(f)}$$

und

$$\begin{aligned} \int_a^b f(x) dx - I_h^{(n)}(f) &= \sum_{i=1}^N \int_{I_i} f(x) dx - \int_a^b \left( I_{[y_{i-1}, y_i]}^{(n)} f \right) (x) dx \\ &\leq \sum_{i=1}^N c_m \cdot (y_{i-1} - y_i)^{m+1} \left| f^{(m)}(\xi_i) \right| \\ &\leq \sum_{i=1}^N c_m h^{m+1} \cdot \left\| f^{(m)} \right\|_{\max[a, b]} \\ &\leq c_m (b - a) \frac{h}{h_{\min}} h^m \cdot \left\| f^{(m)} \right\|_{\max} \end{aligned}$$

mit

$$N \leq \frac{b-a}{h_{\min}}, \quad \frac{h}{h_{\min}} = 1, \text{ falls alle Teilintervalle gleich lang sind}$$

**Beispiel 3.1.8.**  $y_i = a + ih$ ,  $h = \frac{b-a}{N}$ ,  $i = 0, \dots, N$  gleichgroße Teilintervalle. Summierte Trapezregel

$$\begin{aligned} I_h^{(1)}(f) &:= \frac{h}{2} \left( f(a) + \sum_{i=1}^{N-1} 2f(y_i) + f(b) \right), \\ \left| I(f) - I_h^{(1)}(f) \right| &\leq \frac{b-a}{12} h^2 \left\| f^{(2)} \right\|_{\max[a, b]} \end{aligned}$$

Summierte Simpsonregel:

$$I_h^{(2)}(f) := \frac{h}{6} \left( f(a) + \sum_{i=1}^{N-1} 2f(y_i) + \sum_{i=1}^N 4f\left(\frac{y_{i-1} + y_i}{2} + f(b)\right) \right),$$

$$\left| I(f) - I_h^{(2)}(f) \right| \leq \frac{b-a}{2880} h^4 \left\| f^{(4)} \right\|_{\max}$$

Summierte Mittelpunkregel:

$$I_h^{(0)}(f) := h \sum_{i=1}^N f\left(\frac{y_{i-1} + y_i}{2}\right),$$

$$\left| I(f) - I_h^{(0)}(f) \right| \leq \frac{b-a}{24} h^2 \left\| f^{(2)} \right\|_{\max}$$

**Bemerkung 3.1.9.** Ähnlich geht es für Hermite-Splines, d.h. lokale Hermite-Interpolierende

**Motivation.** Mittelpunktsregel und Simpson-Regel sind von höherer Ordnung als man es durch den Polynomgrad alleine erwarten würde, anscheinend allein durch die geschickte Wahl der Stützstellen.

**Frage.** Wie gut kann man werden bei optimaler Wahl der Stützstellen?

### 3.1.2 Gauß-Quadraturformeln

Man sieht leicht, dass die Maximale Ordnung einer interpolierenden Quadraturformel nach oben begrenzt ist

**Lemma 3.1.10.** Eine obere Grenzen für die Ordnung einer interpol. Quadraturformel  $I^{(n)}$  mit  $n+1$  Stützstellen ist  $2n+2$

**Beweis** Wäre Ordnung höher, könnte man alle Polynome vom Grad  $2n+2$  exakt integrieren. Für das Polynom

$$p(x) := \prod_{i=0}^n (x - x_i)^2 \in \mathbb{P}_{2n+2}$$

gilt

$$\forall i = 0, \dots, n : p(x_i) = 0$$

also

$$I^{(n)}(p) = 0$$

da

$$I^{(n)}(p) = \sum_{j=0}^n w_j p(x_j)$$

aber

$$\forall x : p(x) \geq 0, \text{ also } p \not\equiv 0$$

demnach

$$\int_a^b p(x) \, dx > 0$$

□

Man kann bei geschickter Wahl der Stützstellen also alle Polynome  $p \in \mathbb{P}_{2n+1}$  exakt integrieren. Ein Polynom  $p \in \mathbb{P}_{2n+1}$  kann man immer zerlegen in

$$p(x) = r(x) \cdot q(x) + s(x)$$

mit  $q \in \mathbb{P}_{n+1}$  fest vorgegeben,  $\deg q = n + 1$ ,  $r, s \in \mathbb{P}_n$ . Z.B. für  $q(x) = x^{n+1}$ :

$$p(x) = \sum_{i=0}^{2n+1} a_i x^i \implies r(x) = \sum_{i=0}^n a_{i+n+1} x^i, \quad s(x) = \sum_{i=0}^n a_i x^i$$

für eine Wahl  $a \leq x_0 < x_1 < \dots < x_n \leq b$  wählen wir

$$q(x) := \prod_{i=0}^n (x - x_i) \in \mathbb{P}_{n+1}$$

**Frage.** Quadraturformel für  $p$ ?

Es ist

$$\int_a^b p(x) \, dx = \underbrace{\int_a^b r(x)q(x) \, dx}_{I^{(n)}(r \cdot q) + \text{Rest}} + \underbrace{\int_a^b s(x) \, dx}_{I^{(n)}(s)}$$

Falls also

$$\int_a^b p(x) \, dx \stackrel{!}{=} I^{(n)}(p)$$

sein soll für alle  $p \in \mathbb{P}_{2n+1}$ , dann muss

$$\int_a^b r(x)q(x) \, dx = 0 \text{ für alle } r \in \mathbb{P}_n$$

**Frage.** gibt es ein  $q \in \mathbb{P}_{n+1}$ , bzw

$$x_0 < \dots < x_n \in [a, b], \quad q(x) = \prod_{i=0}^n (x - x_i)$$

so, dass

$$\int_a^b r(x)q(x) \, dx = 0$$

für alle  $r \in \mathbb{P}_n$ ?

