

# Mitschrift Numerik 1

Prof. Schaedle

July 22, 2019

# Inhaltsverzeichnis

<b>I Numerische Integration</b>	<b>4</b>
1 Einführung . . . . .	4
2 Ordnung von Quadraturformeln . . . . .	7
3 Quadraturfehler . . . . .	11
4 Quadratur mit hoher Ordnung . . . . .	16
5 Orthogonalpolynome . . . . .	19
6 Ein adaptives Programm . . . . .	23
7 Gauß- und Lobatto Quadraturformeln . . . . .	27
<b>II Interpolation und Approximation</b>	<b>28</b>
8 Newtonsche Interpolationsformel . . . . .	29
9 Fehler bei der Polynominterpolation . . . . .	33
10 Tschebyscheff-Interpolation . . . . .	37
11 Hermité-Interpolation . . . . .	46
12 Spline-Interpolation . . . . .	48
13 Fehler bei der Splineinterpolation . . . . .	53
14 Numerische Differentiation . . . . .	57
<b>III Lineare Gleichungssysteme und lineare Ausgleichsrechnung</b>	<b>60</b>
15 Gaußelimination . . . . .	60
16 Wahl des Pivotelements . . . . .	67
17 Cholesky-Zerlegung für symmetrische positiv definite Matrizen	69
18 Matrixnormen . . . . .	72
19 Kondition eines Problems . . . . .	75
20 Konditionszahl einer Matrix . . . . .	76
21 Stabilität von Verfahren . . . . .	80
22 QR-Zerlegung mit Hilfe der Householdertransformationen . .	82
23 Lineare Ausgleichsrechnung . . . . .	87
<b>IV Nichtlineare Gleichungssysteme</b>	<b>91</b>
24 Newton-Verfahren . . . . .	92
<b>V Gewöhnliche Differentialgleichungen</b>	<b>96</b>
25 Beispiele für gewöhnliche Differentialgleichungen . . . . .	97
26 Erinnerung an die Theorie gewöhnlicher DGLs . . . . .	98
27 Euler-Verfahren . . . . .	101

28	Runge-Kutta Verfahren . . . . .	104
----	---------------------------------	-----

# I Numerische Integration

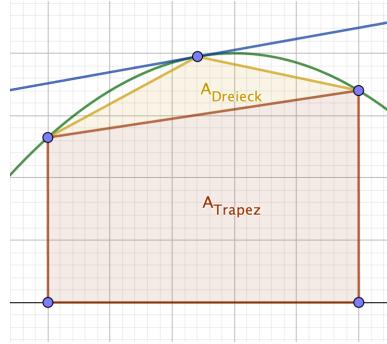
## 1 Einführung

### Problem 1.1

Gegeben  $f : [a, b] \rightarrow \mathbb{R}$  mit  $a, b \in \mathbb{R}$ . Berechne  $\int_a^b f(x)dx$

### Beispiel 1.2

1. Archimedes (282-212 v.Chr.): Fläche unter einer Parabel



$$A_{Parabel} = A_{Trapez} + \frac{4}{3}A_{Dreieck}$$

2. Leibniz + Newton (~1670):

$$\int_a^b f(x)dx = F(b) - F(a),$$

$$\text{wobei } \frac{d}{dx}F(x) = f(x)$$

3. Riemann (~1850):

$$\int_a^b f(x)dx = \lim_{|\Delta| \rightarrow 0} \sum_{j=1}^n f(\xi_j)(x_j - x_{j-1}),$$

wobei  $\Delta = (x_0, \dots, x_n)$  Gitter Zerlegung von  $[a, b]$ ,  $a = x_0 < \dots < x_n = b$ ,  $\xi_j \in [x_{j-1}, x_j]$  und  $|\Delta| := \max_{j=1, \dots, n} |x_j - x_{j-1}|$ . Das Riemannintegral existiert, falls:

$$\forall \varepsilon > 0 \exists \delta > 0 : |\Delta| < \delta \Rightarrow \left| \int_a^b f(X)dx - \sum_{j=1}^n f(\xi_j)(x_j - x_{j-1}) \right| < \varepsilon$$

### Bemerkung 1.3 (Approximation von Integralen)

1. (linke) Rechtecksregel:

$$\int_{x_{j-1}}^{x_{j-1}+h} f(x)dx \approx h f(x_{j-1})$$

$$\int_a^b f(x)dx = \sum_{j=1}^n \int_{x_{j-1}}^{x_j} f(x)dx \approx \sum_{j=1}^n f(x_{j-1})(x_j - x_{j-1})$$

2. Mittelpunktsregel:

$$\int_{x_j}^{x_j+h} f(x)dx \approx f\left(\frac{x_j + x_{j-1} + h}{2}\right)h$$

$$\int_a^b f(x)dx \approx \sum_{j=1}^n f\left(\frac{x_{j-1} + x_j}{2}\right)(x_j - x_{j-1})$$

Da mit Hilfe der Transformationsformel sich jedes Integral  $\int_{x_{j-1}}^{x_j}$  auf ein Integral  $\int_a^b$  transformieren lässt, betrachten wir ohne Einschränkungen Integrale von 0 bis 1. Nutze dazu die Abb.  $[a, b] \rightarrow [x_{j-1}, x_j], t \mapsto x_{j-1} + t(x_j - x_{j-1})$ .

$$\int_{x_{j-1}}^{x_j} f(x)dx = \int_0^1 \underbrace{f(x_{j-1} + t(x_j - x_{j-1}))}_{:=g_{j-1}(t)}(x_j - x_{j-1})dt = \int_0^1 g_{j-1}(t)(x_j - x_{j-1})dt$$

### Definition 1.4 (Quadraturformel)

Eine s-stufige Quadraturformel zur Approximation von  $\int_0^1 g(t)dt$  mit Knoten  $c_i$  und Gewichten  $b_i$  für  $i = 1, \dots, s$  ist gegeben durch

$$\sum_{i=1}^s b_i g(c_i) \left( \approx \int_0^1 g(t)dt \right)$$

### Beispiel 1.5

1. Rechtecksregel:  $s = 1, b_1 = 1, c_1 = 0$

$$\int_0^1 g(t) \approx b_1 g(c_1) = g(0)$$

2. Mittelpunktsregel:  $s = 1, b_1 = 1, c_1 = \frac{1}{2}$

$$\int_0^1 g(t) \approx g\left(\frac{1}{2}\right)$$

3. Trapezregel:  $s = 2, b_1 = b_2 = \frac{1}{2}, c_1 = 0, c_2 = 1$

$$\int_0^1 g(t) \approx \frac{1}{2}g(0) + \frac{1}{2}g(1)$$

4. Simpsonregel:  $s = 3, b_1 = \frac{1}{6}, b_2 = \frac{2}{3}, b_3 = \frac{1}{6}, c_1 = 0, c_2 = \frac{1}{2}, c_3 = 1$

$$\int_0^1 g(t) \approx \frac{1}{6} \left( g(0) + 4g\left(\frac{1}{2}\right) + g(1) \right)$$

**Herleitung:** Man legt eine Parabel  $p$  durch die Punkte  $(0, g(0)), (\frac{1}{2}, g(\frac{1}{2})), (1, g(1))$  und integriert  $p$  von 0 bis 1.

$$p(t) = g(0)(1-t)2(\frac{1}{2}-t) + g(\frac{1}{2})(1-t)4t + g(1)(\frac{1}{2}-t)2t$$

$$\Rightarrow \int_0^1 p(t)dt = \frac{1}{6}g(0) + \frac{2}{3}g\left(\frac{1}{2}\right) + \frac{1}{6}g(1)$$

5. "pulcherrima et utilissima regula" von Newton:

$$\int_0^1 g(t)dt \approx \frac{1}{8} \left( g(0) + 3g\left(\frac{1}{3}\right) + 3g\left(\frac{2}{3}\right) + g(1) \right)$$

**Bemerkung 1.6** (Monte-Carlo Integration)

1. Eindimensionale Monte-Carlo Integration:

Sei  $a, b \in \mathbb{R}$ ,  $a < b$ . Wählt man  $N$  unabhängige gleichverteilte Punkte  $x_i$  in  $[a, b]$  so gilt die Approximation:

$$\int_a^b f(x)dx \approx \frac{1}{N} \sum_{j=1}^N (b-a)f(x_j)$$

Nach dem Gesetz der großen Zahlen konvergiert dieser Ausdruck, falls

$$\int_a^b |f(x)|dx < \infty, \int_a^b f^2(x)dx < \infty$$

2. Mehrdimensionale Monte-Carlo Integration:

Sei  $W = \otimes_{i=1}^d [a_i, b_i]$  ein d-dimensionalen Quader. Wählt man in  $W$  unabh. gleichwert. Zufallsvektoren  $x_i$  in  $W$ , so ist

$$\int_W f(x)dx \approx \frac{1}{N} Vol(W) \sum_{i=1}^N f(x_i),$$

wobei  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ .

**Achtung:** Dieses gewöhnliche MC-Verfahren konvergiert sehr langsam. Verbesserungen sind z.B.: Importance sampling, Control variates, Antithetic variates und stratified sampling.

## 2 Ordnung von Quadraturformeln

### Definition 2.1

Eine Quadraturformel (QF) mit Gewichten und Knoten  $(b_i, c_i)_{i=1}^s$  hat **Ordnung  $p$** , falls sie exakt ist für alle Polynome von Grad  $\leq p - 1$ .

$$\mathcal{P} := \left\{ \sum_{i=0}^n a_i X^i, a_i \in \mathbb{R}(\mathbb{C}) \right\}, \quad \text{Menge aller Polynome}$$

Für  $q \in \mathcal{P}$  ist  $\deg(q)$  der Grad des Polynoms.

### Satz 2.2

Ein QF  $(b_i, c_i)_{i=1}^s$  für  $[0, 1]$  hat Ordnung  $p$  genau dann, wenn

$$\sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}$$

für  $q = 1, \dots, p$ .

*Beweis.*

”  $\Rightarrow$  ”

QF hat Ordnung  $p \Rightarrow$  QF ist exakt für  $g(t) = t^{q-1}$  für  $q = 1, \dots, p$  auf  $[0, 1]$

$\Rightarrow$

$$\sum b_i c_i^{q-1} = \int_0^1 t^{q-1} dt = \left[ \frac{t^q}{q} \right]_{t=0}^1 = \frac{1}{q}$$

”  $\Leftarrow$  ”

Jedes Polynom von Grad  $p - 1$  lässt sich als Linearkombination von  $1, t, t^2, \dots, t^{p-1}$ . Die Behauptung folgt aus der Linearität in  $g$  von

$$\int_0^1 g(t) dt$$

und

$$\sum_{i=1}^s b_i g(c_i)$$

□

### Beispiel 2.3

1. Rechtecksregel:  $p = 1$
2. Mittelpunktsregel:  $p = 2$
3. Trapezregel:  $p = 2$
4. Simpsonregel:  $p \geq 3$  nach Konstruktion

$$q = 4 : \quad \frac{1}{6} * 0^3 + \frac{4}{6} * \left(\frac{1}{2}\right)^3 + \frac{1}{6} * 1^3 = \frac{1}{4}$$

$$q = 5 : \quad \frac{1}{6} * 0^4 + \frac{4}{6} * \left(\frac{1}{2}\right)^4 + \frac{1}{6} * 1^4 = \frac{5}{24} \neq \frac{1}{5}$$

Nach Satz (2.2) ist damit die Ordnung der Simpsonregel  $p = 4$ .

5. ”pulcherina et utilissima”: Übung

### Bemerkung 2.4

Zu vergebenen paarweise verschiedenen Knoten  $c_1, \dots, c_s$  lässt sich mit Satz (2.2) für  $p = s$  ein lineares Gleichungssystem für die Gewichte  $b_1, \dots, b_s$  aufstellen.

$$\underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ c_1 & c_2 & \dots & c_s \\ \vdots & \vdots & & \vdots \\ c_1^{s-1} & c_2^{s-1} & \dots & c_s^{s-1} \end{bmatrix}}_{=V} * \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ \vdots \\ 1/s \end{bmatrix}$$

Falls die Vandermonde-Matrix V invertierbar ist, so lassen sich die Gewichte  $b_1, \dots, b_s$  bestimmen, sodass die QF  $(b_i, c_i)_{i=1}^s$  mindestens Ordnung  $s$  hat.

### Definition 2.5

Eine QF heißt symmetrisch, falls für  $i = 1, \dots, s$  gilt:

1.  $c_i = 1 - c_{s+1-i}$
2.  $b_i = b_{s+1-i}$

### Beispiel 2.6 (Symmetrische QF)

Mittelpunktsregel, Trapezregel, Simpsonregel,...

### Satz 2.7

Die maximal erreichbare Ordnung einer symmetrischen QF ist gerade.

*Beweis.* Sei die QF  $(b_i, c_i)_{i=1}^s$  exakt für Polynome vom Grad  $\leq 2m - 2$  (für  $m \in \mathbb{N}$ ), (dann ist die Ordnung  $\geq 2m - 1$ ).

$$\forall g \in \mathcal{P} : \deg(g) \leq 2m - 2 \Rightarrow \sum_{i=1}^s b_i g(c_i) = \int_0^1 g(t) dt$$

Sei  $f \in \mathcal{P}$  mit  $\deg(f) = 2m - 1$ .

Wir zeigen QF ist exakt für  $f$ .

$$f(t) = ct^{2m-1} + g(t)$$

für  $g \in \mathcal{P}$  mit  $\deg(g) \leq 2m - 2$  mit  $c \neq 0$ .

Trick:  $f(t) = c(t - \frac{1}{2})^{2m-1} + \tilde{g}(t)$  mit  $\tilde{g} \in \mathcal{P}$  und  $\deg(\tilde{g}) \leq 2m - 2$

1. Für  $\tilde{g}$  ist die QF exakt

2.

$$\int_0^1 \left( t - \frac{1}{2} \right)^{2m-1} dt = \left[ \frac{1}{2m-2} \left( t - \frac{1}{2} \right)^{2m-2} \right]_0^1 = 0$$

$$\sum_{i=1}^s b_i \left( c_i - \frac{1}{2} \right)^{2m-1}$$

Symmetrie  $\Rightarrow$

$$= \sum_{i=1}^s b_{s+1-i} \left( \frac{1}{2} - c_{s+1-i} \right)^{2m-1}$$

Definiere  $j := s + 1 - i$

$$= \sum_{i=1}^s b_i \left( \frac{1}{2} - c_i \right)^{2m-1} = - \sum_{i=1}^s b_i \left( c_i - \frac{1}{2} \right)^{2m-1}$$

$$\Rightarrow 2 * \sum_{i=1}^s b_i \left( c_i - \frac{1}{2} \right)^{2m-1} = 0$$

$$\Rightarrow \sum_{i=1}^s b_i \left( c_i - \frac{1}{2} \right)^{2m-1} = 0$$

$$\sum_{i=1}^s b_i f(c_i) = c \sum_{i=1}^s b_i \left( c_i - \frac{1}{2} \right)^{2m-1} + \sum_{i=1}^s b_i \tilde{g}(c_i)$$

$$= c \int_0^1 \left( t - \frac{1}{2} \right)^{2m-1} dt + \int_0^1 \tilde{g}(t) dt = \int_0^1 f(t) dt$$

$\Rightarrow$  QF hat mind. Ordnung  $2m$ .

□

### Satz 2.8

Sind Knoten  $c_1 < c_2 < \dots < c_s$  ( $c_i \in \mathbb{R}, i = 1, \dots, s$ ) gegeben, so existieren eindeutig bestimmte Gewichte  $b_1, \dots, b_s$  derart, dass die QF  $(b_i, c_i)_{i=1}^s$  die maximale Ordnung  $p \geq s$  hat.

Es gilt

$$b_i = \int_0^1 l_i(t) dt$$

mit

$$l_i(t) = \frac{\prod_{j=1, j \neq i}^s (t - c_j)}{\prod_{j=1, j \neq i}^s (c_i - c_j)}$$

*Beweis.*

1. Hat die QF die Ordnung  $p \geq s$ , so ist wegen  $\deg(l_i) = s - 1$ :

$$\int_0^1 l_i(t) dt = \sum_{j=1}^s b_j l_i(c_j) = b_i$$

2. Zu den Knoten  $c_1, \dots, c_s$  definiere  $b_i$  wie angegeben. Die QF ist dann exakt für alle Polynome von Grad  $\leq s - 1$ , da die  $l_1, \dots, l_s$  linear unabhängig sind und eine Basis des Vektorraums der Polynome von Grad  $\leq s - 1$  bilden.

□

**Bemerkung** (zu Satz (2.8))

$l_i$  ist das  $i$ -te Lagrange-Polynom zu den Knoten  $c_1, \dots, c_s$ . Es gilt:

1.  $\deg(l_i) = s - 1$
2.  $l_i(c_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$ , für  $j = 1, \dots, s$

### 3 Quadraturfehler

Allgemeine Voraussetzung:  $f : [a, b] \rightarrow \mathbb{R}$  sei hinreichend oft differenzierbar  
( $f$  ist eine glatte Funktion)

#### Definition 3.1

Der Fehler bei der Approximation des Integrals durch die QF ist

$$err = \int_a^b f(x) dx - \sum_{j=0}^{n-1} \left( h_{j+1} \sum_{i=1}^s b_i f(x_j + h_{j+1} c_i) \right)$$

mit  $h_{j+1} = x_{j+1} - x_j$

$$\begin{aligned} &= \sum_{j=0}^{n-1} \left( \int_{x_j}^{x_{j+1}} f(x_j + \tau) d\tau - h_{j+1} \sum_{i=1}^s b_i f(x_j + h_{j+1} c_i) \right) \\ &= \sum_{j=0}^{n-1} h_{j+1} \left( \int_0^1 g_j(\xi) d\xi - \sum_{i=1}^s b_i g_j(c_i) \right) \end{aligned}$$

mit  $g_j(\xi) = f(x_j + \xi h_{j+1})$

Der Quadraturfehler auf Teilintervallen  $[x_j, x_j + h_{j+1}]$  ist

$$\begin{aligned} E(f, x_j, h_{j+1}) &= \int_{x_j}^{x_{j+1}} f(x) dx - h_{j+1} \sum_{i=1}^s b_i f(x_j + c_i h_{j+1}) \\ &= h_{j+1} \left( \int_0^1 g_j(\xi) d\xi - \sum_{i=1}^s b_i g_j(c_i) \right) \end{aligned}$$

### 3.2 (Fehlerabschätzung - 1. Versuch)

Falls  $f$  auf  $[x_0, x_0 + h]$  glatt genug ist und die QF Ordnung  $p$  hat, aber nicht Ordnung  $p + 1$ , so erhält man durch Taylorentwicklung um  $x_0$  von  $f(x_0 + \xi h) = g_0(\xi)$  und  $f(x_0 + c_i h)$ :

$$\begin{aligned} E(f, x_0, h) &= \sum_{k \geq 0} \frac{h^{k+1}}{k!} \left( \int_0^1 t^k dt - \sum_{i=1}^s b_i c_i^k \right) f^{(k)}(x_0) \\ &= \frac{h^{p+1}}{p!} \left( \frac{1}{p+1} - \sum_{i=1}^s b_i c_i^p \right) f^{(p)}(x_0) + \underbrace{\mathcal{O}(h^{p+2})}_{\text{Taylorrestglied}} \end{aligned}$$

Die Konstante  $C = \frac{1}{p!} \left( \frac{1}{p+1} - \sum_{i=1}^s b_i c_i^p \right)$  heißt Fehlerkonstante.