Wir betrachten den Raum  $\mathbb{P}_{n+1}$  mit Basis  $\{1, x, x^2, \dots, x^{n+1}\}$  mit Skalarprodukt

$$(r, q) := \int_a^b r(x)q(x) dx \quad \text{für alle } r, q \in \mathbb{P}_{n+1}$$

Gram-Schmidt-Orthogonalisierungsverfahren:

$$p_0(x) := 1, \quad p_n(x) := x^k - \sum_{j=0}^{k-1} \frac{(x^k, p_j)}{(p_j, p_j)} p_j, \quad k = 1, \dots, n+1$$

Also steht  $p_{n+1}$  senkrecht auf  $\text{span}\{p_0, \dots, p_n\} = \mathbb{P}_n$ , d.h.

$$(r, p_{n+1}) = 0 \quad \forall r \in \mathbb{P}_n$$

dementsprechend ist  $p_{n+1}$  Kandidat für unser Polynom  $q$ ,  $q = p_{n+1}$ , da die führende Potenz  $1 \cdot x^{n+1}$  die gleiche ist. Man kann zeigen, dass alle  $p_k$  jeweils  $k$  verschiedene, einzelne Nullstellen hat, damit  $q = p_{n+1}$  gerade Nullstellen  $x_i$ ,  $a \leq x_0 < \dots < x_n \leq b$  hat, und damit

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

**Satz 3.1.11** ("Gauß-Quadratur").

Es gibt genau eine interpolatorische Quadraturformel zu  $n+1$  paarweise verschiedene Stützstellen über dem Intervall  $[-1, 1]$  mit der optimalen Ordnung  $2n+1$ . Die zugehörigen Stützstellen sind die Nullstellen des  $(n+1)$ -ten Legendrepolynoms  $L_{n+1} \in \mathbb{P}_{n+1}$ , und die Gewichte sind

$$w_i = \int_{-1}^1 \prod_{j \neq i} (x - x_j)^2 dx > 0$$

Für  $f \in C^{2n+1}[-1, 1]$  ist der Quadraturfehler

$$\int_a^b f(x) dx - \sum_{i=0}^n w_i f(x_i) = \frac{1}{(2n+2)!} f^{(2n+2)}(\xi) \int_{-1}^1 \prod_{i=0}^n (x - x_i)^2 dx$$

**Beweis**

**gewichte  $w_i$ :** Sei  $L_i^{(n)}$  das  $i$ -te Lagrange-Basispolynom

$$\frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \in \mathbb{P}_n$$

$$L_i^{(n)}(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & \text{sonst} \end{cases}$$

Damit ist  $(L_i^{(n)})^2 \in \mathbb{P}_{2n}$ , und Gauß-Quadraturformel exakt

$$\begin{aligned} \int_{-1}^1 (L_i^{(n)})^2 dx &= \sum_{j=0}^n w_j \cdot L_i^{(n)2}(x_j) \\ &= \sum_{j=0}^n w_j \cdot \delta_{ij} = w_i \end{aligned}$$

Also ist  $w_i > 0$

**Eindeutigkeit:** ergibt sich aus Orthogonalität von  $q$  zu  $\mathbb{P}_n$ , orthogonaler Unterraum ist 1-dimensional, alle Vielfachen eines Polynoms  $\neq 0$ , damit haben alle Polynome des orthog. Unterraums die gleichen Nullstellen

**Fehlerabschätzung:** Wir nutzen eine Hermite-Interpolation als Hilfsmittel.

Zu  $f \in C^{2n+2}[-1, 1]$  sei  $h \in \mathbb{P}_{2n+1}$  die Hermite-Interpolierende zu  $f(x_i)$ ,  $f'(x_i)$ ,  $i = 0, \dots, n$ . Dafür hatten wir die Abschätzung

$$f(x) - h(x) = \frac{1}{(2n+2)!} f^{(2n+2)}(\xi_x) \cdot \prod_{i=0}^n (x - x_i)^2$$

Damit ist

$$\begin{aligned} I(f) - I^{(n)}(f) &= (I(f) - I(h)) - \left( I^n(f) - \underbrace{I(h)}_{I^{(n)}(h)} \right) \\ &= \int_{-1}^1 f(x) - h(x) dx - \sum_{i=0}^n w_i \underbrace{(f(x_i) - h(x_i))}_{=0} \\ &= \int_{-1}^1 \frac{1}{(2n+2)!} f^{(2n+2)}(\xi_x) \cdot \prod (x - x_i)^2 dx \\ &\stackrel{\text{MWS}}{=} \frac{1}{(2n+2)!} f^{(2n+2)}(\xi) \cdot \prod_{j=0}^n (x - x_i)^2 \end{aligned}$$

□

"Legendre-Polynome":  
 $\overline{L_k(x)}$ , Orthogonalpolynome bzgl.

$$(r, s) = \int_{-1}^1 r(x)s(x) dx$$



Es ist

$$L_0(x) \equiv 1, \quad L_1(x) = x, \quad (k+1)L_{k+1}(x) = (2k+1)x \cdot L_k(x) - kL_{k-1}(x)$$

also

$$L_2(x) = \frac{1}{2}(3x^2 - 1) = \frac{3}{2}x^2 - \frac{1}{2}$$

Eine andere Formel ist

$$L_k(x) = \frac{1}{2^k \cdot k!} \frac{d^k}{dx^k} \left( (x^2 - 1)^k \right)$$

**Bemerkung 3.1.12.** Analoges Vorgehen bei Integration mit einer Gewichtsfunktion

$$w(x) : I_w(f) := \int_a^b f(x) \cdot w(x) dx$$

Orthogonalisierung bzgl des gewichteten Skalarprodukts

$$(r, s)_w := \int_a^b r(x) \cdot s(x) \cdot w(x) dx, \quad w > 0 \text{ fast überall, z.B. stetig}$$

**Beispiel 3.1.13.** Für  $w(x) := \frac{1}{\sqrt{1-x^2}}$  auf  $[-1, 1]$  ergeben sich als Orthogonalpolynome gerade die Tschebyscheff Polynome  $T_k(x)$ .