Ist  $h$  klein genug, sodass das Taylorrestglied im Vergleich zu  $h^{p+1} C f^{(p)}(x_0)$  vernachlässigbar ist, so gilt:

$$err = \sum_{j=0}^{n-1} E(f, x_j, h)$$

mit  $x_j = x_0 + jh$

$$\begin{aligned} &\approx Ch^p \sum_{j=0}^{n-1} h f^{(p)}(x_j) \\ &\approx Ch^p \int_a^b f^{(p)}(x) dx \\ &= Ch^p (f^{(p-1)}(b) - f^{(p-1)}(a)) \end{aligned}$$

### 3.3 (Rigorose Fehlerabschätzung)

**Satz 1:** Sei  $f : [a, b] \rightarrow \mathbb{R}$   $k$ -mal stetig differenzierbar ( $f \in C^k([a, b])$ ) und habe die QF Ordnung  $p$ , so gilt für  $h < b - a$  und  $k \leq p$

$$E(f, x_0, h) = h^{k+1} \int_0^1 K_k(\tau) f^{(k)}(x_0 + \tau k) d\tau,$$

wobei der Peanokern  $K_k(\tau)$  durch

$$K_k(\tau) := \frac{(1-\tau)^k}{k!} - \sum_{i=1}^s b_i \frac{(c_i - \tau)_+^{k-1}}{(k-1)!},$$

$$\text{mit } (\sigma)_+^{k-1} = \begin{cases} \sigma^{k-1} & \sigma > 0 \\ 0 & \text{sonst} \end{cases}, \text{ gegeben ist.}$$

*Beweis.* Taylorentwicklung mit Integralrestglied und Transformation

$$f(x_0 + th) = \sum_{j=0}^{k-1} \frac{(th)^j}{j!} f^{(j)}(x_0) + h^k \int_0^t \frac{(t-\tau)^{k-1}}{(k-1)!} f^{(k)}(x_0 + \tau h) d\tau$$

eingesetzt in (\*) und die Verwendung von

$$\int_0^{c_i} (c_i - \tau)^{k-1} g(\tau) d\tau = \int_0^1 (c_i - \tau)_+^{k-1} g(\tau) d\tau$$

liefern

$$\begin{aligned}
E(f, x_0, h) &= h \int_0^1 \left( \sum_{j=0}^{k-1} \frac{(th)^j}{j!} f^{(j)}(x_0) + h^k \int_0^t \frac{(t-\tau)^{k-1}}{(k-1)!} f^{(k)}(x_0 + \tau h) d\tau \right) dt \\
&\quad - h \sum_{i=1}^s b_i \left( \sum_{j=0}^{k-1} \frac{(c_i h)^j}{j!} f^{(j)}(x_0) + h^k \int_0^{c_i} \frac{(c_i - \tau)^{k-1}}{(k-1)!} f^{(k)}(x_0 + c_i h) d\tau \right) \\
&= hh^k \left( \int_0^1 \int_0^t \frac{(t-\tau)^{k-1}}{(k-1)!} f^{(k)}(x_0 + \tau h) d\tau dt \right) \\
&\quad - hh^k \left( \sum_{i=1}^s \int_0^1 \frac{(c_i - \tau)_+^{k-1}}{(k-1)!} f^{(k)}(x_0 + \tau h) d\tau \right) \\
&= hh^k \left( \int_0^1 \int_0^1 \frac{(t-\tau)_+^{k-1}}{(k-1)!} f^{(k)}(x_0 + \tau h) d\tau dt \right) \\
&\quad - hh^k \left( \sum_{i=1}^s b_i \int_0^1 \frac{(c_i - \tau)_+^{k-1}}{(k-1)!} f^{(k)}(x_0 + \tau h) d\tau \right) \\
&= h^{k+1} \int_0^1 \left( \int_0^1 \frac{(t-\tau)_+^{k-1}}{(k-1)!} dt - \frac{(c_i - \tau)_+^{k-1}}{(k-1)!} \right) f^{(k)}(x_0 + \tau h) d\tau \\
&= h^{k+1} \int_0^1 K_k(\tau) f^{(k)}(x_0 + \tau h) d\tau,
\end{aligned}$$

da

$$\int_0^1 \frac{(t-\tau)_+^{k-1}}{(k-1)!} dt = \int_0^1 \frac{(t-\tau)^{k-1}}{(k-1)!} = \left[ \frac{1}{k!} (t-\tau)^k \right]_{t=\tau}^1 = \frac{1}{k!} (1-\tau)^k$$

gilt. □

**Satz 2:** (Eigenschaften des Peanokerns)

Für eine QF der Ordnung  $p$  gilt für  $k \leq p$  ( $k, p \in \mathbb{N}$ )

1.  $K'_k(\tau) = -K_{k-1}(\tau)$  für  $k \geq 2$  und  $\tau \neq c_i$  falls  $k = 2$
2.  $K_k(1) = 0$  für  $k \geq 1$ , falls  $c_i \leq 1$  für  $i = 1, \dots, s$
3.  $K_k(0) = 0$  für  $k \geq 2$ , falls  $c_i \leq 1$  für  $i = 1, \dots, s$

4.  $\int_0^1 K_p(\tau) = \frac{1}{p!} \left( \frac{1}{p-1} - \sum_{i=1}^s b_i c_i^p \right) =: C$  (Fehlerkonstante  $C$  aus (3.2))
5.  $K_1(\tau)$  ist stückweise linear mit Steigung  $-1$  und Sprüngen der Höhe  $b_i$  an den Stellen  $c_i$

*Beweis.* Eventuell Übungsaufgabe □

**Beispiel:** Mittelpunktsregel:

$$\begin{aligned} K_1(\tau) &= \frac{(1-\tau)^1}{1!} - 1 \frac{\left(\frac{1}{2}-\tau\right)_+^1}{0!} \\ &= 1 - \tau - \left(\frac{1}{2}-\tau\right)_+^0 \\ &= \begin{cases} 1 - \tau & \tau < \frac{1}{2} \\ 1 - \tau & \tau \geq \frac{1}{2} \end{cases} \end{aligned}$$

$$\begin{aligned} K_2(\tau) &= \frac{(1-\tau)^2}{2!} - 1 \frac{\left(\frac{1}{2}-\tau\right)_+^1}{1!} \\ &= \frac{1}{2}(1-\tau)^2 - \left(\frac{1}{2}-\tau\right)_+^1 \\ &= \begin{cases} \frac{\tau^2}{2} & \tau < \frac{1}{2} \\ \frac{1}{2}(1-\tau)^2 & \tau \geq \frac{1}{2} \end{cases} \end{aligned}$$

**Satz 3:** Sei  $f \in C^k([a, b])$  und habe die QF  $(b_i, c_i)_{i=1}^s$ , Ordnung  $p \geq k$ , so gilt für den Fehler  $err$  aus (3.1)

$$|err| \leq h^k(b-a) \int_0^1 |K_k(\tau)| d\tau \max_{x \in [a,b]} |f^{(k)}(x)|, \quad h = \max_{j=1,\dots,n} h_j.$$

*Beweis.* Mit Satz 1 gilt

$$\begin{aligned} |E(f, x_j, h_{j+1})| &\leq h_{j+1}^{k+1} \int_0^1 |K_k(\tau)| |f^{(k)}(x_j + \tau h_{j+1})| d\tau \\ &\leq h_{j+1}^{k+1} \int_0^1 |K_k(\tau)| d\tau \max_{x \in [x_j, x_j + h_{j+1}]} |f^{(k)}(x)| \end{aligned}$$

Zudem gilt

$$\begin{aligned}
|err| &= \left| \sum_{j=0}^{n-1} E(f, x_j, h_{j+1}) \right| \\
&\leq \sum_{j=0}^{n-1} |E(f, x_j, h_{j+1})| \\
&\leq \underbrace{\sum_{j=0}^{n-1} h_{j+1}}_{b-a} \underbrace{h_{j+1}^k \int_0^1 |K_k(\tau)| d\tau}_{\leq h^k} \underbrace{\max_{x \in [x_j, x_{j+1}]} |f^{(k)}(x)|}_{\leq \max_{x \in [a, b]} |f^{(k)}(x)|}
\end{aligned}$$

Damit folgt die Behauptung.  $\square$

### Beispiele

Für die Mittelpunktsregel (maximale Ordnung = 2) erhält man

$$|err| \leq h^2(b-a) \frac{1}{24} \max_{x \in [a,b]} |f^{(2)}(x)|$$

Für die Trapezregel (maximale Ordnung = 2)

$$|err| \leq h^2(b-a) \frac{1}{12} \max_{x \in [a,b]} |f^{(2)}(x)|$$

Für die Simpsonregel (maximale Ordnung = 4)

$$|err| \leq h^4(b-a) \frac{1}{2880} \max_{x \in [a,b]} |f^{(4)}(x)|$$

$\rightarrow$  Der Fehler wird klein, falls  $h$  klein und die Ordnung  $p$  groß wird.

## 4 Quadratur mit hoher Ordnung

Es seien Knoten  $c_1 < \dots < c_s$  gegeben. Aus §2 wissen wir:

Es gibt Gewichte  $b_1, \dots, b_s$ , sodass  $p \geq s$ .

Fragen:

- Kann man  $c_j$  so wählen, dass  $p > s$ ?

- Wenn ja, wie?
- Wie groß kann  $p$  maximal werden?

Ziel: QF mit Ordnung  $p = s + m$  für  $m \in \mathbb{N}, m > 1$ . Sei  $g \in \mathcal{P}_{s+m-1}$  (Polynome von Grad  $\leq s + m - 1$ ).

$g$  soll durch die QF exakt integriert werden.

Idee: Dividiere  $g$  durch  $M(t) = \prod_{i=1}^s (t - c_i)$  "Knotenpolynom"

$$\deg(M) = s$$

$$g(t) = M(t)h(t) + r(t) \text{ mit Rest } r, \deg(r) \leq s - 1 \text{ und } \deg(h) \leq m - 1$$

Dann gilt einerseits

$$\int_0^1 g(t)dt = \int_0^1 M(t)h(t)dt + \int_0^1 r(t)dt$$

und andererseits

$$\begin{aligned} \sum_{i=1}^s b_i g(c_i) &= \sum_{i=1}^s b_i \underbrace{M(c_i)}_{=0} h(c_i) + \sum_{i=1}^s b_i r(c_i) \\ &= 0 + \int_0^1 r(t)dt, \end{aligned}$$

da  $p \leq s$

Damit ist gezeigt:

#### Satz 4.1

Sei  $(b_i, c_i)_{i=1}^s$  der Ordnung  $p \geq s$ . Äquivalent sind:

1. QF hat Ordnung  $s + m$
2.  $\forall h \in \mathcal{P}_{m-1} : \int_0^1 M(t)h(t)dt = 0$

#### Korollar 4.2

Die Ordnung einer  $s$ -stufigen QF ist höchstens  $2s$ .

*Beweis (indirekt).* Annahme:  $p > 2s$

$$(4.1) \Rightarrow \forall h \in \mathcal{P}_s : \int_0^1 M(t)h(t)dt = 0$$

Setze  $h = M$ , dann ist

$$\int_0^1 M(t)^2 dt = 0$$

↳ zu  $\int_0^1 M(t)^2 dt > 0$ , da  $M(t) \equiv 0$

□

### 4.3 (Beispiele/Korollare)

1. Jede 3-stufige QF mit Ordnung  $\geq 4$  muss

$$\begin{aligned} & \int_0^1 (t - c_1)(t - c_2)(t - c_3) dt = 0 \\ \Leftrightarrow & \int_0^1 t^3 + t^2(-c_1 - c_2 - c_3) + t(c_1c_2 + c_2c_3 + c_1c_3) - c_1c_2c_3 dt \\ = & \frac{1}{4} + \frac{1}{3}(-c_1 - c_2 - c_3) + \frac{1}{2}(c_1c_2 + c_2c_3 + c_1c_3) - c_1c_2c_3 = 0 \end{aligned}$$

erfüllen, dh.

$$c_3 = \frac{\frac{1}{4} - (c_1 + c_2)\frac{1}{3} + c_1c_2\frac{1}{2}}{\frac{1}{3} - (c_2 + c_1)\frac{1}{2} + c_1c_2}$$

2. Zur Berechnung der Knoten einer 3-stufigen QF der Ordnung 6 verwenden wir (4.2) mit  $h(t) = 1, t, t^2$

$$\int_0^1 M(t)h(t) dt = 0$$

$$\begin{aligned} h(t) = 1 & \Rightarrow c_1c_2c_3 - \frac{1}{2}(c_1c_2 + c_2c_3 + c_1c_3) + \frac{1}{3}(c_1 + c_2 + c_3) = \frac{1}{4} \\ h(t) = t & \Rightarrow \frac{1}{2}c_1c_2c_3 - \frac{1}{3}(c_1c_2 + c_2c_3 + c_1c_3) + \frac{1}{4}(c_1 + c_2 + c_3) = \frac{1}{5} \\ h(t) = t^2 & \Rightarrow \frac{1}{3}c_1c_2c_3 - \frac{1}{4}(c_1c_2 + c_2c_3 + c_1c_3) + \frac{1}{5}(c_1 + c_2 + c_3) = \frac{1}{6} \end{aligned}$$

Das ist ein nichtlineares Gleichungssystem in  $c_1, c_2, c_3$ .

Trick:

$$\sigma_1 = c_1 + c_2 + c_3$$

$$\sigma_2 = c_1c_2 + c_1c_3 + c_2c_3$$

$$\sigma_3 = c_1c_2c_3$$

Das sind die Koeffizienten von  $M(t)$  in der Monombasis.

$$M(t) = (t - c_1)(t - c_2)(t - c_3) = t^3 - \sigma_1t^2 + \sigma_2t - \sigma_3$$

und das Gleichungssystem ist linear in  $\sigma_1, \sigma_2, \sigma_3$   
mit Lösung  $\sigma_1 = \frac{3}{2}, \sigma_2 = \frac{3}{5}, \sigma_3 = \frac{1}{20}$   
und damit ist

$$\begin{aligned} M(t) &= t^3 - \frac{3}{2}t^2 + \frac{3}{5}t - \frac{1}{20} \\ &= (t - \frac{1}{2})(t - \frac{5 - \sqrt{15}}{10})(t - \frac{5 + \sqrt{15}}{10}) \end{aligned}$$

Glücklicherweise sind die Wurzeln von  $M(t)$  in  $[0, 1]$ . Damit lassen sich die Gewichte mit (2.4) berechnen und wir erhalten

$$\int_0^1 g(t)dt = \frac{5}{18}g\left(\frac{5 - \sqrt{15}}{10}\right) + \frac{8}{18}g\left(\frac{1}{2}\right) + \frac{5}{18}g\left(\frac{5 + \sqrt{15}}{10}\right)$$

Ziel: Konstruktion von QF der Ordnung  $2s$  mit Hilfe von orthogonalen Polynomen.

## 5 Orthogonalpolynome

Bedingung 2. in Satz (4.1)

$$\forall h \in \mathcal{P}_{m-1} : \int_0^1 M(t)h(t)dt = 0$$

kann als Orthogonalitätsbedingung bzgl. eines Skalarprodukts  $\langle f, g \rangle = \int_0^1 f(t)g(t)dt$  auf dem Vektorraum  $L^2([0, 1])$  oder  $C([0, 1])$  aufgefasst werden.

Erinnerung:

$$\mathcal{P}_s := \left\{ \sum_{j=0}^s \alpha_j X^j, \alpha_j \in \mathbb{R} \right\}$$

ist ein  $\mathbb{R}$ -VR mit  $\dim(\mathcal{P}_s) = s + 1$  und Basis  $\{1, X, X^2, \dots, X^s\}$

$\langle \cdot, \cdot \rangle : C([0, 1]) \times C([0, 1]) \rightarrow \mathbb{R}, \langle f, g \rangle \mapsto \int_0^1 f(t)g(t)dt$  ist

1. symmetrisch  $\langle f, g \rangle = \langle g, f \rangle$
2. linear  $\langle \alpha f + g, h \rangle = \alpha \langle f, h \rangle + \langle g, h \rangle$

3. positiv definit  $\langle f, f \rangle \geq 0$  und  $\langle f, f \rangle = 0 \Rightarrow f = 0$

Wie in der linearen Algebra definieren wir  $f$  steht senkrecht auf  $g$ :  $f \perp g \Leftrightarrow \langle f, g \rangle = 0$

### Satz 5.1

QF hat die Ordnung  $s + m \Leftrightarrow M$  ist orthogonal auf allen Polynome in  $\mathcal{P}_{m-1}$

### Definition 5.2

Für eine Gewichtsfunktion  $\omega : (a, b) \rightarrow \mathbb{R}$  mit

1.  $\omega$  stetig
2.  $\forall x \in (a, b) : \omega(x) > 0$
3.  $\forall k \in \mathbb{N} : \int_a^b \omega(x)|x|^k dx < \infty$

definieren wir auf den Vektorraum

$$V = \left\{ f : [a, b] \rightarrow \mathbb{R} : f \text{ stetig und } \int_a^b f(x)^2 \omega(x) dx < \infty \right\}$$

das gewichtete Skalarprodukt

$$\langle f, g \rangle_\omega := \int_a^b \omega(x)f(x)g(x)dx$$

für  $f, g \in V$ .

Zudem definiere:

$$f \perp_\omega g \Leftrightarrow \langle f, g \rangle_\omega = 0$$

### Satz 5.3

Es existiert eine eindeutige Folge von Polynomen  $p_0, p_1, \dots$  mit

1.  $\deg(p_k) = k$
2.  $\forall q \in \mathcal{P}_{k-1} : p_k \perp q$  für  $k \geq 1$
3.  $p_k(x) = x^k + r$  mit  $\deg(r) \leq k - 1$  "Normierung"

Diese Polynome lassen sich rekursiv berechnen durch

$$\begin{aligned} p_0(x) &:= 1 \\ p_1(x) &:= x \\ p_{k+1}(x) &:= (x - \beta_{k+1})p_k(x) - \gamma_{k+1}^2 p_{k-1}(x), \quad \text{für } k \geq 2 \end{aligned}$$

Wobei  $\beta$  und  $\gamma$  definiert sind durch:

$$\begin{aligned} \beta_{k+1} &:= \frac{\langle xp_k, p_k \rangle}{\langle p_k, p_k \rangle} \\ \gamma_{k+1}^2 &:= \frac{\langle p_k, p_k \rangle}{\langle p_{k-1}, p_{k-1} \rangle} \end{aligned}$$

*Beweis.* (vgl. Gram-Schmidt Orthogonalisierung LinA)

Sei  $p_0, \dots, p_k$  bereits bekannt. Zur Konstruktion von  $p_{k+1}$  setzen wir

$$p_{k+1}(x) = xp_k(x) + \sum_{j=0}^k \alpha_j p_j(x)$$

(damit ist 3. erfüllt)

Zur Bestimmung der  $\alpha_j$ :

$$\begin{aligned} 1. \quad 0 &= \langle p_{k+1}, p_k \rangle = \langle xp_k, p_k \rangle + \alpha_k \langle p_k, p_k \rangle + \sum_{j=0}^{k-1} \alpha_j \underbrace{\langle p_j, p_k \rangle}_{=0} \\ &\Rightarrow \alpha_k = -\frac{\langle xp_k, p_k \rangle}{\langle p_k, p_k \rangle} =: -\beta_{k+1} \end{aligned}$$

2.

$$\begin{aligned} 0 &= \langle p_{k+1}, p_{k-1} \rangle = \langle xp_k, p_{k-1} \rangle + 0 + \alpha_{k-1} \langle p_{k-1}, p_{k-1} \rangle + 0 \\ &= \langle p_k, xp_{k-1} \rangle + \alpha_{k-1} \langle p_{k-1}, p_{k-1} \rangle \end{aligned}$$

Aufgrund von 3.  $\Rightarrow$

$$xp_{k-1} = p_k + r$$

mit  $\deg(r) \leq k - 1$

$$\begin{aligned} &\Rightarrow \langle p_k, xp_{k-1} \rangle = \langle p_k, p_k \rangle + \underbrace{\langle p_k, r \rangle}_{=0} \\ &\Rightarrow \alpha_{k-1} = -\frac{\langle p_k, p_k \rangle}{\langle p_{k-1}, p_{k-1} \rangle} =: -\gamma_{k+1}^2 \end{aligned}$$

3. Für  $j \leq k - 2$ :

$$\begin{aligned} 0 &= \langle p_{k+1}, p_j \rangle = \langle xp_k, p_j \rangle + \alpha_j \langle p_j, p_j \rangle \\ &= \underbrace{\langle p_k, xp_j \rangle}_{=0} + \alpha_j \underbrace{\langle p_j, p_j \rangle}_{\neq 0} \end{aligned}$$

$\langle p_k, xp_j \rangle = 0$  gilt, da  $\deg(xp_j) \leq k + 1$

Insgesamt haben wir

$$p_{k+1}(x) = xp_k(x) - \beta_{k+1}p_k(x) - \gamma_{k+1}^2 p_{k-1}(x)$$

□

### Bemerkung

Für eine QF maximaler Ordnung müssen nach Satz (4.1) die Knoten  $c_i$ ,  $i = 1, \dots, s$  so gewählt werden, dass

$$M(t) = \prod_{i=1}^s (t - c_i)$$

das Orthogonalpolynom vom Grad  $s$  bezüglich des Skalarprodukts mit  $\omega(x) \equiv 1$  auf  $[0, 1]$  ist.

Frage: Sind die Wurzeln (Nullstellen) der Orthogonalpolynome aus (5.3) reell? (Spoiler: Ja)

### Satz 5.4

Sei  $p_k$  das Orthogonalpolynom wie in (5.3) definiert (bzgl.  $\langle f, g \rangle = \int_a^b f(x)g(x)\omega(x)dx$ ). Alle Wurzeln von  $p_k$  sind einfach und liegen im offenen Intervall  $(a, b)$ .

*Beweis.* Seie  $x_1, \dots, x_r$  jene Wurzeln in  $p_k$ , die reell sind, in  $(a, b)$  liegen und bei denen  $p_k$  das Vorzeichen wechselt (Wurzeln mit ungerader Vielfachheit). Klar ist:  $r \leq k$ .

Sei

$$g(x) = \prod_{j=1}^r (x - x_j)$$

Dann ist

$$\langle p_k, g \rangle = \int_a^b \underbrace{p_k(x) g(x)}_{\text{Wechselt das Vorzeichen in (a,b) nicht}} \omega(x) dx \neq 0$$

Andererseits ist  $p_k$  orthogonal zu allen Polynomen vom Grad  $\leq k - 1$

$$\Rightarrow r = \deg(g) \geq k$$

$$\Rightarrow r = k$$

□

### Beispiel 5.5 (Orthogonale Polynome)

Bezeichnung	$(a, b)$	$\omega(x)$	Name
-------------	----------	-------------	------

$P_k$	$(-1, 1)$	1	Legendrepolynome
$T_k$	$(-1, 1)$	$(1 - x^2)^{-1/2}$	Tschebyscheff-Polynome
$P_k^{(\alpha, \beta)}$	$(-1, 1)$	$(1 - x)^\alpha(1 - x)^\beta$	Jacobi-Polynome $\alpha, \beta > -1$
$L_k^{(\alpha)}$	$(0, \infty)$	$x^\alpha e^{-x}$	Laguerre-Polynome
$M_k$	$(-\infty, \infty)$	$e^{-x^2}$	Harmitepolynome

Bemerkung: Teilweise sind andere Normierungen üblich  $P_k(1) = 1$ ,  $T_k(x) = \frac{2^{k-1}x^k + \dots, \dots}{2^{k-1}x^k + \dots, \dots}$

## 6 Ein adaptives Programm

Gegeben sei eine QF mit  $(b_i, c_i)_{i=1}^s$  mit Ordnung  $p = 2s$  (die höchste Ordnung, die es gibt) z.B.  $s = 15$

Ziel: Ein Computerprogramm adagaussqf(f, a, b, Tol), welches für eine Funktion  $f$  auf dem Intervall  $[a, b]$  eine Approximation an  $\int_a^b f(x)dx$  berechnet, sodass der Fehler  $\leq \text{Tol}$  ist (für viele Funktionen).

Konstruiere eine Zerlegung  $\Delta = \{a = x_0 < \dots < x_n = b\}$  des Intervalls, sodass für die Approximation

$$I_\Delta := \sum_{j=0}^{n-1} h_{j+1} \sum_{i=1}^s b_i f(x_j + c_i h_{j+1})$$

gilt

$$\left| I_\Delta - \int_a^b f(x)dx \right| \leq \text{Tol} \int_a^b |f(x)|dx$$

Schwierigkeiten:

- a) Schätzung des Fehlers
- b) Wahl der Zerlegung des Intervalls

### 6.1 (Zerlegung des Intervalls)

Für ein Teilintervall  $[x_j, x_{j+1}]$  von  $[a, b]$  lassen sich

$$res[x_j, x_{j+1}] := h_{j+1} \sum_{i=1}^s b_i f(x_j + c_i h_{j+1})$$

und

$$resabs[x_j, x_{j+1}] := h_{j+1} \sum_{i=1}^s |b_i f(x_j + c_i h_{j+1})|$$

berechnen.

Angenommen wir können eine Schätzung des Fehlers  $err[x, x_{j+1}]$  berechnen mit

$$err[x, x_{j+1}] \approx res[x, x_{j+1}] - \int_{x_j}^{x_{j+1}} f(x) dx,$$

dann bietet sich folgendes Verfahren zur Konstruktion einer Zerlegung an:

1. Berechne  $res[a, b]$ ,  $resabs[a, b]$  und  $err[a, b]$ .

Falls

$$|err[a, b]| \leq Tol \cdot resabs[a, b]$$

Gebe  $res[a, b]$  zurück.

Ansonsten:

2. Zerlege  $[a, b]$  in

$$I_0 = \left[ a, \frac{b-a}{2} \right]$$

und

$$I_1 = \left[ \frac{b-a}{2}, b \right]$$

und berechne

$res I_0$ ,  $resabs I_0$ ,  $err I_0$  und

$res I_1$ ,  $resabs I_1$ ,  $err I_1$

$n = 2$ .

3. Falls

$$\sum_{j=0}^{n-1} |err I_j| \leq Tol \sum_{j=0}^{n-1} resabs I_j$$

Gebe

$$\sum_{j=0}^{n-1} res I_j$$

zurück. Ansonsten:

Unterteile das Intervall  $I_k = [a_k, b_k]$ , in dem der Fehler maximal ist in zwei Teilintervalle

$$I_l = \left[ a_k, \frac{b_k - a_k}{2} \right]$$

und

$$I_m = \left[ \frac{b_k - a_k}{2}, b_k \right]$$

und berechne:

$res I_l, resabs I_l, err I_l$  und

$res I_m, resabs I_m, err I_m$

$n = n + 1$

Gehe zu 3)

## 6.2 (Schätzung des Fehlers)

Ziel: Berechne Approximation an

$$\int_{x_j}^{x_{j+1}} f(x) dx - h_{j+1} \sum_{i=1}^s b_i f(x_j + h_{j+1} c_i)$$

ohne zusätzliche Funktionsauswertungen.

Idee: Konstruiere eingebettete QF, d.h. QF zu den selben Knoten  $c_i$  mit Gewichten  $\hat{b}_i$  und Ordnung  $\hat{p} < p$ .

Bemerkung: Falls  $p = 2s$  ist, so gilt  $\hat{p} \leq s - 1$  (wäre  $\hat{p} \geq s$ , so wäre nach (2.8)  $\hat{b}_i = b_i$ ).

Eine Approximation des Fehlers für die eingebettete QF ist durch

$$\begin{aligned}\text{diff}[x_j, x_{j+1}] &= h_{j+1} \sum_{i=1}^s b_i f(x_j + c_i h_{j+1}) - h_{j+1} \sum_{i=1}^s \hat{b}_i f(x_j + c_i h_{j+1}) \\ &= h_{j+1} \sum_{i=1}^s (b_i - \hat{b}_i) f(x_j + c_i h_{j+1})\end{aligned}$$

gegeben. Es gilt

$$\begin{aligned}\text{diff}[x_j, x_{j+1}] &= h_{j+1} \sum_{i=1}^s b_i f(x_j + c_i h_{j+1}) - \int_{x_j}^{x_{j+1}} f(x) dx \\ &\quad - \left( h_{j+1} \sum_{i=1}^s \hat{b}_i f(x_j + c_i h_{j+1}) - \int_{x_j}^{x_{j+1}} f(x) dx \right) \\ &= \text{Fehler der QF } (b_i, c_i)_{i=1}^s - \text{Fehler der QF } (\hat{b}_i, c_i)_{i=1}^s \\ &= C_1 h_{j+1}^{p+1} + C_2 h_{j+1}^{\hat{p}+1}\end{aligned}$$

Falls  $h_{j+1}$  klein ist, ist  $C_1 h_{j+1}^{p+1} \ll C_2 h_{j+1}^{\hat{p}+1}$ .

Drei Möglichkeiten den Fehler zu schätzen:

- I)  $\text{err}[x_j, x_{j+1}] \approx \text{diff}[x_j, x_{j+1}]$ . Sehr pessimistisch
- II)  $\text{err}[x_j, x_{j+1}] \approx (\text{diff}[x_j, x_{j+1}])^2$ , falls  $p = 2s$  und  $\hat{p} = s - 1$ . Wenig verlässlich
- III) Verwende dritte eingebettete QF

$(\hat{b}_i, c_i)$  der Ordnung 6

zu  $(b_i, c_i)$  der Ordnung  $30 = 2s$ ,  $s = 15$

und  $(\hat{b}_i, c_i)$  der Ordnung 14

$$\hat{\text{diff}} = h_{j+1} \sum_{i=1}^s (b_i - \hat{b}_i) f(x_j + c_i h_{j+1}) \approx C_3 h^7$$

$$\begin{aligned}\text{err}[x_j, x_{j+1}] &= \text{diff}[x_j, x_{j+1}] \left( \frac{\text{diff}}{\hat{\text{diff}}} \right)^2 \\ &= C_2 \frac{C_2^2}{C_3^2} h_{j+1}^{15} \left( \frac{h_{j+1}^{15}}{h_{j+1}^7} \right)^2 = C h_{j+1}^{31}\end{aligned}$$

## 7 Gauß- und Lobatto Quadraturformeln

Ziel: Konstruktion einer s-stufigen QF der Ordnung  $p = 2s$ .

Für  $M(t) = CP_s(2t - 1)$ , wobei  $P_s$  das Legendrepolynom vom Grad  $s$  ist (siehe (5.5)),  $C \in \mathbb{R}$ , erhalten wir mit (5.4) und (4.1):

### Satz 7.1

Für jedes  $s \in \mathbb{N}$  gibt es eine eindeutige QF der Ordnung  $p = 2s$ , die sogenannte Gauß-QF. Ihre Knoten sind die Wurzeln von  $t \mapsto P_s(2t - 1) : [0, 1] \rightarrow \mathbb{R}$ , ihre Gewichte sind durch (2.8) gegeben.

### Beispiel

$s = 1$  Mittelpunktsregel

$s = 2 \quad c_{1,2} = \frac{1}{2} \mp \frac{\sqrt{3}}{6}, b_1 = \frac{1}{2} = b_2$

$s = 3 \quad (4.3) \quad 2)$

### 7.2 (Bezeichnung der Knoten der Gauß-QF)

Details: Siehe Homepage (Übungsaufgabe).

Idee: Die Wurzeln der Polynome, die durch Rekursion (5.3) erzeugt werden, sind die Eigenwerte einer symmetrischen Tridiagonalmatrix (Matrix: Siehe Homepage).

Zusammengefasst:

**Satz:** Es seien  $P_0, \dots, P_n$  Polynome definiert wie in Satz (5.3).

$\lambda \in \mathbb{R}$  ist eine Nullstelle von  $P_n \Leftrightarrow \lambda$  ist ein Eigenwert der Tridiagonalmatrix  $T_n$ .  $\Phi_n(\lambda)$  ist dann der Eigenvektor zum Eigenwert  $\lambda$ .

$T_n$  und  $\Phi_n(\lambda)$  sind gegeben durch:

$$T_n = \begin{bmatrix} \beta_1 & 1 & & & \\ \gamma_2^2 & \beta_2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{n-1}^2 & \beta_{n-1} & 1 \\ & & & \gamma_n^2 & \beta_n \end{bmatrix}$$

$$\Phi_n(\lambda) = [P_0 \ \dots \ P_{n-1}]^T$$

In Numerik II lernen Sie Verfahren kennen, um die Eigenwerte zu berechnen.

### 7.3 (Lobatto Quadraturformeln)

Ein Vorteil der Simpsonquadraturformel war, dass  $c_1 = 0$  und  $c_s = 1$  gilt.

Damit muss man den Integranten in  $x_j$  nur einmal auswerten. Zur Konstruktion einer s-stufigen QF der Ordnung  $p = 2s - 2$  mit  $c_1 = 0$  und  $c_s = 1$  setzt man

$$M(t) = P_s(2t - 1) - P_{s-2}(2t - 1)$$

Da die Legendre-Polynome folgende Rekursion erfüllen

$$P_0(x) = 1 \quad P_1(x) = x$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

ist

$$P_s(1) = 1 \quad \text{und} \quad P_s(-1) = (-1)^s$$

und damit

$$M(0) = 0 = M(1)$$

Die restlichen Nullstellen (oder Wurzeln) von  $M(t)$  sind reell, einfach und liegen in  $(0,1)$ , wie man analog zu (5.4) zeigt.

Damit gilt:

**Satz** Für  $s \in \mathbb{N}$ ,  $s \geq 2$  gibt es eine eindeutige s-stufige QF der Ordnung  $2s - 2$  mit  $c_1 = 0$  und  $c_s = 1$

## II Interpolation und Approximation

**Problemstellung A** Zu gegebenen  $(x_0, y_0), \dots, (x_n, y_n)$  berechne Polynom  $p$  vom Grad  $\leq n$  mit

$$p(x_j) = y_j, \quad j = 0, \dots, n$$

**Problemstellung B**  $f : [a, b] \rightarrow \mathbb{R}$  gegeben. Finde einfach auszuwertende Funktion  $p : [a, b] \rightarrow \mathbb{R}$ , etwa ein Polynom, stückweises Polynom, rationale Funktion, sodass  $f - p$  klein ist.

- i)  $f(x) = p(x)$  für endlich viele vorgegebene Punkte  $x$
- ii)  $\int_a^b (f(x) - p(x))^2 dx$  soll minimal sein.
- iii)  $\max_{x \in [a, b]} |f(x) - p(x)|$  soll minimal sein.

## 8 Newtonsche Interpolationsformel

### Beispiel 8.1

n=1:

$(x_0, y_0), (x_1, y_1)$ ,  $p \in \mathcal{P}_1$  das beide Punkte verbindet.

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

n=2:

$(x_0, y_0), (x_1, y_1), (x_2, y_2)$

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

Bestimme  $a$  so, dass  $p(x_2) = y_2$

$$\begin{aligned} y_2 &\stackrel{!}{=} y_0 + \left( \frac{-x_1+x_1}{x_2-x_0} \right) \frac{y_1-y_0}{x_1-x_0} + a(x_2-x_0)(x_2-x_1) \\ a(x_2-x_0)(x_2-x_1) &= y_2 - y_0 - (x_2-x_1) \frac{y_1-y_0}{x_1-x_0} - y_1 + y_0 \\ \Rightarrow a &= \frac{1}{x_2-x_0} \left( \frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0} \right) \end{aligned}$$

### Definition 8.2 (dividierte Differenzen)

Für  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  mit paarweise verschiedenen Stützstellen  $x_j$  definieren wir

$$\begin{aligned} y[x_j] &:= y_j \quad (= \delta^0 y[x_j]) \\ \delta y[x_j, x_{j+1}] &:= \frac{y_{j+1} - y_j}{x_{j+1} - x_j} = \frac{\delta^0 y[x_{j+1}] - \delta^0 y[x_j]}{x_{j+1} - x_j} \\ \delta^2 y[x_j, x_{j+1}, x_{j+2}] &:= \frac{\delta y[x_{j+1}, x_{j+2}] - \delta y[x_j, x_{j+1}]}{x_{j+2} - x_j} \\ \delta^k y[x_j, x_{j+1}, \dots, x_{j+k}] &:= \frac{1}{x_{j+k} - x_j} (\delta^{k-1} y[x_{j+1}, \dots, x_{j+k}] - \delta^{k-1} y[x_j, \dots, x_{j+k-1}]) \end{aligned}$$

Schema:

$x_0$	$y_0$	$\delta^1 y[x_0, x_1]$		
$x_1$	$y_1$		$\delta^2 y[x_0, x_1, x_2]$	$\delta^3 y[x_0, x_1, x_2, x_3]$
$x_2$	$y_2$	$\delta^1 y[x_1, x_2]$	$\delta^2 y[x_1, x_2, x_3]$	
$x_3$	$y_3$	$\delta^1 y[x_2, x_3]$		

### Bemerkung 8.3

Falls die  $x_i$  äquidistant, dh.  $x_i = x_0 + ih$  so ist:

$$\begin{aligned}\delta y[x_i, x_{i+1}] &= \frac{y_{i+1} - y_i}{h} =: \frac{1}{h} \Delta y_i \\ \delta^2 y[x_i, x_{i+1}, x_{i+2}] &= \frac{\frac{1}{h} \Delta y_{i+1} - \frac{1}{h} \Delta y_i}{2h} = \frac{1}{2h^2} \Delta^2 y_i \\ \delta^k y[x_i, \dots, x_{i+k}] &= \frac{1}{k!h^k} \Delta^k y_i,\end{aligned}$$

wobei  $\Delta^k y_i := \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$ .

### Satz 8.4 (Newton'sche Interpolationsformel)

Zu paarweise verschiedenen reellen  $x_i$ ,  $i = 0, \dots, n$ , existiert ein eindeutiges Polynom  $p \in \mathcal{P}_n$  durch die Punkte  $(x_i, y_i)$ ,  $i = 0, \dots, n$  (d.h.  $p(x_i) = y_i$  für  $i = 1, \dots, n$ ). Es lässt sich berechnen durch:

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

*Beweis.* (Induktion)

**IA**  $n = 1$  (und  $n = 2$ ) vgl. Beispiel (1.1)

**IS**  $n - 1 \rightarrow n$

$$p_0(x) = y[x_0] + (x - x_0)\delta y[x_0, x_1] + \dots + (x - x_0)\dots(x - x_{n-2})\delta^{n-1} y[x_0, \dots, x_{n-1}]$$

ist das eindeutige interpolierende Polynom mit

$$\deg(p_0) \leq n - 1$$

zu  $(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ .

Für den Ansatz

$$p(x) = p_0(x) + a(x - x_0)(x - x_1)\dots(x - x_{n-1})$$

ergibt die Forderung  $p(x_n) = y_n$

$$a = \frac{y_n - p_0(x_n)}{(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-1})}$$

Da  $a$  eindeutig ist, ist  $p$  eindeutig.

Es bleibt zu zeigen:  $a = \delta^n y[x_0, \dots, x_n]$

Sei dazu ein Polynom  $p_1(x)$ , welches durch  $(x_1, y_1), \dots, (x_n, y_n)$  läuft, mit  $\deg(p_1) \leq n - 1$ . Nach Induktionsannahme gilt

$$\begin{aligned} p_1(x) &= y[x_1] + (x - x_1)\delta^1 y[x_1, x_2] + \dots + (x - x_1)\dots(x - x_{n-1})\delta^{n-1} y[x_1, \dots, x_n] \\ &= x^{n-1}\delta^{n-1} y[x_1, \dots, x_n] + r \end{aligned}$$

mit  $\deg(r) \leq n - 2$ .

Setze Polynom

$$p(x) := \frac{x_n - x}{x_n - x_0} p_0(x) + \frac{x - x_0}{x_n - x_0} p_1(x)$$

mit  $\deg(p) \leq n$  durch  $(x_0, y_0), \dots, (x_n, y_n)$ .