Für  $w(x) \equiv 1$ : Legendre-Polynome, s.o.

Für Integration auf dem Intervall  $[a, b]$ : Transformation

$$[-1, 1] \rightarrow [a, b], \quad \int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{a+b}{2} + t \frac{b-a}{2}\right) dt$$

Um Integrale genauer zu berechnen, wird auch bei den Gauß-Formeln nicht unbedingt der Polynomgrad weiter erhöht so, dass wieder eine Summe über kleinere Teilintervalle  $I_j, j = 1, \dots, N$  genutzt,  $I_j = [y_{j-1}, y_j]$ ,  $a = y_0 < y_1 < \dots < y_N = b$  Und für jedes Teilintervall die entsprechend transformierte Quadraturformel. Dafür bekommt man dann eine Fehlerabschätzung

$$\begin{aligned} |I(f) - I_h^n(f)| &= \left| \int_a^b f(x) dx - \sum_{j=1}^N \frac{y_j - y_{j-1}}{2} \cdot \left( \sum_{i=0}^n w_i \cdot \tilde{f}_j(x_i) \right) \right| \\ &\leq (b-a) \cdot c \cdot h^{2n+2} \cdot \left\| f^{(2n+2)} \right\|_{\max[a,b]} \end{aligned}$$

mit  $h := \max_{j=1, \dots, N} (y_j - y_{j-1})$ . Dabei ist  $c$  unabhängig von  $a, b, h, f$ . Bei Halbierung der Teilintervalllängen reduziert sich der Quadraturfehler also um Faktor  $\left(\frac{1}{2}\right)^{2n+2}$ .

### 3.1.3 Richardson-Extrapolation

Eine Idee/Methode, die man auch in anderen Situationen gut und gerne anwenden kann:

Angenommen Berechnungsvorschrift mit Diskretisierungsgröße  $h > 0$ ,  $h \searrow 0$  so, dass wir für die Approximation einer Größe  $E_0$  eine Entwicklung der berechneten Größe  $E(h)$  haben der Form

$$E(h) = E_0 + c_1 h^{k_1} + c_2 h^{k_2} \in \mathcal{O}(h^{k_2}) \text{ mit } 0 < k_1 < k_2$$

Dann können wir Berechnungen mit zwei verschiedenen Diskretisierungen  $h_1 > h_2 > 0$  durchführen, und mit diesen Ergebnissen die Ordnung  $h^{k_1}$  eliminieren

$$\begin{aligned} E(h_1) &= E_0 + c_1 h_1^{K_1} + c_2 h_1^{K_2} & | \cdot h_2^{K_1} \\ E(h_2) &= E_0 + c_1 h_2^{K_1} + c_2 h_2^{K_2} & | \cdot h_1^{K_1} \\ h_1^{K_1} E(h_2) - h_2^{K_1} E(h_1) &= (h_1^{K_1} - h_2^{K_1}) E_0 + c_1 \cdot 0 + c_2 (h_1^{K_1} h_2^{K_2} - h_2^{K_1} h_1^{K_2}) \\ \frac{h_1^{K_1} E(h_2) - h_2^{K_1} E(h_1)}{h_1^{K_1} - h_2^{K_1}} &= E_0 + c_2 \frac{h_1^{K_1} h_2^{K_2} - h_2^{K_1} h_1^{K_2}}{h_1^{K_1} - h_2^{K_1}} \end{aligned}$$

$h_2 = d h_1$  mit  $0 < d < 1$

$$\begin{aligned} \frac{h_1^{K_1} (E(h_2) - d^{K_1} E(h_1))}{h_1^{K_1} (1 - d^{K_1})} &= E_0 + c_2 \frac{(d^{K_2} - d^{K_1}) h_1^{K_1 + K_2}}{h_1^{K_1} (1 - d^{K_1})} \\ \Rightarrow \frac{E(h_2) - d^{K_1} E(h_1)}{1 - d^{K_1}} &= E_0 + c_2 \frac{d^{K_2} - d^{K_1}}{1 - d^{K_1}} h_1^{K_2} \end{aligned}$$

Durch geschickte Kombination erhalten wir eine Approximation der Ordnung  $\mathcal{O}(h^{K_2})$  auch ohne  $K_2$  explizit zu kennen.

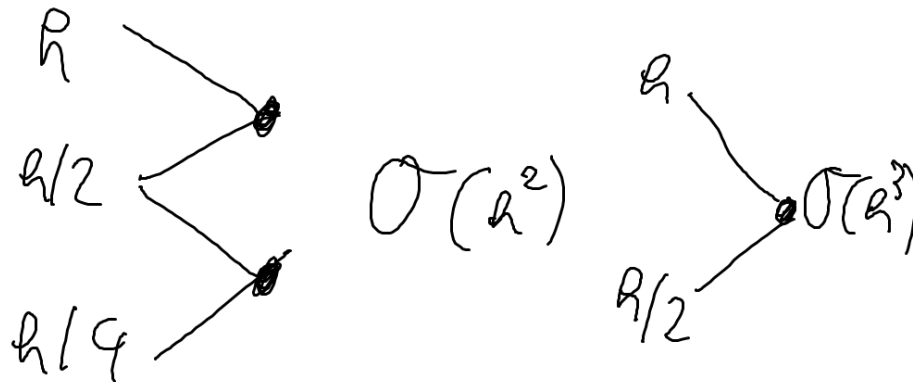
### Anwendung auf numerische Integration

Wir kennen (falls die zu integrierende Funktion glatt genug ist) die führenden Ordnungen des Quadraturfehlers vieler interpolierender Quadraturformeln. Durch geeignete Kombination von Integrationen mit gleichlangen Teilintervallen mit Längen  $h_1, h_2$  erreicht man dann eine Approximation des Integrals  $I(f)$  von besserer Ordnung. Hat man eine weitergehende Entwicklung wie

$$E(h) = E_0 + c_1 h^{K_1} + c_2 h^{K_2} + c_3 h^{K_3} \dots$$

können entsprechend auch weiter höhere Ordnungen eliminiert werden, indem genügend verschiedene Berechnungen mit verschiedenen  $h$ 's durchgeführt werden.

### Beispiel 3.1.14.



Für die summierte Trapezregel auf einer gleichmäßigen Zerlegung mit Teilintervallen der Länge  $h = \frac{b-a}{N}$ ,  $x_i = a + ih$ ,  $i = 1, \dots, N$  kann man zeigen:

**Satz 3.1.15** ("Euler-Maclaurinsche Summenformel").

Ist  $f \in C^{2m+2}[a, b]$  und

$$a(h) \left( \frac{1}{2}f(a) + \sum_{i=1}^{N-1} f(x_i) + \frac{1}{2}f(b) \right)$$

das Ergebnis der summierten Trapezregel.

Darum gilt die Entwicklung

$$\int_a^b f(x) dx =$$

$$a(h) - \sum_{k=1}^m \left[ h^{2k} \frac{B_{2k}}{(2k)!} \left( f^{(2k-1)}(b) - f^{(2k-1)}(a) \right) \right] - h^{2m+2} \frac{(b-a)}{(2m+2)!} B_{2m+2} \cdot f^{(2m+2)}(\xi)$$

mit  $\xi \in [a, b]$

mit den "Bernoulli-Zahlen"  $B_j$ ,

$$B_0 = 1, \quad B_k = - \sum_{j=0}^{k-1} \frac{k!}{j!(k-j-1)!} B_j$$

oder auch

$$\frac{x}{e^x - 1} = \sum_{j=0}^{\infty} \frac{B_j}{j!} x^j$$

(ohne Beweis)

Damit ergibt sich eine Entwicklung des Quadraturfehlers in gerade  $h$ -Potenzen  $h^2, h^4, h^6, \dots$ , falls  $f$  glatt genug ist.

Dies kann genutzt werden, um aus Werten für verschiedene  $h_l$  eine immer bessere Approximation des Integrals zu berechnen.

"Romberg-Integration":

Anwendung für  $h_l := \frac{h_0}{2^l}$ ,  $l = 0, \dots$ :

$$h_0, \frac{h_0}{2}, \frac{h_0}{4}, \frac{h_0}{8}, \dots$$

**Vorteil:** Wiederverwendung der Funktionswerte  $f(x_j)$  aus den vorherigen Zerlegungen möglich.

**Nachteil:** Im  $l$ -ten Schritt müssten  $2^l$  Operationen durchgeführt werden, was mit steigenden  $l$  relativ schnell groß wird.

**Bemerkung 3.1.16.** Folge  $h, \frac{h}{2}, \frac{h}{4}, \frac{h}{8}, \dots$  heißt auch "Romberg-Folge"

Extrapolation kann nicht nur zur Verbesserung des Berechnungsprozesses, sondern auch zur numerischen Abschätzung des Fehlers genutzt werden:

Abschätzungen wie

$$|I(f) - I_h^{(1)}(f)| \leq \frac{b-a}{12} \cdot h^2 \|f''\|_{\max}$$

sind i.A. nicht auswertbar oder liefern zu grobe Ergebnisse. Falls eine Toleranz für die Berechnung vorgegeben ist, nützt das z.B. wenig.

Legt man eine Entwicklung des Fehlers zugrunde, wie oben getan, dann kann

man auch versuchen aus 2 Berechnungen den Fehler sowie eine optimale Diskretisierung abzuschätzen. Für

$$E(h_1) = E_0 + c_1 h_1^{K_1} + c_2 h_1^{K_2}, \quad \text{Rechnung mit } h_1, h_2 = \frac{h_1}{2}$$

$$E(h_2) = E_0 + c_1 \left(\frac{h_1}{2}\right)^{K_1} + c_2 \left(\frac{h_1}{2}\right)^{K_2}$$

Also

$$E(h) - E\left(\frac{h}{2}\right) = c_1 \cdot \left(1 - \frac{1}{2^{K_1}}\right) h^{K_1} + \mathcal{O}\left(h^{K_2}\right)$$

somit

$$c_1 = \frac{E(h) - E\left(\frac{h}{2}\right)}{1 - \frac{1}{2^{K_1}}} h^{-K_1} + \mathcal{O}\left(h^{K_2-K_1}\right) \quad K_2 - K_1 > 0$$

Schaut man für den Fehler nur die führende Ordnung an,  $E(h) - E_0 \approx c_1 h^{K_1}$ , dann ist

$$E(h) - E_0 \approx \frac{E(h) - E\left(\frac{h}{2}\right)}{1 - \frac{1}{2^{K_1}}}$$

Ist eine Toleranz TOL für den Fehler vorgegeben, dann ist die dazu passende Gitterweite

$h_{\text{opt}}$  durch  $\text{TOL} \approx c_1 h_{\text{opt}}^{K_1}$  bestimmt, also

$$h_{\text{opt}} = \left(\frac{\text{TOL}}{c_1}\right)^{\frac{1}{K_1}}$$

”a-posteriori“-Fehlerabschätzung, ”im Nachhinein“, aus numerischen Resultaten versuchen, den Fehler abzuschätzen

”a-priori“: Im Vorhinein, Abschätzung durch Daten etc, ohne vorherige Berechnung

## Kapitel 4

# Numerische Lösung Linearer Gleichungssysteme

**Aufgabe:** Zu gegebener Matrix  $A \in \mathbb{R}^{n \times n}$  und rechter Seite  $b \in \mathbb{R}^n$  ist ein  $x \in \mathbb{R}^n$  gesucht, so dass  $Ax = b$  gilt.