Das gilt, da:

$$p(x_0) = p_0(x_0) = y_0$$

$$p(x_n) = p_1(x_n) = y_n$$

Für  $i = 1, \dots, n - 1$ :

$$p(x_i) = \frac{x_n - x_i}{x_n - x_0} \underbrace{p_0(x_i)}_{y_i} + \frac{x_i - x_0}{x_n - x_0} \underbrace{p_1(x_i)}_{y_i} = y_i$$

Andererseits:

$$p(x) = ax^n + r \quad \text{mit } \deg(r) \leq n - 1$$

Koeffizientenvergleich:

$$\begin{aligned} a &= -\frac{1}{x_n - x_0} \delta^{n-1} y[x_0, \dots, x_{n-1}] + \frac{1}{x_n - x_0} \delta^{n-1} y[x_1, \dots, x_n] \\ &= \delta^n y[x_0, \dots, x_n] \end{aligned}$$

□

## 8.5 (Hornerschema)

Zur Auswertung des Interpolationspolynom  $p$  an der Stelle  $x$  verwendet man

$$p(x) = y[x_0] + (x - x_0) (\delta y[x_0, x_1] + (x - x_1) (\delta^2 y[x_0, x_1, x_2] + (x - x_2) (\dots (\delta^n y[x_0, \dots, x_n]))))$$

Algorithmus:

```

 $s = \delta^n y[x_0, \dots, x_n]$ 
for  $k = n - 1, \dots, 0$  do
     $s = \delta^k y[x_0, \dots, x_k] + (x - x_k)s$ 
end for

```

### Beispiel 8.6

$i$	$x_i$	$y_i$	$\delta^1 y[x_0, x_1]$	$\delta^2 y[x_0, x_1, x_2]$	$\delta^3 y[x_0, \dots, x_3]$	$\delta^4 y[x_0, \dots, x_4]$
0	-1	0	$\frac{1-0}{0-(-1)} = 1$			
1	0	1		$\frac{0-1}{2-(-1)} = -\frac{1}{3}$	$\frac{\frac{2}{3}-(-\frac{1}{3})}{3-(-1)} = \frac{1}{4}$	
2	2	1		$\frac{2-0}{3-0} = \frac{2}{3}$		$\frac{-\frac{2}{5}-\frac{1}{4}}{5-(-1)} = -\frac{13}{120}$
3	3	3	$\frac{3-1}{3-2} = 2$	$\frac{-2-2}{5-2} = -\frac{4}{3}$	$\frac{-\frac{4}{3}-\frac{2}{3}}{5-0} = -\frac{2}{5}$	
4	5	-1	$\frac{-1-3}{5-3} = -2$			

Das Interpolationspolynom ist also

$$p(x) = 0 + (x+1)*1 - \frac{1}{3}(x+1)(x) + \frac{1}{4}(x+1)x(x-2) + (x+1)x(x-2)(x-3) \left( -\frac{13}{120} \right)$$

bzw. nach Hornerschema

$$p(x) = 0 + (x+1) \left( 1 + x \left( -\frac{1}{3} + (x-2) \left( \frac{1}{4} + (x-3) \left( -\frac{13}{120} \right) \right) \right) \right)$$

Werte  $p(x)$  an der Stelle 1 aus:

$$\begin{aligned} -\frac{13}{120} * (-2) &= \frac{26}{120} \\ \left(\frac{1}{4} + \frac{26}{120}\right)(-1) &= -\frac{56}{120} = -\frac{7}{15} \\ \left(-\frac{7}{15} - \frac{1}{3}\right)1 &= -\frac{12}{15} = -\frac{4}{5} \\ \left(-\frac{4}{5} + 1\right)2 &= \frac{2}{5} = p(1) \end{aligned}$$

## 9 Fehler bei der Polynominterpolation

Problem:  $f: [a, b] \rightarrow \mathbb{R}$  werde interpoliert in Stützstellen  $x_0, \dots, x_n \in [a, b]$  durch  $p \in \mathcal{P}_n$  mit  $p(x_i) = f(x_i)$  für  $i = 0, \dots, n$ .

Wie groß ist der Fehler  $f(x) - p(x)$ ?

### Satz 9.1

Sei  $f: [a, b] \rightarrow \mathbb{R}$   $(n+1)$ -mal stetig differenzierbar,  $p \in \mathcal{P}_n$  mit  $p(x_i) = f(x_i)$  ( $i = 0, \dots, n$ ) das Interpolationspolynom zu paarweise verschiedenen Stützstellen  $x_i \in [a, b]$  ( $i = 0, \dots, n$ ). Dann gilt:

$$\forall x \in [a, b] \exists \xi = \xi(x) \in (a, b): f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

*Beweis.* Siehe (9.4) □

**Beispiel 9.2** (Berechnung von Logarithmentafeln: Briggs, 17. Jhd)

$$f(x) = \log_{10}(x), \quad x \in [55, 58]$$

Wähle Stützstellen:

$$x_0 = 55, \quad x_1 = 56, \quad x_2 = 57, \quad x_3 = 58$$

Es seien

$$\log_{10}(55), \log_{10}(56), \log_{10}(57) \text{ und } \log_{10}(58)$$

bereits bekannt. Berechne eine Näherung von  $f$  bei  $\log_{10}(56.5)$

→ Interpolationspolynom  $p$ :

$$\log_{10}(56.5) = 1.752048448$$

$$p(56.5) = 1.75204845$$

$$f'(x) = \frac{1}{\ln(10)x}$$

$$f''(x) = -\frac{1}{\ln(10)x^2}$$

$$f^{(3)}(x) = \frac{2}{\ln(10)x^3}$$

$$f^{(4)}(x) = -\frac{6}{\ln(10)x^4}$$

Für  $x \in [55, 58]$ :

$$|f^{(4)}(x)| \leq \frac{6}{55^4 \ln(10)} \Rightarrow$$

$$\begin{aligned} |\log_{10}(56.5) - p(56.5)| &\leq 1.5 * 0.5 * 0.5 * 1.5 * \frac{6}{55^4 \ln(10) \frac{1}{4!}} \\ &\approx 6.7 * 10^{-9} \end{aligned}$$

Für den Beweis von (9.1) wird folgendes Lemma benötigt:

### Lemma 9.3

Sei  $f \in C^n([a, b])$  und sei für paarweise verschiedene  $x_i \in [a, b]$  ( $i = 0, \dots, n$ )  $y_i := f(x_i)$ . Dann existiert  $\xi \in (\min_i(x_i), \max_i(x_i))$ , sodass

$$\delta^n y[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!} \quad (x_0 < x_1 < \dots < x_n)$$

*Beweis.* Sei  $p$  ein Interpolationspolynom zu  $(x_i, y_i)_{i=0}^n$ . Setzt man  $d := p - f$ , so gilt  $d(x_i) = 0$  für  $i = 0, \dots, n$ .

n-maliges anwenden des Mittelwertsatzes liefert paarweise verschiedene  $\xi_i$ , ( $i = 0, \dots, n - 1$ ) mit  $d'(\xi_i) = 0$  für  $\xi_i \in (\min_j(x_j), \max_j(x_j))$ .

Dasselbe Argument angewandt auf  $d'$  liefert  $\eta_0, \dots, \eta_{n-2}$  mit  $d''(\eta_i) = 0$  für  $i = 0, \dots, n - 2$ .

Wiederhole dies bis:

Es existiert  $\rho_0$  mit  $d^{(n)}(\rho_0) = 0$

$\Rightarrow f^{(n)}(\rho_0) = p^{(n)}(\rho_0) = n! \delta^n y[x_0, \dots, x_n]$ ,

da  $\delta^n y[x_0, \dots, x_n]$  der Koeffizient von  $x^n$  in  $p$  ist.  $\square$

### Bemerkung

Für  $n = 1$  ist Lemma (9.3) der Mittelwertsatz (oder Satz von Rolle) aus Ana I:

$$\exists \xi : \frac{f(x_1) - f(x_2)}{x_1 - x_2} = f'(\xi)$$

#### 9.4 (Beweis von (9.1))

Sei  $\bar{x} \in [a, b]$  beliebig.

1. **Fall**  $\bar{x} = x_i$  für ein  $i \in \{0, \dots, n\}$ , so ist wegen  $p(x_i) - f(x_i) = 0$  nichts zu zeigen.
2. **Fall**  $\bar{x} \neq x_i$  für alle  $i \in \{0, \dots, n\}$ . Sei  $\bar{p}$  das Interpolationspolynom mit  $\deg(\bar{p}) \leq n + 1$  zu  $(x_i, f(x_i))_{i=0}^n$  und  $(\bar{x}, f(\bar{x}))$ . Die Newton'sche Interpolationsformel liefert dann

$$\begin{aligned} \bar{p}(x) &= p(x) + \prod_{i=0}^n (x - x_i) \delta^{n+1} y[x_0, \dots, x_n, \bar{x}] \\ &\stackrel{(9.3)}{=} p(x) + \prod_{i=0}^n (x - x_i) \frac{f^{(n+1)}(\xi)}{(n+1)!} \end{aligned}$$

Für  $x = \bar{x}$  gilt  $\bar{p}(\bar{x}) = f(\bar{x})$ . Damit ist Satz (9.1) für  $x \in [a, b]$  gezeigt.  $\square$

Fragen:

- Für welche Wahl der Stützstellen  $x_i$  ( $i = 0, \dots, n$ ,  $n$  fest) ist

$$\max_{x \in [a, b]} \left| \prod_{i=0}^n (x - x_i) \right|$$

minimal? (Siehe Abschnitt 10)

- Wie wirken sich Fehler in den Funktionsauswertungen (etwa Messfehler oder Rechenfehler) auf das Interpolationspolynom aus?

### Satz 9.5 (Lagrange Interpolationsformel)

Das Interpolationspolynom  $p$  zu  $(x_i, y_i)_{i=0}^n$  ist gegeben durch

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

mit

$$l_i(x) = \frac{\prod_{j=0, j \neq i}^n (x - x_j)}{\prod_{j=0, j \neq i}^n (x_i - x_j)}$$

*Beweis.*  $\deg(l_i) = n$ ,  $l_i(x_j) = \begin{cases} 1 & \text{für } j = i \\ 0 & \text{sonst} \end{cases}$   
 $\Rightarrow p(x_j) = \sum_{i=0}^n y_i l_i(x_j) = y_j$

□

### Bemerkung

Lagranges und Newtons Interpolationsformeln liefern beide das gleiche Polynom nur in unterschiedlichen Darstellungen.

### Definition 9.6

$$\Lambda_n := \max_{x \in [a,b]} \sum_{i=0}^n |l_i(x)|$$

heißt die **Lebesgue Konstante** zu den Stützstellen  $x_i$ ,  $i = 0, \dots, n$  auf dem Intervall  $[a, b]$ .

Damit gilt:

### Satz 9.7

Sei  $p$  das Interpolationspolynom (vom Grad  $\leq n$ ) zu  $(x_i, y_i)_{i=0}^n$  und  $\tilde{p}$  das Interpolationspolynom zu  $(x_i, \tilde{y}_i)_{i=0}^n$ , so gilt:

$$\max_{x \in [a,b]} |p(x) - \tilde{p}(x)| \leq \Lambda_n \max_{i=0, \dots, n} |y_i - \tilde{y}_i|$$

*Beweis.* klar

□

### Beispiel 9.8

- Für äquidistante Stützstellen  $x_i = a + i \frac{b-a}{n}$  ( $i = 0, \dots, n$ ) ist

$$\Lambda_{10} \approx 40$$

$$\Lambda_{20} \approx 3 * 10^4$$

$$\Lambda_{40} \approx 10^{10}$$

$$\Lambda_n \approx \frac{2^n}{\ln(n) * e * n} \quad \text{für } n \rightarrow \infty$$

$\Rightarrow$  Vorsicht bei Polynominterpolation mit vielen äquidistanten Stützstellen!  
In §10 werden wir Stützstellen kennenlernen mit  $\Lambda_n \leq 4$  für  $n \leq 100$ .

### Satz 9.9

Sei  $f: [a, b] \rightarrow \mathbb{R}$  stetig,  $p$  Interpolationspolynom zu  $f$  in den Stützstellen  $x_0, \dots, x_n \in [a, b]$ . So gilt:

$$\forall q \in \mathcal{P}_{n+1}: \max_{x \in [a, b]} |f(x) - p(x)| \leq (1 + \Lambda_n) \max_{x \in [a, b]} |q(x) - f(x)|.$$

Hierbei ist  $\Lambda_n$  die Lebesgue-Konstante zu  $(x_i)_{i=0}^n$  auf  $[a, b]$ .

*Beweis.* Sei  $q \in \mathcal{P}$ .

$$f - p = (f - q) + (q - p)$$

$q$  ist das Interpolationspolynom zu sich selbst in den  $x_0, \dots, x_n$ . Nach (9.7) gilt für  $y_i = f(x_i)$   $\tilde{y}_i = q(x_i)$ .

$$\begin{aligned} \max_{x \in [a, b]} |p(x) - q(x)| &\leq \Lambda_n \max_{i=0, \dots, n} |f(x_i) - q(x_i)| \\ &\leq \Lambda_n \max_{x \in [a, b]} |f(x) - q(x)| \\ \Rightarrow \max_{x \in [a, b]} |f(x) - p(x)| &\leq \max_{x \in [a, b]} |f(x) - q(x)| + \max_{x \in [a, b]} |p(x) - q(x)| \\ &\leq (1 + \Lambda_n) \max_{x \in [a, b]} |q(x) - f(x)| \end{aligned}$$

□

## 10 Tschebyscheff-Interpolation

Ziel: Interpoliere  $f: [a, b] \rightarrow \mathbb{R}$  in "guten" Stützstellen.

Ohne Einschränkungen sei  $[a, b] = [-1, 1]$

### Definition 10.1

$T_n(x) = \cos(n * \arccos(x))$  für  $x \in [-1, 1]$  heißt n-tes Tschebyscheff-Polynom.

### Lemma 10.2

$T_n(x)$  ist für  $x \in [-1, 1]$  ein Polynom mit folgenden Eigenschaften:

- i)  $T_0(x) = 1, T_1(x) = x$
- ii)  $T_n(x) = 2^{n-1}x^n + r(x)$  mit  $r_n \in \mathcal{P}_{n-1}$
- iii)  $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$
- iv)  $\forall x \in [-1, 1]: |T_n(x)| \leq 1$

$$v) \quad T_n(\cos(\frac{k\pi}{n})) = (-1)^k, \quad k = 0, \dots, n$$

$$vi) \quad T_n(\cos(\frac{(2k+1)\pi}{2n})) = 0, \quad k = 0, \dots, n-1$$

*Beweis.*

i) klar, da  $T_0(x) = \cos(0) = 1$ ,  $T_1(x) = \cos(\arccos(x)) = x$ ,  $x \in [a, b]$

$$\begin{aligned} iii) \quad & \cos((n+1)\phi) + \cos((n-1)\phi) \\ &= \cos(n\phi)\cos(\phi) - \sin(n\phi)\sin(\phi) + \cos(n\phi)\cos(-\phi) - \sin(n\phi)\sin(-\phi) \\ &= 2\cos(n\phi)\cos(\phi) \end{aligned}$$

ii) folgt aus i) und iii)

iv) klar, da  $\cos: \mathbb{R} \rightarrow [-1, 1]$

$$v) + vi) \quad \text{ebenfalls klar, da } T_n(\cos(\frac{k\pi}{n})) = \cos(n\frac{k\pi}{n}) = \cos(k\pi) = (-1)^k$$

analog:  $T_n(\cos(\frac{(2k+1)\pi}{2n})) = \cos(n\frac{(2k+1)\pi}{2n}) = 0$

□

### Beispiel 10.3

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x$$

### Lemma 10.4

Sei  $q \in \mathcal{P}_n$ ,  $q(x) = 2^{n-1}x^n + r(x)$  mit  $r(x) \in \mathcal{P}_{n-1}$ ,  $q \neq T_n$ . Dann gilt

$$\max_{x \in [-1, 1]} |q(x)| > \max_{x \in [-1, 1]} |T_n(x)| \quad (= 1)$$

*Beweis.* Annahme:  $\forall x \in [-1, 1]: |q(x)| \leq 1$

$$T_n(1) = 1$$

$$T_n(\cos(\frac{\pi}{n})) = -1$$

Nach dem Zwischenwertsatz hat  $q - T_n$  eine Nullstelle im Intervall  $[\cos(\frac{\pi}{n}), 1]$ . Falls ein "Randpunkt"  $x$  eine Nullstelle ist, so handelt es sich um eine doppelte Nullstelle, da  $q'(x) = 0 = T'_n(x)$ . Ebenso existiert in  $[\cos(\frac{2\pi}{n}), \cos(\frac{\pi}{n})]$  und allgemein in  $[\cos(\frac{(k+1)\pi}{n}), \cos(\frac{k\pi}{n})]$  für  $k = 0, \dots, n-1$ .

Nullstelle  $\Rightarrow q - T_n$  hat  $n$  Nullstellen.

Andererseits ist  $q - T_n \in \mathcal{P}_{n-1} \Rightarrow q - T_n \equiv 0 \Rightarrow q = T_n \not\subseteq$

□

### Satz 10.5

Unter allen Unterteilungen  $\{x_0, \dots, x_n\}$  von  $[-1, 1]$  wird

$$\max_{x \in [-1, 1]} |(x - x_0)(x - x_1) \dots (x - x_n)|$$

minimal für  $x_k = \cos\left(\frac{2k+1}{n+1}\frac{\pi}{2}\right)$ ,  $k = 0, \dots, n$  (d.h.  $x_k$  sind die Wurzeln von  $T_{n+1}$ )

*Beweis.* Nach Lemma (10.4) wird  $\max_{x \in [-1, 1]} |(x - x_0)(x - x_1) \dots (x - x_n)|$  minimal gdw.  $(x - x_0) \dots (x - x_n) = 2^{-n}T_{n+1}(x)$ , d.h. falls  $x_k$  Wurzeln von  $T_{n+1}$  sind.  $\square$

### Satz 10.6

Die Lebesguekonstanten  $\Lambda_n$  zu den Tschebyscheffknoten (Wurzeln von  $T_{n+1}$ ) erfüllen

$$\Lambda_n \leq 3 \text{ für } n \leq 20$$

$$\Lambda_n \leq 4 \text{ für } n \leq 100$$

$$\Lambda_n \approx \frac{2}{\pi} \log(n) \text{ für } n \rightarrow \infty$$

*Beweis.* ohne Beweis.  $\square$

Nach Satz (9.9) liefert die Interpolation in den Wurzeln der Tschebyscheffpolynome eine fast optimale Polynominterpolation an  $f$ .

Dazu kommen Eigenschaften, die die Berechnung eines Interpolationspolynoms in den Tschebyscheffknoten (Wurzeln der Tschebyscheffpolynome) vereinfachen.

### Lemma 10.7

Die Tschebyscheffpolynome sind orthogonal, bzgl. des Skalarprodukts

$$\langle f, g \rangle := \int_{-1}^1 f(x)g(x) \frac{1}{\sqrt{1-x^2}} dx$$

*Beweis.* Übungsaufgabe  $\square$

### Lemma 10.8

Die Tschebyscheffpolynome  $T_k$ ,  $k = 0, \dots, n$  sind orthogonal bzgl. des Skalarprodukts (auf  $\mathcal{P}_n$ )

$$(f, g) := \sum_{l=0}^n f(x_l)g(x_l), \quad \text{wobei } x_0, \dots, x_n \text{ Wurzeln von } T_{n+1}(x)$$

*Beweis.*

$$\begin{aligned} T_k(x_l) &= \cos \left( k * \arccos \left( \cos \left( \frac{2l-1}{n+1} \frac{\pi}{2} \right) \right) \right) \\ &= \cos \left( k \frac{2l+1}{n+1} \frac{\pi}{2} \right) \\ &= \cos \left( k \left( l + \frac{1}{2} \right) h \right) \end{aligned}$$

für  $h = \frac{\pi}{n+1}$

Damit ist

$$(T_k, T_j) = \sum_{l=0}^n \cos \left( k \left( l + \frac{1}{2} \right) h \right) * \cos \left( j \left( l + \frac{1}{2} \right) h \right),$$

$$\text{da } \cos(x)\cos(y) = \frac{1}{2}(\cos(x+y) + \cos(x-y))$$

$$= \frac{1}{2} \sum_{l=0}^n \cos \left( (k+j) \left( l + \frac{1}{2} \right) h \right) * \cos \left( (k-j) \left( l + \frac{1}{2} \right) h \right)$$

Es gilt:  $\cos(x) = \operatorname{Re}(e^{ix})$

$$\begin{aligned} &= \frac{1}{2} \operatorname{Re} \left( \sum_{l=0}^n e^{i(k+j)(l+\frac{1}{2})h} + e^{i(k-j)(l+\frac{1}{2})h} \right) \\ &= \frac{1}{2} \operatorname{Re} \left( \sum_{l=0}^n \left( e^{i(k+j)lh} e^{i(k+j)\frac{h}{2}} + e^{i(k-j)lh} e^{i(k-j)\frac{h}{2}} \right) \right) \\ &= \frac{1}{2} \operatorname{Re} \left( e^{i(k+j)\frac{h}{2}} \frac{e^{i(k+j)h(n+1)} - 1}{e^{i(k+j)h} - 1} + e^{i(k-j)\frac{h}{2}} \frac{e^{i(k-j)h(n+1)} - 1}{e^{i(k-j)h} - 1} \right), \quad \text{für } k \neq j \end{aligned}$$

Es gilt  $k(n+1) = \pi$

$$\stackrel{\text{Behauptung}}{=} \begin{cases} 0 & k \neq j \\ \frac{1}{2}(n+1) & k = j \neq 0 \\ (n+1) & k = j = 0 \end{cases}$$

**Fall 1:**  $k = j = 0 \Rightarrow \frac{1}{2} \sum_{l=0}^n (1+1) = (n+1)$

**Fall 2:**  $k = j \neq 0 \Rightarrow \frac{1}{2}\operatorname{Re} \left( (n+1) + e^{ijh} \underbrace{\frac{e^{i2j} \overbrace{(n+1)h}^{\stackrel{=\pi}{=}} - 1}{e^{i2jh} - 1}}_{=0} \right) = \frac{1}{2}(n+1)$

**Fall 3:**  $k \neq j$ :

**Fall 1:**  $k+j$  ist gerade  $\Rightarrow k-j$  ist gerade  
 $\Rightarrow \frac{1}{2}\operatorname{Re}(0+0) = 0$

**Fall 2:**  $k+j$  ist ungerade  $\Rightarrow k-j$  ist ungerade

$$\begin{aligned} & \Rightarrow \frac{1}{2}\operatorname{Re} \left( e^{i(k+j)\frac{h}{2}} \frac{-2}{e^{i(k+j)h} - 1} + e^{i(k-j)\frac{h}{2}} \frac{-2}{e^{i(k-j)h} - 1} \right) \\ & = \frac{1}{2}\operatorname{Re} \left( \underbrace{\frac{-2}{e^{i(k+j)\frac{h}{2}} + e^{-i(k+j)\frac{h}{2}}}_{\text{rein imaginär}} + \underbrace{\frac{-2}{e^{i(k-j)\frac{h}{2}} - e^{-i(k-j)\frac{h}{2}}}_{\text{rein imaginär}}} \right) \\ & = 0 \end{aligned}$$

□

### Bemerkung

$(\cdot, \cdot)$  ist ein Skalarprodukt auf  $\mathcal{P}_n$ , da

- i) bilinear
  - ii) symmetrisch
  - iii) positiv definit
- $$(f, f) = \sum_{l=0}^n f(x_l)^2 \geq 0$$
- $$(f, f) = 0 \stackrel{!}{\Rightarrow} f \equiv 0$$
- $$\sum_{l=0}^n f(x_l)^2 = 0 \Rightarrow \forall l : f(x_l) = 0 \Leftrightarrow f \equiv 0, \text{ da } \deg(f) \leq n$$

### Satz 10.9

Sei  $p$  das Interpolationspolynom zur Funktion  $f$  in den Tschebyscheffknoten  $x_0, \dots, x_n$  (Wurzeln von  $T_{n+1}$ ), so gilt:

$$p(x) = \frac{1}{2}c_0 + \sum_{j=1}^n c_j T_j(x),$$

wobei

$$c_k = \frac{2}{n+1} \sum_{l=0}^n f(x_l) \cos \left( k \frac{2l+1}{n+1} \frac{\pi}{2} \right), \quad \text{für } k = 0, \dots, n$$

*Beweis.* Betrachte  $(p, T_k)$

$$\begin{aligned} (p, T_k) &= \frac{1}{2}(T_0, T_k) + \sum_{l=1}^n c_l(T_l, T_k) \\ &= \begin{cases} c_k(T_k, T_k) & \text{für } k \neq 0 \\ \frac{1}{2}c_0(T_0, T_0) & \text{für } k = 0 \end{cases} \\ &\stackrel{(10.8)}{=} \frac{n+1}{2} c_k \\ &= \frac{n+1}{2} \sum_{l=0}^n f(x_l) T_k(x_l) \\ &= \frac{n+1}{2} \sum_{l=0}^n f(x_l) \cos \left( k \underbrace{\arccos \left( \cos \left( \frac{2l+1}{n+1} \frac{\pi}{2} \right) \right)}_{= \frac{2l+1}{n+1} \frac{\pi}{2}} \right) \end{aligned}$$

□

$p(x)$  lässt sich als bei bekannten Koeffizienten  $c_k$  leicht berechnen/auswerten.

**Satz 10.10** (Clenshaw Algorithmus)

Sei  $p \in \mathcal{P}_n$  durch die Koeffizienten  $c_0, \dots, c_n$  in der Form

$$p(x) = \frac{1}{2}c_0 + \sum_{j=1}^n c_j T_j(x)$$

gegeben. Setzt man

$$d_{n+1} = d_{n+2} = 0$$

und definiert für  $x$

$$d_k = c_k + 2xd_{k+1} - d_{k+2}, \quad \text{für } k = n, n-1, \dots, 1, 0$$

so gilt:

$$p(x) = \frac{1}{2}(d_0 - d_2)$$

*Beweis.* Verwende die Rekursionsformel aus (10.2) iii) ( $T_{k+1} = 2xT_k + T_{k-1}$ ). Dann ist

$$\begin{aligned} p(x) &= \frac{1}{2}c_0 + \sum_{l=1}^n c_l T_l(x) \\ &= \frac{1}{2}c_0 + \sum_{l=1}^{n-3} c_l T_l(x) + c_{n-2} T_{n-2}(x) + c_{n-1} T_{n-1}(x) + c_n T_n(x) \\ &= \frac{1}{2}c_0 + \sum_{l=1}^{n-3} c_l T_l(x) + (c_{n-2} - \underbrace{c_n}_{=d_n}) T_{n-2}(x) + \underbrace{(c_{n-1} + 2xc_n)}_{=d_{n-1}} T_{n-1}(x) \\ &= \frac{1}{2}c_0 + \sum_{l=1}^{n-4} c_l T_l(x) + (c_{n-3} - d_{n-1}) T_{n-3}(x) + \underbrace{(c_{n-2} - d_n + 2xd_{n-1})}_{=d_{n-2}} T_{n-2}(x) \end{aligned}$$

induktiv erhält man

$$\begin{aligned} &= \left( \frac{1}{2}c_0 - d_2 \right) \underbrace{T_0(x)}_{=1} + \underbrace{(c_1 - d_3 + 2xd_2)}_{=d_1} \underbrace{T_1(x)}_{=x} \\ &= \frac{1}{2} \underbrace{(c_0 - 2d_1 x - d_2 - d_2)}_{=d_0} \\ &= \frac{1}{2}(d_0 - d_2) \end{aligned}$$

□

### Bemerkung

Bei der Verwendung von Rekursionen ist es wichtig zu verstehen, wie sich Rundungsfehler auswirken.

**Beispiel:**  $x_{n+1} = 10x_n - 9$ ,  $x_0 = 1$

$$\Rightarrow \forall n \in \mathbb{N} : x_n = 1$$

Was passiert bei fehlerhafter Startwerten  $\tilde{x}_0 = 1 + \varepsilon$ ?

$$\tilde{x}_{n+1} = 10\tilde{x}_n - 9, \quad \tilde{x}_n = 1 + 10^n\varepsilon$$

Der Clenshaw-Algorithmus ist stabil, wie im Folgenden gezeigt wird:

### Satz 10.11

Für den Clenshaw-Algorithmus mit Fehlern  $\varepsilon_k$  in der Rekursion, d.h. für

$$\begin{aligned}\tilde{d}_{n+1} &= \tilde{d}_{n+2} = 0 \\ \tilde{d}_k &= c_k + 2x\tilde{d}_{k+1} - \tilde{d}_{k+2} + \varepsilon_k, \quad k = n, n-1, \dots, 0\end{aligned}$$

Dabei ist  $\varepsilon_k$  der Rundungsfehler in der  $k$ -ten Iteration. Für  $\tilde{p}(x) = \frac{1}{2}(\tilde{d}_0 - \tilde{d}_2)$  gilt:

$$|\tilde{p}(x) - p(x)| \leq \sum_{j=0}^n |\varepsilon_j|, \quad \text{für } |x| < 1,$$

wobei  $p(x)$  mit (10.10) berechnet wird.

*Beweis.* Setze  $\varepsilon_k := \tilde{d}_k - d_k$  (für  $d_k$  aus (10.10)). Dann gilt:

$$\begin{aligned}\varepsilon_k &= \varepsilon_k + 2x\varepsilon_{k+1} - \varepsilon_{k+2}, \quad \text{für } k = n, \dots, 0 \\ \varepsilon_{n+1} &= 0 \quad \text{und} \quad \varepsilon_{n+2} = 0\end{aligned}$$

Mit Satz (10.10) gilt für  $c_k = \varepsilon_k$  und  $d_k = \varepsilon_k$ :

$$\frac{1}{2}(\varepsilon_0 - \varepsilon_2) = \frac{1}{2}\varepsilon_0 + \sum_{j=1}^n \varepsilon_j T_j(x)$$

Da  $|T_j(x)| \leq 1$  für  $x \in [1, 1]$  gilt:

$$|\tilde{p}(x) - p(x)| \stackrel{\Delta-UGL}{\leq} \frac{1}{2}|\varepsilon_0| + \sum_{j=1}^n |\varepsilon_j|$$

□

### Bemerkung

Die Approximation einer Funktion durch die Summe von Tschebyscheffpolynomen wird im Computer zur Berechnung von Funktionen wie  $\log$ ,  $\exp$ ,  $\sin$ ,  $\cos$ ,... verwendet.

### Beispiel 10.12

Ziel: Berechne  $\ln(x)$  für  $0 \leq x_{\min} < x \leq x_{\max}$ .  $x_{\min}, x_{\max}$  ist die kleinste/größte positive darstellbare Zahl auf dem gegebenen Computer.

$$x = \underbrace{[1, b_1, b_2, \dots, b_M]}_{\text{"Mantisse"}} * 2^N, \quad b_j \in \{0, 1\}$$

$$\text{d.h. } x = 2^N \left( 1 + b_1 \frac{1}{2} + b_2 \frac{1}{4} + \dots + b_M \frac{1}{2^M} \right) = 2^N (1 + t), \quad t \in (0, 1)$$

$$\ln(x) = \ln(1 + t) + N \underbrace{\ln(2)}_{\text{Konstante}}$$

Das Problem  $\ln(x)$  zu berechnen ist damit auf das Problem  $\ln(1 + t)$  für  $t \in [0, 1]$  zu berechnen reduziert worden.

Tschebyscheffinterpolation:  $[-1, 1] \rightarrow [0, 1]$ ,  $x \mapsto t = \frac{1+x}{2}$  ( $\Leftrightarrow x = 2t - 1$ )

Für den Interpolationsfehler gilt:

$$\begin{aligned} \ln \left( 1 + \frac{1+x}{2} \right) - p(x) &= \underbrace{\prod_{j=0}^n (x - x_j)}_{=2^{-n} \text{ für Tschebyscheff}} \frac{1}{(n+1)!} \frac{(-1)^{n-1}(n-1)!}{\left(1 + \frac{1+\xi}{2}\right)^n} \left(\frac{1}{2}\right)^n, \quad \xi \in [-1, 1] \\ \Leftrightarrow \left| \ln \left( 1 + \frac{1+x}{2} \right) - p(x) \right| &= \frac{1}{4^n} \frac{1}{(n+1)^n} \end{aligned}$$

Für  $n=15$  ist  $\frac{1}{4^n} \frac{1}{(n+1)^n} \leq 10^{-11}$

Berechnet werden also  $c_0, \dots, c_{15}$  (einmal für alle Zeiten):

$$c_0 = 0.75290562\dots$$

$$c_1 = 0.34\dots$$

$$c_2 = -0.029\dots$$

$$c_3 = 0.0036\dots$$

$$c_4 = -0.00004$$

$$|c_k| \leq 10^{-9}, \quad \text{für } k > 10$$

Beobachtung:  $c_k$  werden schnell klein.

Um eine Genauigkeit von  $10^{-8}$  (einfache Genauigkeit) zu erreichen, benötigt

man nur  $c_0, \dots, c_9$ .

Die Auswertung mit dem Clenshaw-Algorithmus benötigen wir 10 Multiplikationen (vgl. Taylor  $\log(1+t) = \sum_{k=1}^{\infty} \frac{(-1)^k}{k} t^k$ ).

## 11 Hermité-Interpolation

Gegeben sind  $(x_i, y_i, y'_i)_{i=0}^n$ ,  $x_i \in [a, b]$  paarweise verschieden. Gesucht ist ein Polynom  $p \in \mathcal{P}$ , sodass

$$\begin{aligned} p(x_i) &= y_i \quad \text{und} \\ p'(x_i) &= y'_i, \quad \text{für } i = 0, \dots, n. \end{aligned}$$

Idee: Lasse  $\varepsilon \rightarrow 0$  laufen im Newtonschem Schema:

$$\begin{array}{ll} x_0 & y_0 \\ & \delta y[x_0, x_0 + \varepsilon] = \frac{(y_0 + \varepsilon y'_0) - y_0}{(x_0 + \varepsilon) - x_0} = y'_0 \\ x_0 + \varepsilon & y_0 + \varepsilon y'_0 \\ & \delta y[x_0 + \varepsilon, x_1] \xrightarrow[\varepsilon \rightarrow 0]{} \delta y[x_0, x_1] \\ x_1 & y_1 \\ & \delta y[x_1, x_1 + \varepsilon] = y'_1 \\ x_1 + \varepsilon & y_1 + \varepsilon y'_1 \end{array}$$

Newton'sche Interpolationsformel:

$$\begin{aligned} p_\varepsilon(x) &= y_0 + (x - x_0)\delta y[x_0, x_0 + \varepsilon] \\ &\quad + (x - x_0)(x - (x_0 + \varepsilon))\delta^2 y[x_0, x_0 + \varepsilon, x_1] \\ &\quad + \dots \\ &\quad + \left( \prod_{j=0}^{n-1} (x - x_j)(x - (x_j + \varepsilon)) \right) (x - x_n) \delta^{2n+1} y[x_0, \dots, x_n] \end{aligned}$$

damit ist:

$$\begin{aligned} p_\varepsilon(x_i) &= y_i \\ p_\varepsilon(x_i + \varepsilon) &= y_i + \varepsilon y'_i \\ \Rightarrow y'_i &= \frac{p_\varepsilon(x_i + \varepsilon) - p_\varepsilon(x_i)}{\varepsilon} \underset{MWS}{=} p'_\varepsilon(\xi_i), \quad \text{für } \xi_i \in [x_i, x_i + \varepsilon] \end{aligned}$$

Für  $\varepsilon \rightarrow 0$  definieren wir

$$\delta^k y[x_0, x_0, x_1, x_1, \dots] := \lim_{\varepsilon \rightarrow 0} \delta^k y[x_0, x_0 + \varepsilon, x_1, x_1 + \varepsilon, \dots]$$

und

$$\begin{aligned} p(x) &:= \lim_{\varepsilon \rightarrow 0} p_\varepsilon(x) \\ &= y_0 + (x - x_0) \underbrace{\delta y[x_0, x_0]}_{y'_0} + (x - x_0)^2 \delta^2 y[x_0, x_0, x_1] \\ &\quad + (x - x_0)^2 (x - x_1) \delta^3 y[x_0, x_0, x_1, x_1] \\ &\quad + \dots + \prod_{j=0}^{n-1} (x - x_j)^2 (x - x_n) \delta^{2n-1} y[x_0, x_0, \dots, x_n, x_n] \\ p(x_i) &= \lim_{\varepsilon \rightarrow 0} p_\varepsilon(x_i) = y_i \\ p'(x_i) &= \lim_{\varepsilon \rightarrow 0} p'_\varepsilon(x_i) = \lim_{\varepsilon \rightarrow 0} (\xi_{i,\varepsilon}) = y'_i \end{aligned}$$

für  $\xi_{i,\varepsilon} \in [x_i, x_i + \varepsilon]$

Schema:

$x_0$	$y_0$				
	$y'_0$				
$x_0$	$y_0$	$\delta^2[x_0, x_0, x_1]$			
	$\delta y[x_0, x_1]$		$\delta^3[x_0, x_0, x_1, x_1]$		
$x_1$	$y_1$	$\delta^2[x_0, x_1, x_1]$			$\dots$
	$y'_1$		$\dots$		
$x_1$	$y_1$	$\delta^2[x_1, x_1, x_2]$			$\dots$
	$\delta y[x_1, x_2]$		$\dots$		
$x_2$	$y_2$	$\dots$			
	$y'_2$				
$x_2$	$y_2$				

Eindeutigkeit:

Annahme:  $\exists q \in \mathcal{P}_{2n+1}$  mit  $q(x_i) = y_i, q'(x_i) = y'_i$   
Dann ist  $q - p \in \mathcal{P}_{2n+1}$

$$\begin{aligned}
q - p &\text{ besitzt doppelte Nullstelle in } x_i \\
q - p &= c \prod (x - x_i)^2, \text{ da } \deg(\prod_{i=0}^n (x - x_i)^2) = 2n + 2 \\
\Rightarrow c &= 0 \quad \Rightarrow q = p
\end{aligned}$$

Damit ist der folgende Satz bewiesen.

### Satz 11.1

Zu gegebenen  $(x_i, y_i, y'_i)_{i=0}^n$  mit paarweise verschiedenen  $x_i \in [a, b]$  existiert ein eindeutiges Polynom  $p \in \mathcal{P}_{2n+1}$  mit  $p(x_i) = y_i$  und  $p'(x_i) = y'_i$  ( $i = 0, \dots, n$ ).  $p$  kann mit Hilfe des Newtonschen Differenzenschemas mit doppelten eingeschriebenen Nullstellen (Knoten) berechnet werden.

### Satz 11.2 (vgl. Satz (9.1))

Sei  $f: [a, b] \rightarrow \mathbb{R}$   $(2n+2)$ -mal stetig differenzierbar ( $f \in C^{2n+2}([a, b], \mathbb{R})$ ), seien  $x_0, \dots, x_n \in [a, b]$  paarweise verschieden und sei  $p$  Hermitépolynom aus (11.1) zu  $(x_i, y_i, y'_i)_{i=0}^n$ . Dann gilt:

$$\forall x \in [a, b] \exists \xi \in [a, b]: f(x) - p(x) = \prod_{j=0}^n (x - x_j)^2 \frac{f^{(2n+2)}(\xi)}{(2n+2)!}$$

*Beweis.* Betrachte  $\varepsilon \rightarrow 0$  für  $p_\varepsilon(x)$  in der Fehlerformel (9.1):

$$f(x) - p(x) = \prod_{j=0}^n (x - x_j)(x - (x_j + \varepsilon)) \frac{f^{(2n+2)}(\xi_\varepsilon)}{(2n+2)!}, \quad \text{für } \xi_\varepsilon \in [a, b]$$

Sei  $\xi$  ein Häufungspunkt von  $\{\xi_\varepsilon, \varepsilon > 0\}$ . Dann existiert eine Nullfolge  $(\varepsilon_k)_{k \in \mathbb{N}}$  mit  $\xi_{\varepsilon_k} \rightarrow \xi$  für  $k \rightarrow \infty$ .  $\Rightarrow$

$$\begin{aligned}
f(x) - p(x) &= \lim_{k \rightarrow \infty} (f(x) - p_{\varepsilon_k}(x)) \\
&= \prod_{j=0}^n (x - x_j)^2 \frac{f^{(2n+2)}(\xi)}{(2n+2)!}
\end{aligned}$$

□

## 12 Spline-Interpolation

Spline ist engl. für Holz- oder Metallfeder.