**Satz 4.0.1.** Sei  $n \in \mathbb{N}$ ,  $A \in \mathbb{R}^{n \times n}$  mit  $\det(A) \neq 0$ . Dann existiert zu jedem  $b \in \mathbb{R}^n$  genau ein  $x \in \mathbb{R}^n$  so, dass  $Ax = b$ .

**Beweis** Lineare Abbildung zu Matrix  $A$  ist bijektiv, wenn  $\det(A) \neq 0$ . Dann existiert auch die inverse Matrix  $A^{-1}$  und  $x = A^{-1}b$   $\square$

Berechnung der Lösung?

**”Direkte Verfahren”:** berechne  $x$  (bis auf Rundungsfehler, Zahlendarstellung ...)

**”Iterative verfahren”:** ausgehend von  $x_0 \in \mathbb{R}^n$  berechne Folge  $x_1, x_2, \dots, \in \mathbb{R}^n$  und  $x_i \rightarrow x$  ( $i \rightarrow \infty$  bzw.  $x_1, \dots, x_N \in \mathbb{R}^N$  mit  $\|x_N - x\| \leq TOL$ )

Jetzt:

### 4.1 Direkte Verfahren

Es gibt einige spezielle Matrizen, für die sich die Lösung einfach berechnen lässt:

**Diagonalmatrizen:**

$$A = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{NN} \end{pmatrix} \Rightarrow x_i = \frac{b_i}{a_{ii}} \quad a_{ii} \neq 0, \text{ wenn } \det(A) \neq 0, \text{ für alle } i = 1, \dots, n$$

## Dreiecksmatrizen

**Definition 4.1.1.** Eine Matrix  $A \in \mathbb{R}^{n \times n}$  heißt rechte obere Dreiecksmatrix, wenn

$$\forall j < i : a_{ij} = 0, a_{ii} \neq 0, i = 1, \dots, n$$

Bei einer oberen Dreiecksmatrix lässt sich das lineare Gleichungssystem einfach von unten nach oben auflösen:

$$x_{11} = \frac{b_n}{a_{nn}}, \quad x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j>i} a_{ij} b_j \right), \quad i = n-1, \dots, 1$$

**Definition 4.1.2.** Eine Matrix  $A \in \mathbb{R}^{n \times n}$  heißt linke untere Dreiecksmatrix, wenn

$$\forall j > i : a_{ij} = 0, a_{ii} \neq 0, i = 1, \dots, n$$

auflösen:

$$x_i = \frac{b_1}{a_{11}}, \quad x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} b_j \right), \quad i = 2, \dots, n$$

**Produkte aus Dreiecksmatrizen** z.B.  $A = L \cdot R$

$$Ax = b \Leftrightarrow L \cdot \underbrace{Rx}_y = b \Leftrightarrow \begin{cases} Ly = b \\ Rx = y \end{cases}$$

**Definition 4.1.3.** Die Zerlegung einer Matrix  $A \in \mathbb{R}^{n \times n}$  in ein Produkt  $A = LR$  mit linker unterer Dreiecksmatrix  $L$  und oberer rechter Dreiecksmatrix  $R$  löst "LR-Zerlegung"

**Bemerkung 4.1.4.** Im englischen: "LU-decomposition",  $L$ : lower,  $U$ : upper

## 4.2 LR-Zerlegung einer Matrix

**Bemerkung 4.2.1.** Nicht jede reguläre Matrix  $A$  mit  $\det(A) \neq 0$  besitzt eine LR-Zerlegung.

**Beispiel 4.2.2.**  $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \neq L \cdot R$ , da

$$\begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix} \Rightarrow (L \cdot R)_{11} = l_{11} r_{11} = a_{11} = 0 \Rightarrow l_{11} = 0 \vee r_{11} = 0$$

Bei Vertauschung der Zeilen funktioniert es aber

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Zu jeder regulären Matrix  $A \in \mathbb{R}^{n \times n}$  mit  $\det(A) \neq 0$  gibt es eine Permutationsmatrix  $P$  so, dass  $P \cdot A$  eine LR-Zerlegung besitzt.

Berechnung der LR-Zerlegung: Gauß Algorithmus

Wir nehmen zunächst an, dass alle Operationen durchgeführt werden können, dass alle auftretenden Diagonalelemente  $\neq 0$  sind. Erster Schritt:

$$A^{(1)} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \rightarrow A^{(2)} = \begin{pmatrix} a_{11} & \dots & \dots & a_{1n} \\ 0 & \tilde{a}_{22} & \dots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & \dots & \dots & a_{nn} \end{pmatrix}$$

durch subtrahieren von Vielfachen der ersten Zeile von den anderen Zeilen. Diese Operation lässt sich auch als Matrix-Produkt darstellen:  $A^{(2)} = L_1 A^{(1)}$  mit linker unterer Dreiecksmatrix  $L_1$

$$L_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ -\frac{a_{n1}}{a_{11}} & 0 & \dots & 1 \end{pmatrix}$$

Weiter entsprechend

$$A^{(i)} = \begin{pmatrix} * & * & * & * \\ 0 & \ddots & * & * \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{pmatrix} \rightsquigarrow \begin{pmatrix} * & * & * \\ 0 & \ddots & * \\ 0 & 0 & * \end{pmatrix}$$

durch Matrix

$$L_i = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{a_{i+1,i}}{a_{ii}} & \ddots & 0 & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \ddots & 0 \\ 0 & 0 & -\frac{\tilde{a}_{n,i}}{a_{ii}} & 0 & 0 & 0 & \ddots \end{pmatrix}$$

für  $i = 2, \dots, n-1$ :

$$L_{n-1} \cdots L_i \cdot L_{i-1} \cdots L_1 \cdot A = R$$

Man zeigt leicht:

$$L_{n-1} \cdots L_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ l_{2n} & \dots & l_{n,n-1} & 1 \end{pmatrix}$$



und

$$(L_{n-1} \cdots L_1)^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -l_{21} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ -l_{n,1} & \cdots & l_{n,n-1} & 1 \end{pmatrix} =: L$$

Dementsprechend

$$(L_{n-1} \cdots L_1)^{-1} (L_{n-1} \cdots L_1) A = (L_{n-1} \cdots L_1)^{-1} R \implies A = LR$$

Tritt beim Algorithmus ein Diagonalelement  $\tilde{a}_{jj} = 0$  auf, dann muss durch Zeilentausch von Zahlen  $j$  und  $k$ ,  $k > j$ , ein Diagonalelement erzeugt werden, das  $\neq 0$  ist. Falls  $\det(A) \neq 0$ , muss das immer möglich sein.

Numerisch ist es aus Konditions- und Stabilitätsgründen vorteilhaft, wenn alle  $|l_{ij}| \leq 1$  sind für  $j < i$ .

Tausch der Zeilen  $j$  und  $k$  kann auch durch Multiplikation mit einer "Permutationsmatrix"  $P = P_{ik}$  beschrieben werden

$$P_{jk} = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 0 & \cdot & \cdot & 1 \\ & & \cdot & 1 & & \cdot \\ & & \cdot & & 1 & \cdot \\ & & 1 & \cdot & \cdot & 0 \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix}$$

Matrixbeispiel siehe Tafelbild "Tafel\_20221212\_4.jpg" unten rechts.

Der Gauß-Algorithmus inklusive Zeilentausch lässt sich durch ein Matrix-Produkt darstellen

$$L_{n-1} \cdot P_{n-1} \cdots L_2 \cdot P_2 \cdot L_1 \cdot P_{1k_1} \cdot A = R$$

Man kann zeigen:

$$i < j : P_j L_i = \tilde{L}_i P_j$$

für unsere Matrizen  $L_i$  von oben, wobei in  $\tilde{L}_i$  nur Einträge  $l_{j,i}, l_{k,i}$  vertauscht sind gegenüber  $L_i$ , also

$$L_{n-1} \cdot P_{n-1} \cdots L_1 \cdot P_1 A = \underbrace{(\tilde{L}_{n-1} \cdots \tilde{L}_1)}_{()^{-1}=L} \underbrace{(P_{n-1} \cdots P_1)}_P A = R \implies PA = LR$$

Diese Matrizen  $L_i$  heißen "Frobenius-Matrizen"

**Satz 4.2.3.** Ist  $A \in \mathbb{R}^{n \times n}$  mit  $\det(A) \neq 0$ , dann gibt es eine Permutationsmatrix  $P$  so, dass  $PA$  eine LR-Zerlegung besitzt.

$$PA = LR = \frac{1}{2}L \cdot 2R$$

ist die Diagonale von  $L \begin{pmatrix} L_{11} & 0 & 0 \\ * & \ddots & 0 \\ * & * & L_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ * & \ddots & 0 \\ * & * & 1 \end{pmatrix}$ , dann die Zerlegung eindeutig. (für die Permutationsmatrix  $P$ )

**Bemerkung 4.2.4.** LR-Zerlegung kann auf dem Rechner sparsam gespeichert werden:

$$\begin{pmatrix} r_{11} & \dots & \dots & r_{1n} \\ P_{21} & \ddots & R & \vdots \\ \vdots & L & \ddots & \vdots \\ l_{n1} & & l_{n,n-1} & r_n \end{pmatrix}$$

**Bemerkung 4.2.5.** Matlab/Octave: `lu(...)`

Für manche Matrizen kann die LR-Zerlegung mit deutlich weniger Rechenoperationen berechnet werden:

### 4.2.1 LR-Zerlegung von Bandmatrizen

**Definition 4.2.6.** Eine Matrix  $A \in \mathbb{R}^{n \times n}$ ,  $A = (a_{ij})_{i,j}$  heißt "Bandmatrix" vom Bandtyp  $(m_l, m_r)$ , mit  $0 \leq m_l, m_r \leq n-1$ , wenn gilt

$$a_{jk} = 0 \text{ für } k < j - m_l \text{ oder } k > j + m_r$$

$1 + m_l + m_r$  heißt "Bandbreite" der Matrix.

**Beispiel 4.2.7.**

**Typ (0,0):** Diagonalmatrix,

**Typ (1,1):** Tridiagonalmatrix

**Typ (n-1,0):** linke untere Dreiecksmatrix,

**Typ (0,n-1):** rechte obere Dreiecksmatrix

!D Randwertproblem für gewöhnliche DGL 2. Ordnung

$$\begin{aligned} -u''(x) &= f(x) & x &\in (0,1) \\ u(0) &= u_0 \\ u(1) &= u_1 \end{aligned}$$

**Bild**

$$\begin{aligned} -u''(x) &\approx \frac{-u(x-h) + 2u(x) - u(x+h)}{h^2} \\ &\approx \frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} \end{aligned}$$

$\Rightarrow$  Tridiagonalmatrix

$$\frac{1}{h^2} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & \diagdown & \diagdown & \\ & \diagdown & \diagdown & -1 \\ 0 & & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} f_1 + u_0 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n + u_1 \end{pmatrix}$$

2D Randproblem für partielle DGL 2. Ordnung:

$$-\Delta u(x_i, y_i) \approx \frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} u$$

$$\begin{aligned} -\Delta u(x) &= f(x) \text{ in } \Omega = (0, 1)^2 \\ u(x) &= g(x) \text{ auf } (\text{da ist ein komisches gespiegeltes C})\Omega \end{aligned}$$