Theorie: stammt von Schoenberg aus dem Jahr 1946

Idee: Suche 'glatte' Funktion  $s$  durch vorgegebene Punkte  $(x_i, y_i)_{i=0}^n$

- i)  $s(x_i) = y_i$  ( $i = 0, \dots, n$ ) 'Interpolationseigenschaft'
- ii)  $s$  muss mind. 2-mal stetig differenzierbar sein und  $\int_a^b (s''(x))^2 dx$  soll minimal sein. 'glatt'

Dadurch vermeidet man Oszillationen, wie sie bei der Polynominterpolation hohen Grades entstehen.

Wir suchen also eine Funktion  $s$ , sodass für  $\varepsilon \in \mathbb{R}$  und  $h \in \mathcal{C}^2([a, b], \mathbb{R})$ ,  $h(x_i) = 0$  ( $i = 0, \dots, n$ ) und

$$\begin{aligned} \int_a^b (s''(x))^2 dx &\stackrel{!}{\leq} \int_a^b ((s(x) + \varepsilon h(x))'')^2 dx \\ &= \int_a^b (s''(x) + \varepsilon h''(x))^2 dx \\ &= \int_a^b (s''(x))^2 dx + 2\varepsilon \int_a^b s''(x)h''(x)dx + \underbrace{\varepsilon^2 \int_a^b (h''(x))^2 dx}_{\geq 0} \end{aligned}$$

Obige Ungleichung ist erfüllt, falls

$$\forall h \in \mathcal{C}^2([a, b]) \text{ mit } h(x_i) = 0 : \int_a^b h''(x)s''(x)dx = 0$$

Dabei gilt:

$$\int_a^b h''(x)s''(x)dx = [s''(x)h'(x)]_{x=a}^b - \int_a^b s'''(x)h'(x)dx$$

Falls  $s'''(x) \equiv \alpha_i$  für  $x \in [x_{i-1}, x_i]$ , dann ist

$$\begin{aligned} \int_a^b s'''(x)h'(x)dx &= \sum_{i=1}^n \alpha_i \int_{x_{i-1}}^{x_i} h'(x)dx \\ &= \sum_{i=1}^n \alpha_i \left( \underbrace{h(x_i)}_{=0} - \underbrace{h(x_{i-1})}_{=0} \right) \\ &= 0 \end{aligned}$$

$$\Rightarrow \text{Forderung: } [s''(x)h'(x)]_{x=a}^b = s''(b)h'(b) - s''(a)h'(a) \stackrel{!}{=} 0$$

### Satz 12.1

Seien  $f, s \in C^2([a, b], \mathbb{R})$  zwei Funktionen, die in  $a = x_0 < x_1 < \dots < x_n = b$  dieselben Werte annehmen, d.h.

$$f(x_i) = s(x_i) \quad (i = 0, \dots, n) \quad \text{und} \quad s|_{[x_{i-1}, x_i]} \in \mathcal{P}_3 \quad \text{für } i = 1, \dots, n$$

Falls

$$s''(a)[f'(a) - s'(a)] = s''(b)[f'(b) - s'(b)], \quad (*)$$

so gilt:

$$\int_a^b (s''(x))^2 dx \leq \int_a^b (f''(x))^2 dx$$

*Beweis.* Obige Rechnung für  $h = f - s$  und  $\varepsilon = 1$ ,  $h(x_i) = 0$   
 $[s''(x)h'(x)]_{x=a}^b = 0 \Leftrightarrow (*)$

□

### Bemerkung 12.2

Die Bedingung  $(*)$  kann erreicht werden durch

- a) Vorgabe von  $s'(a) = f'(a)$ ,  $s'(b) = f'(b)$   
 Der dadurch bestimmte Spline heißt **eingespannter** Spline.
- b) Vorgabe von  $s''(a) = 0 = s''(b)$   
 Der dadurch bestimmte Spline heißt **natürlicher** Spline. Dieser hat aber schlechtere Approximationseigenschaften.

### 12.3 (Konstruktion des Splines)

Gegeben sind  $(x_i, y_i)$   $i = 0, \dots, n$ ,  $a = x_0 < x_1 < \dots < x_n = b$ .

Suche für  $i = 1, \dots, n$   $s_i := s|_{[x_{i-1}, x_i]} \in \mathcal{P}_3$ .

Hermite-Interpolation:

$$\begin{aligned} s_i(x_i) &= y_i \\ s_i(x_{i-1}) &= y_{i-1} \\ s'_i(x_i) &= \tau_i \\ s'_i(x_{i-1}) &= \tau_{i-1} \end{aligned}$$

Dabei sind  $\tau$  unbekannte Steigungen.

Ansatz:

$$\begin{aligned}
 s_i(x) &= y_{i-1} + (x - x_{i-1})\delta y[x_{i-1}, x_i] + (x - x_{i-1})(x - x_i)(\alpha(x - x_{i-1}) + \beta(x - x_i)) \\
 s'_i(x_{i-1}) &= \delta y[x_{i-1}, x_i] + \beta(x_{i-1} - x_i)^2 \stackrel{!}{=} \tau_{i-1} \\
 s'_i(x_i) &= \delta y[x_{i-1}, x_i] + \alpha(x_i - x_{i-1})^2 \stackrel{!}{=} \tau_i \\
 \Rightarrow \alpha &= \frac{\tau_i - \delta y[x_{i-1}, x_i]}{(x_i - x_{i-1})^2} \\
 \beta &= \frac{\tau_{i-1} - \delta y[x_{i-1}, x_i]}{(x_{i-1} - x_i)^2} \\
 h_i &= x_i - x_{i-1} \\
 \Rightarrow s_i(x) &= y_{i-1} + (x - x_{i-1})\delta y[x_{i-1}, x_i] \\
 &\quad + \frac{(x - x_{i-1})(x - x_i)}{h_i^2} ((\tau_i - \delta y[x_{i-1}, x_i])(x - x_{i-1}) + (\tau_{i-1} - \delta y[x_{i-1}, x_i])(x - x_i))
 \end{aligned}$$

Für beliebige  $\tau_0, \dots, \tau_n$  erhalten wir  $s: [a, b] \rightarrow \mathbb{R}$  mit

- i)  $s|_{[x_{i-1}, x_i]} \in \mathcal{P}_3$
- ii)  $s(x_i) = y_i$
- iii)  $s \in \mathcal{C}^1([a, b])$

Bestimme  $\tau_0, \dots, \tau_n$  so, dass  $s \in \mathcal{C}^2([a, b])$ , d.h.  $s''_i(x_i) = s''_{i+1}(x_i)$  für  $i = 1, \dots, n-1$ . Das sind  $(n-1)$  Bedingungen. (#)

Beim eingespannten Spline sind  $\tau_0$  und  $\tau_n$  bekannt und die  $\tau_1, \dots, \tau_{n-1}$  sind die Unbekannten.

Mit

$$(fg)'' = f''g + 2f'g' + fg''$$

gilt wegen

$$\frac{d^2}{dx^2} ((x - x_{i-1})^2(x - x_i))|_{x=x_i} = 4h_i$$

und

$$\frac{d^2}{dx^2} ((x - x_{i-1})(x - x_i)^2)|_{x=x_i} = 2h_i$$

folgendes:

$$\begin{aligned}s_i''(x_i) &= \frac{1}{h_i^2} ((\tau_i - \delta y[x_{i-1}, x_i])4h_i + (\tau_{i-1} - \delta y[x_{i-1}, x_i])2h_i) \\ &= \frac{2}{h_i} (2\tau_i - 3\delta y[x_{i-1}, x_i] + \tau_{i-1})\end{aligned}$$

Ebenso zeigt man:

$$s_{i+1}''(x_i) = -\frac{2}{h_{i+1}} (2\tau_i - 3\delta y[x_i, x_{i+1}] + \tau_{i+1})$$

Die Bedingung (#)  $s_i''(x_i) = s_{i+1}''(x_i) \quad i = 1, \dots, n-1$  wird damit zu

$$\frac{\tau_{i-1}}{h_i} + 2\left(\frac{1}{h_i} + \frac{1}{h_{i+1}}\right)\tau_i + \frac{\tau_{i+1}}{h_{i+1}} = 3\left(\frac{\delta y[x_{i-1}, x_i]}{h_i} + \frac{\delta y[x_i, x_{i+1}]}{h_{i+1}}\right)$$

Damit erhalten wir ein LGS für  $\tau_1, \dots, \tau_{n-1}$

$$\underbrace{\begin{bmatrix} \left(\frac{2}{h_1} + \frac{2}{h_2}\right) & \frac{1}{h_2} & 0 & \dots & 0 \\ \frac{1}{h_2} & \left(\frac{2}{h_2} + \frac{2}{h_3}\right) & \frac{1}{h_3} & \ddots & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & \ddots & & \frac{1}{h_{n-1}} \\ 0 & \dots & 0 & \frac{1}{h_{n-1}} & \left(\frac{2}{h_{n-1}} + \frac{2}{h_n}\right) \end{bmatrix}}_A \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{n-1} \end{bmatrix}}_\tau = \underbrace{\begin{bmatrix} 3\left(\frac{\delta y[x_0, x_1]}{h_1} + \frac{\delta y[x_1, x_2]}{h_2}\right) - \frac{\tau_0}{h_1} \\ 3\left(\frac{\delta y[x_1, x_2]}{h_2} + \frac{\delta y[x_2, x_3]}{h_3}\right) \\ \vdots \\ 3\left(\frac{\delta y[x_{n-2}, x_{n-1}]}{h_{n-1}} + \frac{\delta y[x_{n-1}, x_n]}{h_n}\right) - \frac{\tau_n}{h_n} \end{bmatrix}}_b$$

#### Satz 12.4

Sei  $A$  wie in (12.3) und  $A\tau = b$ , dann gilt

$$\max_i |\tau_i| \leq \frac{h}{2} \max_i |b_i|,$$

wobei  $\tau = (\tau_1, \dots, \tau_{n-1})^T$ ,  $b = (b, \dots, bn-1)^T$ ,  $h = \max_i h_i$ .

*Beweis.* Sei  $j \in \{1, \dots, n-1\}$  so, dass  $|\tau_j| = \max_i |\tau_i|$ . Dann gilt:

$$\begin{aligned}
2 \left( \frac{1}{h_j} + \frac{1}{h_{j+1}} \right) \tau_j &= -\frac{\tau_{j-1}}{h_j} - \frac{\tau_{j+1}}{h_{j+1}} + b_j \\
\Rightarrow 2 \left| \frac{1}{h_j} + \frac{1}{h_{j+1}} \right| &\leq \left| \frac{\tau_{j-1}}{h_j} \right| + \left| \frac{\tau_{j+1}}{h_{j+1}} \right| + |b_j| \\
&\leq \left( \frac{1}{h_j} + \frac{1}{h_{j+1}} \right) |\tau_j| + \max_i |b_i| \\
\Rightarrow \left( \frac{1}{h_j} + \frac{1}{h_{j+1}} \right) |\tau_j| &\leq \max_i |b_i| \\
\Rightarrow \max_i |\tau_i| = |\tau_j| &\leq \frac{h}{2} \max_i |b_i|
\end{aligned}$$

□

### Korollar 12.5

Die Matrix A aus (12.4) ist invertierbar.

*Beweis.* Die einzige Lösung von  $A\tau = 0$  ist  $\tau = 0$ ,  $0 \in \mathbb{R}^{n-1}$

□

### Korollar 12.6

Der eingespannte Spline existiert und ist eindeutig.

*Beweis.* Folgt aus (12.5)

□

## 13 Fehler bei der Splineinterpolation

Vorraussetzungen für diesen Abschnitt:

$$a = x_0 < x_1 < \dots < x_n = b,$$

$$h_i = x_i - x_{i-1},$$

$$h := \max_i |h_i|$$

### Satz 13.1

Sei  $f \in C^4([a, b])$ ,  $s$  der eingespannte Spline, d.h.  $s'(a) = f'(a)$ ,  $s'(b) = f'(b)$ ,  $s(x_i) = f(x_i)$  für  $i = 0, \dots, n$ . Dann gilt für  $x \in [a, b]$

$$|f(x) - s(x)| \leq \frac{5}{384} h^4 \max_{\xi \in [a, b]} |f^{(4)}(\xi)|$$

*Beweis.* Siehe (13.3)

□

**Lemma 13.2**

Unter den Voraussetzungen von (13.1) gilt für  $s'(x_i) = \tau_i$ :

$$|f'(x_i) - \tau_i| \leq \frac{h^3}{24} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|$$

*Beweis (für den äquidistanten Fall).* Für  $i = 1, \dots, n-1$  erfüllen die  $\tau_i$

$$\frac{1}{h}(\tau_{i-1} + 4\tau_i + \tau_{i+1}) = \frac{3}{h^2}(f(x_{i+1}) - f(x_{i-1})) = b_i$$

Ersetze nun  $\tau_i$  durch  $f'(x_i)$ , so gilt:

$$\frac{1}{h}(f'(x_{i-1}) + 4f'(x_i) + f'(x_{i+1})) - \frac{3}{h^2}(f(x_{i+1}) - f(x_{i-1})) =: \delta_j$$

Taylorentwicklungen von  $f'(x_{i-1}), f'(x_{i+1}), f(x_{i-1}), f(x_{i+1})$  um  $x_i$

$$\begin{aligned} f(x_{i+1}) &= f(x_i + h) = f(x_i) + hf'(x_i) + \frac{h^2}{2!}f''(x_i) \\ &\quad + \frac{h^3}{3!}f'''(x_i) + h^4 \int_0^1 \frac{(1-t)^3}{3!} f^{(4)}(x_i + th) dt \\ f'(x_{i+1}) &= f'(x_i) + hf''(x_i) + \frac{h^2}{2!}f''(x_i) \\ &\quad + h^3 \int_0^1 \frac{(1-t)^2}{2!} f^{(4)}(x_i + th) dt \end{aligned}$$

und analog für  $f(x_{i-1}) = f(x_i - h)$  und  $f'(x_{i-1})$ .  $\Rightarrow$

$$\begin{aligned}
\delta_j &= \frac{1}{h} (f'(x_i) - h f''(x_i) + \frac{h^2}{2} f''(x_i) + R'_{i-} \\
&\quad + 4f'(x_i) + f'(x_i) + h f''(x_i) + \frac{h^2}{2} f'''(x_i) + R'_{i+}) \\
&\quad - \frac{3}{h^2} (f(x_i) + h f'(x_i) + \frac{h^2}{2} f''(x_i) + \frac{h^3}{3!} f'''(x_i) + R_{i+}) \\
&\quad - f(x_i) + h f'(x_i) - \frac{h^2}{2} f''(x_i) + \frac{h^3}{3!} f'''(x_i) + R_{i-}) \\
&= h^2 \int_0^1 \left( \frac{(1-t)^2}{2!} - 3 \frac{(1-t)^3}{3!} \right) f^{(4)}(x+th) dt \\
&\quad + h^2 \int_0^1 \left( \frac{(1-t)^2}{2!} - 3 \frac{(1-t)^3}{3!} \right) f^{(4)}(x-th) dt \\
&= h^2 (f^{(4)}(\xi_i) + f^{(4)}(\eta_i)) \int_0^1 \frac{(1-t)^2}{2} - \frac{(1-t)^3}{2} dt \\
&= \frac{h^2}{24} (f^{(4)}(\xi_i) + f^{(4)}(\eta_i)), \quad \xi_i \in [x_{i-1}, x_i], \eta_i \in [x_i, x_{i+1}] \\
\Rightarrow |\delta_i| &\leq \frac{h^2}{12} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|
\end{aligned}$$

Definiere nun  $e_i := f'(x_i) - \tau_i$  für  $i = 0, \dots, n$ . Diese erfüllen die Bedingung  $e_0 = 0$ ,  $e_n = 0$  vom eingespannten Spline. Für  $f' = (f'(x_1), \dots, f'(x_{n-1}))^T$ ,  $\delta = (\delta_1, \dots, \delta_{n-1})^T$  und  $e = (e_1, \dots, e_{n-1})$  gilt:  $A\tau = b$  und  $Af' = b + \delta$ . Mit (12.4) gilt dann

$$\max_i |e_i| \leq \frac{h}{2} \max_i |\delta_i| \leq \frac{h^3}{24} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|$$

□

### 13.3 (Beweis von (13.1))

Für  $x \in [x_{i-1}, x_i]$  ist  $f(x) - s_i(x) = f(x) - p_i(x) + p_i(x) - s_i(x)$ , wobei  $p_i$  das kubische Hermiteinterpolationspolynom zu  $f$  ist mit  $p_i(x_i) = f(x_i)$ ,  $p_i(x_{i-1}) = f(x_{i-1})$ ,  $p'_i(x_i) = f'(x_i)$ ,  $p'_i(x_{i-1}) = f'(x_{i-1})$

Nach Satz (11.2) gilt für ein  $\xi \in [x_{i-1}, x_i]$ :

$$\begin{aligned}|f(x) - p_i(x)| &= |(x - x_i)^2(x - x_{i-1})^2| \left| \frac{f^{(4)}(\xi)}{24} \right| \\ &\leq \frac{h^4}{16 * 24} |f^{(4)}(\xi)| = \frac{h^4}{384} |f^{(4)}(\xi)|\end{aligned}$$

Weiter gilt:

$$\begin{aligned}s_i(x) - p_i(x) &= (x - x_{i-1})(x - x_i)((\tau_i - f'(x_i))(x - x_{i-1}) \\ &\quad + (\tau_{i-1} - f'(x_{i-1}))(x - x_i)) \frac{1}{h^2}\end{aligned}$$

Da  $\frac{(x-x_{i-1})(x-x_i)}{h^2} \leq \frac{1}{4}$  für  $x \in [x_{i-1}, x_i]$  gilt mit (13.2)

$$\begin{aligned}|s_i(x) - p_i(x)| &\leq \frac{1}{4} \frac{h^3}{24} \max_{\xi \in [a,b]} |f^{(4)}(\xi)| \underbrace{(|x - x_{i-1}| + |x - x_i|)}_{=h} \\ &= \frac{h^4}{96} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|\end{aligned}$$

Ingesamt gilt also:

$$|f(x) - s_i(x)| \leq h^4 \frac{1+4}{384} \max_{\xi \in [a,b]} |f^{(4)}(\xi)|$$

□

### Bemerkung

Wie wirken sich Störungen/Fehler in den Daten auf den interpolierenden Spline aus?

Gegeben seien  $(x_i, y_i)_{i=0}^n$  und  $(y'_0, y'_n)$ . Dadurch erhält man einen Spline  $s(x)$ . Für Daten  $(x_i, \tilde{y}_i)_{i=0}^n$  und  $(y'_0, y'_n)$  erhält man einen Spline  $\tilde{s}(x)$ . Der Einfachheit halber sind  $y'_0$  und  $y'_n$  fehlerfrei.

Nun gilt:

$$s(x) - \tilde{s}(x) = \sum_{i=0}^n (y_i - \tilde{y}_i) l_i(x),$$

wobei  $l_i(x)$  ein kubischer Spline mit

$$l_i(x_j) = \begin{cases} 1 & , \text{ falls } i = j \\ 0 & , \text{ sonst} \end{cases}$$

und  $l'_i(a) = 0 = l'_i(b)$  ist ("Lagrange-Spline").