**Satz 4.2.8.** Sei  $A \in \mathbb{R}^{n \times n}$  Bandmatrix vom Typ  $(m_l, m_r)$ , die eine LR-Zerlegung (ohne Zeilentausch) erlaubt, dann sind die Faktoren  $L, R$  ebenfalls Bandmatrizen vom Typ  $(m_l, 0)$  bzw  $(0, m_r)$ . Der Aufwand für die Berechnung der LR-Zerlegung ist dann

$$\frac{1}{3}n \cdot m_l \cdot m_r + \mathcal{O}(n \cdot (m_l + m_r))$$



**Beweis** Nachrechnen

□

**Beispiel 4.2.9.** Tridiagonalmatrix  $(1, 1)$ :  $\mathcal{O}(n)$  Operationen  $\leadsto$  lösen eines LGS mit Tridiagonalmatrix

#### 4.2.2 Cholesky-Zerlegung

Spezialfall für symmetrische positiv definite Matrizen.

**Satz 4.2.10.** Sei  $A \in \mathbb{R}^{n \times n}$  eine Symmetrische und positiv definite Matrix. Dann besitzt  $A$  eine LR-Zerlegung (ohne Zeilentausch) mit positiven  $\tilde{a}_{ii}$ ,  $i = 1, \dots, n$ .

**Beweis** In ersten Schritt der LR-Zerlegung:  $a_{11} > 0$ , da  $A$  positiv definit ist,

dann  $e_1^T A e_1 = a_{11} > 0$ ,  $e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$  Elimination der ersten Spalte:

$$\tilde{a}_{JK} = a_{JK} - \frac{a_{j1}}{a_{11}} \cdot a_{1K} = a_{KJ} - \frac{a_{k1}}{a_{11}} \cdot a_{1J} = \tilde{a}_{KJ} \Rightarrow \tilde{A} \text{ ist Symmetrisch}$$

Ist  $\tilde{A}$  positiv definit?

Sei

$$\tilde{x} = (\tilde{x}_2, \dots, \tilde{x}_n)^T \in \mathbb{R}^{n-1}$$

beliebig. Setze

$$x_1 := \frac{-1}{a_{11}} \cdot \sum_{K=2}^n a_{1K} x_K.$$

Dann ist  $(x_1, \dots, x_n)^T \in \mathbb{R}^n$ .

Dann ist  $(A \text{ pos. def.})$ .

$$\begin{aligned} 0 &< \underbrace{\sum_{J,K=1}^n a_{JK} x_J x_K}_{x^T A x} \\ &= \sum_{J,K=1}^n a_{JK} x_J x_K + a_{11} x_1^2 + 2a_{11} \sum_{K=2}^n a_{1K} x_K + \underbrace{\frac{1}{a_{11}} \left( \sum_{K=2}^n a_{1K} x_K \right)^2 - \frac{1}{a_{11}} \left( \sum_{J,K=1}^n a_{1J} a_{1K} x_J x_K \right)}_{=0} \\ &= \sum_{j,k=2}^n \left( a_{jk} - \frac{a_{k1}}{a_{11}} \cdot a_{1j} \right) x_j x_k + a_{11} \underbrace{\left( x_1 + \frac{1}{a_{11} \sum_{k=2}^n a_{1k} x_k} \right)^2}_{=0, \text{ Wahl von } x_1} \\ &= \tilde{x}^T \tilde{A} \tilde{x} \end{aligned}$$

Also ist  $\tilde{A}$  positiv definit. □

**Satz 4.2.11.** Symmetrische positiv definite Matrizen  $A \in \mathbb{R}^{n \times n}$  gestatten eine "Cholesky-Zerlegung"

$$A = LDL^T = \tilde{L} \tilde{L}^T$$

mit positiver Diagonalmatrix

$$D = \begin{pmatrix} d_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & d_{nn} \end{pmatrix}, \quad d_{ii} > 0$$

bzw (unskalierter) linker unterer Dreiecksmatrix

$$\tilde{L} = LD^{\frac{1}{2}}, \quad D^{\frac{1}{2}} = \begin{pmatrix} \sqrt{d_{11}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{nn}} \end{pmatrix}$$

**Beweis**  $A = LR$  mit  $L = \begin{pmatrix} 1 & 0 & 0 \\ * & \ddots & 0 \\ * & * & 1 \end{pmatrix}$  "skaliert",  $R = \begin{pmatrix} r_{11} & * & * \\ 0 & \ddots & * \\ 0 & 0 & r_{nn} \end{pmatrix}$ ,

$r_{ii} > 0$ . Dann ist  $R = D\tilde{R}$  mit

$$D = \begin{pmatrix} r_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & r_{nn} \end{pmatrix}, \quad \tilde{R} = \begin{pmatrix} 1 & \frac{r_{ij}}{r_{ii}} \\ 0 & \ddots \\ 0 & 0 & 1 \end{pmatrix}$$

Also ist

$$A = A^T = (LR)^T = (LD\tilde{R})^T = \tilde{R}DL^T$$

mit linker unterer Dreiecksmatrix  $\tilde{R}^T$ , skaliert, und rechter oberer Dreiecksmatrix  $DL^T$ . Also

$$\tilde{R}^T = L, \quad DL^T = R$$

□

⇒ R muss gar nicht explizit berechnet werden, es genügt L und D zu kennen.

⇒ Rechenoperationen etwa nur halb so viele nötig, Speicherplatz ähnlich.

### 4.2.3 "Lösung" nicht regulärer Systeme

Jetzt muss die Matrix nicht quadratisch sein.

$$A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m$$

gegeben ⇒ lineares Gleichungssystem

$$Ax = b \text{ für } x \in \mathbb{R}^n$$

Lineare Algebra: Ist  $\text{Rang}(A) = \text{Rang}(A|b)$  dann ist das lineare Gleichungssystem lösbar ( $b \in \text{Span}$  der Spalten von  $A$ ), aber im Allgemeinen nicht eindeutig lösbar.

Ist  $\text{Rang}(A) < \text{Rang}(A|b)$ , dann ist das lineare Gleichungssystem nicht klassisch lösbar. Man kann aber versuchen, den "Defekt"  $d := Ax - b$  zu minimieren, z.B. bezüglich der euklidischen Norm mit der "Methode der kleinsten Fehlerquadrate"

$$\|d\|_2 := \sqrt{\sum_{i=1}^n d_i^2}$$

**Satz 4.2.12** ("Least-squares-Lösung").  $A \in \mathbb{R}^{n \times m}$ ,  $b \in \mathbb{R}^n$  gegeben. Dann existiert immer eine Lösung  $\bar{x} \in \mathbb{R}^n$  mit kleinstem Fehlerquadrat

$$\|A\bar{x} - b\|_2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2$$

Dies ist äquivalent dazu, dass  $\bar{x} \in \mathbb{R}^n$  eine Lösung der "Normalgleichung"

$$A^T A \bar{x} = A^T b$$

ist. Ist  $\text{Rang}(A) = n$ , dann ist  $\bar{x}$  eindeutig bestimmt, ansonsten ist mit  $\bar{x}$  auch für jedes  $y \in \ker(A)$   $\bar{x} + y$  eine Lösung, und dies beschreibt alle Lösungen.

**Satz 4.2.13** (Least-Squares Lösungen). Sei  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ . Dann existiert immer ein  $\tilde{x} \in \mathbb{R}^n$  mit

$$\|A\tilde{x} - b\|^2 = \min_{x \in \mathbb{R}^n} \|Ax - b\|^2$$

und  $\tilde{x}$  ist Lösung der Normalgleichung

$$A^T A \tilde{x} = A^T b$$

$\tilde{x}$  ist eindeutig, wenn  $\text{Rang}(A) = n$

**Beweis** Sei  $\tilde{x}$  Lösung von  $A^T A \tilde{x} = A^T b$ . Wir wissen aus

**Analysis:** Minimum einer stetigen Funktion,

$$\min_i |x_i| \rightarrow \infty \implies \underbrace{\|Ax - b\|_2^2}_{:=F(x)} \rightarrow \infty \quad \text{Außer } A = 0$$

also

$$\forall M > 0 \exists r_0 : F(x) \geq M \forall x : \min |x_i| \geq r_0$$

dementsprechend nimmt  $F(x)$  auf der kompletten Menge  $B_{r_0}(0)$  ihr Minimum an.

**Lineare Algebra:** Es gelten die orthogonalen Zerlegungen:

$$\mathbb{R}^m = \text{Im}$$

□

$$Ax = b \implies (A + \delta A)(x + \delta x) = (b + \delta b)$$

zur Lösbarkeit des gestörten Systems

**Lemma 4.2.14.** Eine Matrix  $B \in \mathbb{R}^{n \times m}$  habe Norm  $\|B\| < 1$ . Dann ist  $I + B$  regulär und es gilt

$$\|(I + B)^{-1}\| \leq \frac{1}{1 - \|B\|}$$

**Beweis**

$$\forall x \in \mathbb{R}^n : x = Ix = (I + B)x - Bx$$

also

$$\|x\| \leq \|(I + B)x\| + \|Bx\|$$

und

$$\forall x \neq 0 : \|(I + B)x\| \geq \|x\| - \|Bx\| \geq \|x\| - \|B\| \cdot \|x\| = \underbrace{(1 - \|B\|)}_{>0} \|x\| > 0$$

Also ist  $\ker(I + B) = \{0\}$ ,  $I + B$  regulär. Weiter ist

$$\begin{aligned} 1 &= \|I\| = \|(I + B)(I + B)^{-1}\| \\ &= \|(I + B)^{-1} + B(I + B)^{-1}\| \\ &\geq \|(I + B)^{-1}\| - \|B\| \|(I + B)^{-1}\| \\ &= (1 - \|B\|) \|(I + B)^{-1}\| \end{aligned}$$

□

**Beweis** Also

$$\begin{aligned} \|\delta x\| &\leq \|(A + \delta A)^{-1}\| (\|\delta b\| + \|\delta A\| \|x\|) \\ &= \|(A(I + A^{-1}\delta A))^{-1}\| \\ &= \|(I + A^{-1}\delta A)^{-1} A^{-1}\| \\ &\leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| (\|\delta b\| + \|\delta A\| \|x\|) \\ &\leq \frac{1}{1 - \|A^{-1}\delta A\|} \|A^{-1}\| (\|\delta b\| + \|\delta A\| \|x\|) \\ &\leq \frac{1}{1 - \|A^{-1}\| \|\delta A\| \frac{\|A\|}{\|A\|}} \cdot \frac{\|A\| \|A^{-1}\|}{\|A\|} \|x\| \left( \frac{\|\delta b\|}{\|x\|} + \|\delta A\| \right) \\ &\leq \frac{1}{1 - \kappa(A) \cdot \frac{\|\delta A\|}{\|A\|}} \cdot \kappa(A) \cdot \left( \frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \cdot \|x\| \end{aligned}$$

□

**Bemerkung 4.2.15.** Ist  $\kappa(A) \frac{\|\delta A\|}{\|A\|} \leq 1$ , dann gilt im Wesentlichen

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(A) \cdot \left( \frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right)$$

das heißt Fehler in den Daten werden im Wesentlichen durch die Kondition der Matrix verstärkt. Dies kann groß sein:

$$\text{Kond} \left[ \begin{pmatrix} 2 & -1 & 0 \\ -1 & \ddots & \ddots \\ 0 & \ddots & \ddots \end{pmatrix} \right] = \mathcal{O} \left( \frac{1}{h^2} \right) = \mathcal{O} (n^2)$$