Diese zeigen keine Oszillationen wie Lagrange-Polynome auf äquidistanten Stützstellen. Es gilt

$$\max_{x \in [a,b]} |s(x) - \tilde{s}(x)| \leq \Lambda_n \max_i |y_i - \tilde{y}_i|$$

mit der Spline Lebesguekonstante

$$\Lambda_n = \max_{x \in [a,b]} \sum_{i=0}^n |l_i(x)|$$

Ohne Beweis: Für äquidistante Verteilungen gilt für Splines  $\forall n \in \mathbb{N} : \Lambda_n \leq 2$

## 14 Numerische Differentiation

**Problemstellung:** Zu  $f : [a, b] \rightarrow \mathbb{R}$  berechne näherungsweise  $f'(x)$  für  $x \in [a, b]$ :

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Falls  $f \in \mathcal{C}^2([a, b])$  gilt:

$$\begin{aligned} f(x+h) &= f(x) + h f'(x) + \frac{h^2}{2} f''(\xi), \quad \text{für } \xi \in [a, b] \\ \Rightarrow \frac{f(x+h) - f(x)}{h} &= f'(x) + \frac{h}{2} f''(\xi) \end{aligned}$$

Allerdings ist ein Grenzübergang  $h \rightarrow 0$  auf einem Computer problematisch, da statt  $\frac{f(x+h)-f(x)}{h}$  nur  $\frac{f(x+h)-f(x)+\varepsilon}{h}$  berechnet werden kann

für ein  $\varepsilon < \text{eps}$  (Maschinengenauigkeit)

$\text{eps} \approx 10^{-16}$

**Idee:** Um  $f'$  zu approximieren, ersetze  $f$  durch ein Polynom  $p$  oder ein Spline  $s$  und approximiere  $f'$  durch  $s'$  oder  $p'$ .

**Berechnung von  $p'(x)$ :** Dividierte Differenzen für  $n = 3$ :

$$\begin{array}{ll}
x & p(x) = b_0 \\
& \quad \quad \quad b_1 \\
x_0 & y_0 = f(x_0) \quad \quad \quad b_2 \\
& \quad \quad \quad \delta^1 y[x_0, x_1] \quad \quad \quad b_3 \\
x_1 & y_1 = f(x_1) \quad \quad \quad \delta^2 y[x_0, x_1, x_2] \quad \parallel \\
& \quad \quad \quad \delta^1 y[x_1, x_2] \quad \quad \quad \delta^3 y[x_0, x_1, x_2, x_3] \\
x_2 & y_2 = f(x_2) \quad \quad \quad \delta^2 y[x_1, x_2, x_3] \\
& \quad \quad \quad \delta^1 y[x_2, x_3] \\
x_3 & y_3 = f(x_3)
\end{array}$$

Interpolationspolynom  $p \in \mathcal{P}_n$ :

$$\begin{aligned}
p(x) &= \sum_{i=0}^n \prod_{j=0}^{i-1} (x - x_i) \delta^i y[x_0, \dots, x_i] \\
&= x^n \delta^n y[x_0, \dots, x_n] + r, \quad \text{für } r \in \mathcal{P}_{n-1} \\
p^{(n)} &\equiv n! \delta^n y[x_0, \dots, x_n]
\end{aligned}$$

Füge weitere Diagonale zu Knoten  $x$  in obigem Schema hinzu mit  $b_0 = p(x)$  und  $b_k = \delta^k y[x, x_0, x_1, \dots, x_{k-1}]$ . Nach Definition ist

$$b_{k+1} = \frac{b_k - \delta^k y[x_0, \dots, x_k]}{x - x_k}$$

Rechne nun im Newtonschema von rechts nach links (da  $b_n = \delta^n y[x_0, \dots, x_n]$ ).

```

 $b_n = \delta^n y[x_0, \dots, x_n]$ 
for  $k = n-1, \dots, 0$  do
   $b_k = b_{k+1}(x - x_k) + \delta^k y[x_0, \dots, x_k]$ 
end for
 $p(x) = b_0$ 

```

Nach dem Horner-Schema.

Berechne nun die Ableitungen:

Füge weitere Diagonale zu Knoten  $x + \varepsilon$  hinzu und lasse  $\varepsilon \rightarrow 0$  laufen

$$\begin{array}{ll}
x + \varepsilon & p(x + \varepsilon) = c_0 \\
x & p(x) = b_0 \\
x_0 & y_0 = f(x_0) \\
x_1 & y_1 = f(x_1) \\
x_2 & y_2 = f(x_2) \\
x_3 & y_3 = f(x_3)
\end{array}
\quad
\begin{array}{ll}
c_1 = p'(x) \\
b_1 \\
\delta^1 y[x_0, x_1] \\
\delta^1 y[x_1, x_2] \\
\delta^1 y[x_2, x_3]
\end{array}
\quad
\begin{array}{ll}
c_2 \\
b_2 \\
\delta^2 y[x_0, x_1, x_2] \\
\delta^2 y[x_1, x_2, x_3]
\end{array}
\quad
\begin{array}{ll}
c_3 \\
b_3 \\
\parallel \\
\parallel
\end{array}$$

Algorithmus zur Berechnung von  $p'(x)$ :

---

```

 $c_n = b_n$ 
for  $k = n - 1, \dots, 1$  do
     $c_k = b_k + (x - x_{k-1})c_{k+1}$ 
end for
 $p'(x) = c_1$ 

```

### Satz 14.1

Sei  $f \in C^{n+2}([a, b])$ ,  $p$  Interpolationspolynom zu  $f$  in  $x_0, \dots, x_n \in [a, b]$  paarweise verschieden ( $p \in \mathcal{P}_n$ ).

$\forall x \in [a, b] \exists \xi, \xi' \in [a, b] :$

$$f'(x) - p'(x) = \left( \sum_{i=0}^n \prod_{\substack{j=0 \\ j \neq i}}^n (x - x_j) \frac{f^{(n+1)}(\xi)}{(n+1)!} \right) + \prod_{j=0}^n (x - x_j) \frac{f^{(n+2)}(\xi')}{(n+2)!}$$

*Beweisskizze.* (vgl. 9.1)

Sei  $\bar{x}$  fest aber beliebig,  $\bar{p}$  das Hermiteinterpolationspolynom zu

$$\bar{p}(x_i) = f(x_i), \quad i = 0, \dots, n$$

$$\bar{p}(x) = f(x),$$

$$\bar{p}'(x) = f'(x)$$

Newtonschema und Newtoninterpolationspolynome liefert das Ergebnis.  $\square$

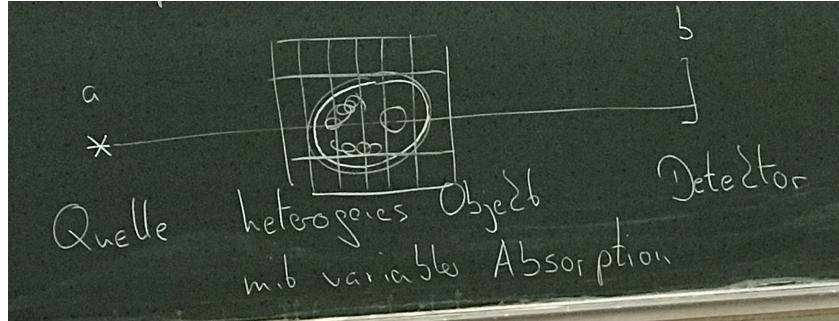
### III Lineare Gleichungssysteme und lineare Ausgleichsrechnung

- Ziele:**
- Berechne  $x \in \mathbb{R}^n$ , welches Lösung von  $Ax = b$  ist, wobei  $A \in \mathbb{R}^{n \times n}$  invertierbar und  $b \in \mathbb{R}^n$ .
  - Berechne  $x \in \mathbb{R}^m$ , welches Lösung von  $\min_{x \in \mathbb{R}^m} \|Ax - b\|_2$  ist, wobei  $A \in \mathbb{R}^{n \times m}$ ,  $b \in \mathbb{R}^n$  und  $n > m$ .

#### 15 Gaußelimination

##### Beispiel 15.1

- Splineinterpolation  $A\tau = b$ ,  $A$  tridiagonal und symmetrisch.
- Computertomographie



$\Delta I$ : gemessener Intensitätsunterschied zwischen Quelle und Detektor  
 $\Delta I = \int_{[a,b]} \alpha(x) dx$

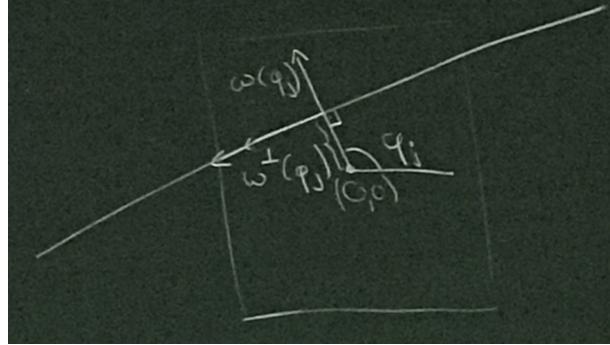
Dabei ist  $\alpha(x)$  der Absorptionskoeffizient

Annahme:  $\alpha$  ist konstant in jeder Volumenzelle (Voxel)  $\Rightarrow$

$$\Delta I = \sum_{j \in \text{Voxel}} \alpha_j l_j$$

$l_j$ : Länge des Weges  $[a, b]$  in Voxel  $j$

Viele Strahlen:  $L_j(t) = \omega(\varphi_j)s_j + \omega^\perp(\varphi_j)t$



$$\begin{aligned}\omega(\varphi_j) &= (\cos(\varphi_j), \sin(\varphi_j)) \\ \omega^\perp(\varphi_j) &= (-\sin(\varphi_j), \cos(\varphi_j))\end{aligned}$$

$$\begin{bmatrix} l_{11} & l_{12} & \dots & l_{1M} \\ \vdots & \vdots & & \vdots \\ l_{N1} & l_{N2} & \dots & l_{NM} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{bmatrix} = \begin{bmatrix} \Delta I_1 \\ \vdots \\ \Delta I_N \end{bmatrix}$$

$M$  ist dabei die Anzahl der Voxel.

$l_{ij}$ : Länge des i-ten Strahl im j-ten Voxel

$\alpha_j$ : Absorption im j-ten Voxel

$\Delta I_i$ : Intensitätsunterschied entlang vom Strahl i

### 15.2 (Herleitung des Verfahrens (Wdh. LA))

$$Ax = b, A = (a_{ij})_{i,j=1}^n, b = (b_i)_{i=1}^n$$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$\vdots$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

Ohne Einschränkungen sei  $a_{11} \neq 0$ . Da  $A$  invertierbar ist, ist mindestens ein Element aus  $\{a_{i1}, i = 1, \dots, n\}$  ungleich 0. Man kann also Zeilen/Gleichungen so vertauschen, dass  $a_{11} \neq 0$ . Für  $i = 2, 3, \dots, n$  multipliziere die i-te Zeile mit  $l_{i1} := \frac{a_{i1}}{a_{11}}$  und ersetze die i-te Zeile durch (i-te Zeile)  $- l_{i1} * (1\text{-ste Zeile})$ . Dann ergibt sich folgendes Gleichungssystem:

$$a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \dots + a_{1n}^{(1)}x_n = b_1^{(1)}$$

$$0 + a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}$$

$\vdots$

$$0 + a_{n2}^{(1)}x_2 + \cdots + a_{nn}^{(1)}x_n = b_n^{(1)}$$

Dabei ist

$$a_{1j}^{(1)} = a_{1j} \text{ für } j = 1, \dots, n,$$

$$b_1^{(1)} = b_1,$$

$$(a_{i1}^{(1)} = 0)$$

$$a_{ij}^{(1)} = a_{ij} - l_{i1}a_{1j},$$

$$b_i^{(1)} = b_i - l_{i1}b_1 \text{ für } i = 2, \dots, n, j = 1, \dots, n.$$

Da die  $(n-1) \times (n-1)$  Untermatrix  $A^{(1)}(2:n, 2:n)$  ebenfalls invertierbar ist, wiederholt man den eben beschriebenen Schritt.

Nach eventuellem Zeilentausch ist  $a_{22}^{(1)} \neq 0$

$$\begin{aligned} l_{i2} &:= \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}, \quad i = 3, \dots, n \\ b_2^{(2)} &= b_2^{(1)}, \quad a_{2j}^{(2)} = a_{2j}^{(1)}, \quad j = 2, \dots, n \\ b_i^{(2)} &= b_i^{(1)} - l_{i2}b_2^{(1)}, \quad i = 3, \dots, n \\ a_{ij}^{(2)} &= a_{ij}^{(1)} - l_{i2}a_{2j}^{(1)}, \quad i = 3, \dots, n, j = 2, \dots, n \end{aligned}$$

Damit entsteht eine Folge

$$(A, b), (A^{(1)}, b^{(1)}), (A^{(2)}, b^{(2)}), \dots, (A^{(n-1)}, b^{(n-1)}) =: (R, c)$$

für eine obere Dreiecksmatrix  $R$  (d.h. alle Einträge unter der Diagonalen sind 0).

Das Gleichungssystem mit  $(r_{ii} \neq 0)$

$$r_{11}x_1 + r_{12}x_2 + \cdots + r_{1n}x_n = c_1$$

$$r_{22}x_2 + \cdots + r_{2n}x_n = c_2$$

$$\vdots$$

$$r_{nn}x_n = c_n$$

Dabei ist

$$x_n = \frac{c_n}{r_{nn}}$$

$$x_i = \frac{1}{r_{ii}}(c_i - \sum_{j=i+1}^n r_{ij}x_j) \text{ für } i = n-1, \dots, 1$$

### Satz 15.3

Für eine invertierbare Matrix  $A \in \mathbb{R}^{n \times n}$  liefert das in (15.2) beschriebene Verfahren

$$PA = LR,$$

wobei

$$L = \begin{bmatrix} 1 & & & 0 \\ l_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ l_{n1} & \dots & l_{n(n-1)} & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & \ddots & & \vdots \\ 0 & & & r_{nn} \end{bmatrix}$$

und  $P$  eine Permutationsmatrix ist.

*Beweis.* Nehme an, dass die notwendige Zeilenvertauschungen bereits durchgeführt wurden, d.h. ersetze  $A$  durch  $PA$  (Zeilen und Spalten von  $P$  bestehen aus kanonischen Einheitsvektoren z.B.  $P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ ).

Bezeichne mit  $L_i \in \mathbb{R}^{n \times n}$ :

$$L_i = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & 1 & \\ & & -l_{i+1,i} & \\ & & \vdots & \ddots \\ & & -l_{n,i} & 1 \\ & \underbrace{\quad}_{\text{i-te Spalte}} & & \end{bmatrix}$$

Damit ist

$$\begin{aligned} A^{(1)} &= L_1 A, \quad a_{ij}^{(1)} = a_{ij} - l_{i1} a_{1j} \\ A^{(k)} &= L_k A^{(k-1)}, \quad k = 2, \dots, n-1 \\ R &= A^{(n-1)} = L_{n-1} L_{n-2} \dots L_1 A \\ \Rightarrow A &= \underbrace{L_1^{-1} \dots L_{n-2}^{-1} L_{n-1}^{-1}}_{=L} R \end{aligned}$$

Setzt man

$$V_i = \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & l_{i+1,i} & & \\ & & \vdots & \ddots & \\ & & l_{n,i} & & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

i-te Spalte

so ist  $L_i = I_n - V_i$ , da  $V_i V_k = 0$  für  $i \leq k$ .

$$\underbrace{(I_n - V_i)(I_n + V_i)}_{=L_i} = I_n + V_i - V_i + \underbrace{V_i V_i}_{=0} = I_n$$

d.h.  $L_i^{-1} = I_n + V_i$ .

Damit folgt  $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} = (I_n + V_1)(I_n + V_2) \dots (I_n + V_{n-1}) = I_n + V_1 + V_2 + \dots + V_{n-1} + \underbrace{V_1 V_2 + \dots}_{=0} = L$

Das schließende  $L$  ist dabei das  $L$  aus (15.2).  $\square$

### Bemerkung

$$\det(PA) = \det(P) \det(A) = (-1)^{\# \text{ Vertauschungen}} \det(A)$$

$$\det(PA) = \det(LR) = \underbrace{\det(L)}_{=1} \det(R) = \prod_{i=1}^n r_{ii}$$

### Algorithmus 15.4 (Vorwärts- und Rückwärts-Substitution)

Sobald man die LR-Zerlegung (lu-decomposition) von  $A$  kennt, löst man  $Ax = b$  wie folgt:

$$Pb = PAx = L \underbrace{Rx}_{=:c}$$

Löse  $Lc = Pb$  ("Vorwärtssubstitution"):

```

 $c_1 = (Pb)_1$ 
for  $i = 2, \dots, n$  do
   $c_i = (Pb)_i - \sum_{j=1}^{i-1} l_{ij} c_j$ 
end for

```

und anschließend:

Löse  $Rx = c$  wie in (15.2) angegeben ("Rückwärtssubstitution").

### 15.5 (Aufwand)

Beim Schritt  $A \rightarrow A^{(1)}$  benötigt man

- $(n - 1) \in \mathcal{O}(n)$  Divisionen
- $(n - 1)^2 \in \mathcal{O}(n^2)$  Multiplikationen
- $(n - 1)^2 \in \mathcal{O}(n^2)$  Additionen

Also insgesamt Operationen aus  $\mathcal{O}(n^2)$ .

$A^{(1)} \rightarrow A^{(2)}$ :  $(n - 1)^2$  Operationen.

$A^{(2)} \rightarrow A^{(3)}$ :  $(n - 2)^2$  Operationen.

$\vdots$

$$A \rightarrow L, R: \sum_{j=1}^n j^2 \approx \frac{1}{3}n^3 (\in \mathcal{O}(n^3)), \text{ da } \underbrace{\frac{1}{n} \sum_{j=0}^n \left(\frac{j}{n}\right)^2 n^3}_{\approx \int_0^1 x^2 = \frac{1}{3}}$$

Die Lösung von  $Lc = Pb$  kostet ebenso wie die Lösung von  $Rx = c$

$1 + 2 + \dots + (n - 1) \approx \frac{1}{2}n^2$  Operationen.

Der Hauptaufwand steckt also in der Berechnung der Zerlegung  $PA = LR$ .

**Bemerkung** (Einschub zur Gleitkommarechnung (floating point arithmetic))

Jeder reelle Zahl  $0 \neq x$  kann für festes  $B \in \mathbb{N}$ ,  $B \geq 2$  eindeutig durch

$$x = \pm m B^e$$

dargestellt werden, wobei  $m \in [1, B)$ , die Mantisse,  $e \in \mathbb{Z}$ , der Exponent und  $B$  die Basis ist.

Durch den Computer kommen folgende Einschränkungen hinzu:

- Es stehen nur  $l$  Ziffern für die Mantisse  $m$  zur Verfügung  $\rightarrow m$  wird gerundet.
- Es stehen nur  $r$  Ziffern für den Exponenten zur Verfügung

### Definition

Eine  $l$ -stellige-Basis- $B$ -Gleitkommazahl mit Exponentialbereich  $[e_{\min}, e_{\max}]$  ist ein Tripel  $(\sigma, m, e)$ . Dabei ist

- $\sigma$  das Vorzeichen
- $m$  eine  $l$ -stellige Zahl zur Basis  $B$  mit festgelegter Kommastelle
- $e$  die ganze Zahl in  $[e_{\min}, e_{\max}]$

Der Wert von  $(\sigma, m, e)$  ist  $\sigma * m * B^e$ .

### Beispiel

Betrachte den Standard IEEE 754.

Dieser stellt eine Zahl mit einfacher Genauigkeit dar zur Basis  $B = 2$  mit  $l = 32$  und  $[e_{\min}, e_{\max}] = [-128, 127]$ .

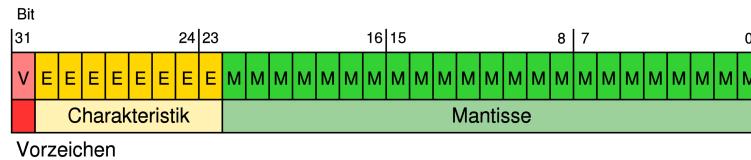


Figure 1: From [https://de.wikipedia.org/wiki/IEEE\\_754](https://de.wikipedia.org/wiki/IEEE_754)

Der Wert lässt sich berechnen aus

$$(-1)^\sigma * (1, m) * 2^{\sum_j e_j^{2^j} - 127}$$

Dabei ist  $m$  binär dargestellt.

### Definition

Für eine reelle Zahl  $x$  bezeichnen wir mit  $fl(x)$  eine  $l$ -stellige-Basis-10-Darstellung von  $x$  mit unbeschränktem Exponenten  $e$ , sodass

$$fl(x) = \pm m 10^e,$$

wobei  $m$  eine Zahl mit  $l$  Stellen ist.

Die Maschinengenauigkeit  $\text{eps}$  ist die kleinste positive Zahl, sodass  $fl(1 + \text{eps}) > 1$ . Also ist  $\text{eps}$  der Abstand zwischen zwei benachbarten Mantissen.

### Beispiele

Dezimalsystem ( $B = 10$ )  $\Rightarrow$   $\text{eps} = 5 * 10^{-e}$

Binärsystem ( $B = 2$ )  $\Rightarrow$   $\text{eps} = 2^{-e}$

## 16 Wahl des Pivotelements

### Beispiel 16.1

$$\begin{aligned} 10^{-4}x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned}$$

exakte Lösung:

$$\begin{aligned} x_1 &= \frac{1}{0.9999} = 1.0001\overline{0001} \\ x_2 &= \frac{0.9998}{0.9999} = 0.9998\overline{9998} \end{aligned}$$

bei dreistelliger dezimaler Gleitkommaartihmetik (Mantissenlänge 3, Basis 10)

$$\begin{aligned} 0.100 * 10^{-3}x_1 + 0.100 * 10^1x_2 &= 0.100 * 10^1 \\ 0.100 * 10^1x_1 + 0.100 * 10^1x_2 &= 0.200 * 10^1 \end{aligned}$$

und damit erhält man für

a)  $a_{11} = 10^{-4}$  (Pivot)

$$\begin{aligned} l_{21} &= \frac{a_{21}}{a_{11}} = 10^4 = 0.100 * 10^5 \\ a_{22}^{(1)} &= 0.100 * 10^1 - 0.100 * 10^5 = -0.100 * 10^5 \\ b_2^{(1)} &= 0.200 * 10^1 - 0.100 * 10^5 = -0.100 * 10^5 \end{aligned}$$

Aus  $-0.100 * 10^5x_2 = -0.100 * 10^5$  folgt  $x_2 = 0.100 * 10^1 = 1$   
 $\Rightarrow x_1 = \frac{b_1 - a_{12}x_2}{a_{11}} = \frac{0.100 * 10^1 - 0.100 * 10^1 * 1}{0.100 * 10^1} = 0$

b) Wähle Pivot  $a_{21} = 1$ :

$$\begin{aligned} x_1 + x_2 &= 2 \\ 10^{-4}x_1 + x_2 &= 1 \end{aligned}$$

...  $l_{21} = 10^{-4} \Rightarrow x_2 = 1, x_1 = 1$

### Erläuterung:

Falls  $|l_{21}|$  groß ist, ergibt sich

$$a_{22}^{(1)} = a_{22} - l_{21}a_{12} \approx l_{21}a_{12}$$

$$b_2^{(1)} = b_2 - l_{21}b_1 \approx l_{21}b_1$$

$$x_2 = \frac{b_2^{(1)}}{a_{22}^{(1)}} \approx \frac{b_1}{a_{21}}$$

Bei der Berechnung von  $x_1$  kommt es zu einer Stellenauslöschung  $x_1 = (b_1 - \underbrace{a_{12}x_2}_{b_1}) : a_{11} \approx 0$

Ausweg: Zeilentausch, sodass  $|a_{21}| \leq |a_{11}|$ . Dann ist  $|l_{21}| \leq 1$ .

Spaltenpivotsuche:

Nehme Pivotelement im  $(k+1)$ -ten Schritt

$$a_{j(k+1)}^{(k)} \quad \text{mit} \quad |a_{j(k+1)}^{(k)}| = \max_{i=k+1, \dots, n} |a_{i(k+1)}^{(k)}|,$$

d.h. das betragsmäßig größte Element der  $(k+1)$ -ten Spalte von  $A^{(k)}$  unterhalb der Diagonalen inklusive des Diagonalelements.

Damit erreicht man

$$|l_{i,(k+1)}| = \frac{|a_{i,k+1}^{(k)}|}{|a_{k+1,k+1}^{(k)}|} \leq 1, \quad i = k+2, \dots, n$$

## 17 Cholesky-Zerlegung für symmetrische positiv definite Matrizen

### Definition 17.1

Eine Matrix  $A = (a_{ij})_{i,j=1}^n \in \mathbb{R}^{n \times n}$  heißt symmetrisch, falls  $\forall i, j = 1, \dots, n : a_{ij} = a_{ji}$ , d.h.  $A = A^T$ .

Eine Matrix  $A$  ist positiv definit, falls

$$\forall x \in \mathbb{R}^n : x^T Ax > 0$$

### Satz 17.2

Sei  $A$  symmetrisch positiv definit (kurz: spd),  $A \in \mathbb{R}^{n \times n}$ . Dann gilt:

- i) Die Gaußelimination kann ohne Zeilenvertauschungen durchgeführt werden
- ii) Für die Zerlegung  $A = LR$  gilt  $R = DL^T$  für eine Diagonalmatrix

$$D = \begin{bmatrix} r_{11} & & 0 \\ & \ddots & \\ 0 & & r_{nn} \end{bmatrix}, \quad \text{wobei } \forall i = 1, \dots, n : r_{ii} > 0$$

*Beweis.* Es gilt:  $a_{11} = e_1^T A e_1 > 0$ , da  $A$  spd, wobei  $e_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^n$  der 1. kanonische Basisvektor ist. Also ist  $a_{11}$  ein möglicher Pivot. Schreibe  $A$  nun als:

$$\left[ \begin{array}{c|c} a_{11} & z^T \\ \hline z & C \end{array} \right]$$

wobei  $z = (a_{21}, \dots, a_{n1})^T \in \mathbb{R}^{n-1}$  und  $C$  eine symmetrische  $(n-1) \times (n-1)$ -Matrix ist.

Nun ist

$$A^{(1)} = \left[ \begin{array}{c|c} a_{11} & z^T \\ \hline 0 & C^{(1)} \end{array} \right]$$

$C^{(1)}$  ist symmetrisch, da

$$c_{ij}^{(1)} = a_{ij} - \underbrace{\frac{a_{i1}}{a_{11}}}_{=l_{i1}} a_{1j} = a_{ji} - \underbrace{\frac{a_{j1}}{a_{11}}}_{=l_{j1}} a_{1i} = c_{ji}^{(1)}$$

Also ist  $C^{(1)}$  insbesondere spd, da für  $\begin{pmatrix} x_1 \\ \vdots \\ y \\ \vdots \end{pmatrix} \neq 0$  gilt:

$$0 < \left( \begin{pmatrix} x_1 \\ \vdots \\ y \\ \vdots \end{pmatrix}^T A \begin{pmatrix} x_1 \\ \vdots \\ y \\ \vdots \end{pmatrix} \right) = \left( \begin{pmatrix} x_1 \\ \vdots \\ y \\ \vdots \end{pmatrix} \left[ \begin{array}{c|c} a_{11} & z^T \\ z & C \end{array} \right] \begin{pmatrix} x_1 \\ \vdots \\ y \\ \vdots \end{pmatrix} \right) = a_{11}x_1^2 + \underbrace{y^T z x_1 + x_1 z^T y}_{=2x_1 y^T z} + y^T C y$$

Weiter ist:

$$y^T C^{(1)} y = y^T C y - \frac{1}{a_{11}} y^T z z^T y = y^T C y - \frac{1}{a_{11}} (y^T z)^2$$

Wählt man nun  $x_1 = -\frac{y^T z}{a_{11}}$ , so gilt:

$$\begin{aligned} 0 &< a_{11} \left( -\frac{y^T z}{a_{11}} \right)^2 - 2 \frac{y^T z}{a_{11}} y^T z + y^T C y \\ &= -\frac{(y^T z)^2}{a_{11}} + y^T C y \\ &= y^T C^{(1)} y \end{aligned}$$

für beliebiges  $y \in \mathbb{R}^{n-1} \setminus \{0\}$ .

Induktiv folgt dann  $a_{22}^{(1)} > 0$   $C^{(2)}$  spd, ...

Zeige nun noch ii):

Es gilt

$$l_{i1} = \frac{a_{i1}}{a_{11}} = \frac{a_{i1}}{r_{11}} = \frac{r_{1i}}{r_{11}}$$

da  $r_{1i} = a_{1i} = a_{i1}$  für  $i = 2, \dots, n$ .

Außerdem gilt

$$l_{i2} = \frac{a_{i2}}{a_{22}} = \frac{a_{i2}}{r_{22}} = \frac{r_{2i}}{r_{22}}$$

da  $r_{2i} = a_{2i}^{(1)} = a_{i2}^{(1)}$  für  $i = 3, \dots, n$ .

Allgemein gilt also

$$\forall i > j : l_{ij} = \frac{r_{ji}}{r_{jj}},$$

wobei  $r_{ii} = a_{ii}^{(i-1)} > 0$  und  $r_{ij} = l_{ij} * r_{jj} = r_{jj} * l_{ij}$ , d.h.  $R = DL^T$  für

$$D = \begin{bmatrix} r_1 & & 0 \\ & \ddots & \\ 0 & & r_{nn} \end{bmatrix}$$

Es wird die i-te Zeile von  $L^T$  mit  $r_{ii}$  skaliert.  $\square$

### Bemerkung

Wegen  $r_{ii} > 0$  ist  $D = D^{1/2}D^{1/2}$  mit

$$D^{1/2} = \begin{bmatrix} \sqrt{r_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{r_{nn}} \end{bmatrix}$$

Damit erhält man für  $\tilde{L} = LD^{1/2}$  (Spaltenskalierung)

$$A = LDL^T = LD^{1/2}D^{1/2}L^T = (LD^{1/2})(LD^{1/2})^T = \tilde{L}\tilde{L}^T$$

Wir bezeichnen die Elemente von  $\tilde{L}$  wieder mit  $l_{ij}$ :

$$\tilde{L} = \begin{bmatrix} l_{11} & & 0 \\ \vdots & \ddots & \\ l_{1n} & \dots & l_{nn} \end{bmatrix}$$

Diese  $l_{ij}$ 's lassen sich direkt aus der Gleichung  $A = \tilde{L}\tilde{L}^T$  berechnen.

$$\begin{bmatrix} l_{11} & & 0 \\ \vdots & \ddots & \\ l_{1n} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \dots & l_{n1} \\ & \ddots & \vdots \\ 0 & & l_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & & \\ a_{12} & a_{22} & a_{32} & \\ & & \ddots & \\ & & & a_{nn} \end{bmatrix}$$

Nun folgt:

$$\begin{aligned} l_{11}^2 = a_{11} > 0 &\Rightarrow l_{11} = \sqrt{a_{11}} \\ l_{11}l_{i1} = a_{i1} &\Rightarrow l_{i1} = \frac{a_{i1}}{l_{11}} \end{aligned}$$

allgemein gilt:

$$a_{kk} = l_{k1}^2 + l_{k2}^2 + \dots + l_{k,k-1}^2 + l_{kk}^2$$

$$l_{kk} = \left( a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 \right)^{1/2}$$

und für  $i > k$ :

$$a_{ik} = l_{i1}l_{k1} + l_{i2}l_{k2} + \dots + l_{ik-1}l_{kk-1} + l_{ik}l_{kk}$$

$$l_{ik} = \frac{\left( a_{ik} - \sum_{j=1}^{k-1} l_{ij}l_{kj} \right)}{l_{kk}}$$

Choleski-Verfahren:

```

for  $k = 1, \dots, n$  do
     $l_{kk} = \left( a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 \right)^{1/2}$ 
    for  $i = k+1, \dots, n$  do
         $l_{ik} = (a_{ik} - \sum_{j=1}^{k-1} l_{ij}l_{kj}) / l_{kk}$ 
    end for
end for

```

Nun stellen sich folgende zwei Fragen:

- Wie wirken sich Störungen in  $A$  und  $b$  auf die Lösung von  $Ax = b$  aus?
- Wie wirken sich Rundungsfehler im Verfahren auf die berechnete Lösung aus?

## 18 Matrixnormen

### Definition 18.1

Die Abbildung  $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}^n, x \mapsto \|x\|$  ist eine Norm auf dem  $\mathbb{R}$ -Vektorraum  $\mathbb{R}^n$ , falls gilt:

- $\forall x \in \mathbb{R}^n : \|x\| \geq 0$
- $\|x\| = 0 \Rightarrow x = 0 \in \mathbb{R}^n$
- $\forall \alpha \in \mathbb{R} \forall x \in \mathbb{R}^n : \|\alpha x\| = |\alpha| \|x\|$

iv)  $\|x + y\| \leq \|x\| + \|y\|$

### Beispiel 18.2

i)  $\|x\|_1 = \sum_{j=1}^n |x_j|$  für  $x = (x_1, \dots, x_n)^T$

ii)  $\|x\|_2 = \left( \sum_{j=1}^n |x_j|^2 \right)^{1/2}$  oder allgemein  $\|x\|_p = \left( \sum_{j=1}^n |x_j|^p \right)^{1/p}$  für  $1 \leq p < \infty$

iii)  $\|x\|_\infty = \max_{i=1,\dots,n} |x_i|$

### Definition 18.3

Sei  $A \in \mathbb{R}^{m \times n}$ , d.h.  $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$  linear. Nun heißt

$$\|A\| := \sup_{\substack{x \in \mathbb{R}^n \\ x \neq 0}} \frac{\|Ax\|}{\|x\|}$$

die von den Vektorraumnormen induzierte Norm.

### Bemerkung 18.4

Sei  $A \in \mathbb{R}^{n \times m}$ ,  $\alpha \in \mathbb{R}$ . Es gilt für die in (18.3) definierte Matrixnorm

i)  $\forall x \in \mathbb{R}^n : \|Ax\| \leq \|A\| \|x\|$

$\|A\|$  ist die kleinste Zahl mit dieser Eigenschaft.

ii) Es gilt  $\|A\| \geq 0$ . Weiter gilt  $\|A\| = 0 \Rightarrow A = 0$

iii)  $\|\alpha A\| = |\alpha| \|A\|$

iv)  $\|A + B\| \leq \|A\| + \|B\|$ .

Damit ist  $\|\cdot\|$  tatsächlich eine Norm.

v)  $\|I\| = 1$  falls  $m = n$ ,  $\|\cdot\|_{\mathbb{R}^m} = \|\cdot\|_{\mathbb{R}^n}$

vi)  $\|AB\| \leq \|A\| \|B\|$  (Submultiplikativität)

### Satz 18.5

Sei  $A = (a_{ij})_{i,j=1}^{m,n} \in \mathbb{R}^{m \times n}$ . Es gilt für  $\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$  für  $p \geq 1$

i)  $\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^m |a_{ij}|$  ist die maximale Spaltenbetragssumme

ii)  $\|A\|_2$  ist die Wurzel des größten Eigenwerts von  $A^T A$

iii)  $\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^n |a_{ij}|$  ist die maximale Zeilenbetragssumme

*Beweis.*

i) + iii) Übungsaufgabe

ii)  $A^T A$  ist symmetrisch und positiv semidefinit. Es gilt nämlich

$$(A^T A)^T = A^T A^{T^T} = A^T A$$

und

$$x^T A^T A x = (Ax)^T A x = \|Ax\|_2^2 \geq 0$$

Damit ist  $A^T A$  orthogonal diagonalisierbar, d.h. es ex.  $Q$  mit  $Q^T Q = I$  sodass  $Q^T A^T A Q = D$  mit

$$D = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}$$

wobei  $\lambda_j \geq 0$  ( $j = 1, \dots, m$ ) die Eigenwerte von  $A^T A$  sind.  
Damit ist

$$\begin{aligned} \|Ax\|_2^2 &= x^T A^T A x \underset{x=Qy}{=} y^T Q^T A^T A Q y = \sum_{j=1}^m \lambda_j y_j^2 \\ &\leq \lambda_{\max} \sum_{j=1}^m y_j^2 = \lambda_{\max} y^T y = \lambda_{\max} \|y\|_2^2 = \lambda_{\max} \|x\|_2^2 \end{aligned}$$

$\Rightarrow \|A\|_2 \leq \sqrt{\lambda_{\max}}$  für den größten Eigenwert  $\lambda_{\max}$  von  $A^T A$ .

Sei  $\tilde{x} = Q\tilde{y}$  mit  $\tilde{y} = (0, \dots, 0, \underset{j_0\text{-ter Eintrag}}{1}, 0, \dots, 0)^T$  mit  $\lambda_{j_0} = \lambda_{\max}$ . Dann ist  $\|A\tilde{x}\|_2^2 = \lambda_{\max} \|\tilde{x}\|_2^2 \Rightarrow \|A\|_2 = \sqrt{\lambda_{\max}}$ .

□

## 19 Kondition eines Problems

### Definition 19.1

Seien  $X, Y$  normierte Vektorräume  $(X, \|\cdot\|_X), (Y, \|\cdot\|_Y)$ . Ein Problem bzw. eine Problemstellung ist eine Abbildung  $f: X \rightarrow Y$ , wobei  $X$  die Eingaben und  $Y$  die Ausgaben enthält.

### Beispiel 19.2

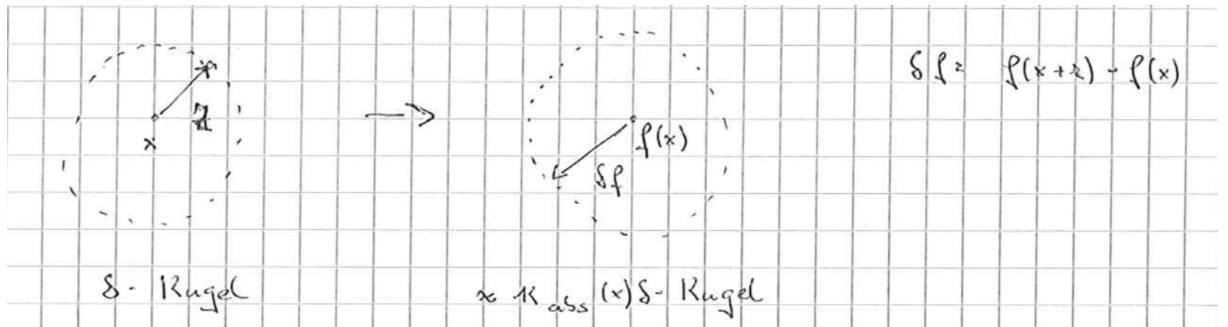
Sei  $X = Y = \mathbb{R}^2$  und  $\|\cdot\|_X = \|\cdot\|_Y = \|\cdot\|_2$ .

- i)  $f: (x_1, x_2) \mapsto A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$  für eine  $2 \times 2$  Matrix  $A$ . "Anwendung der linearen Abbildung  $A$ "
- ii)  $f: (p, q) \mapsto$  Wurzeln von  $z^2 + pz + q = 0$ . "Berechnung der Wurzeln eines normierten quadratischen Polynoms"

### Definition 19.3 (Absolute Kondition)

Seien  $X, Y$  normierte Vektorräume  $f: X \rightarrow Y$  ein Problem. Die **absolute Kondition** von  $f$  in  $x \in X$  ist

$$\kappa_{\text{abs}}(f, x) := \lim_{\delta \rightarrow 0} \sup_{\|z\|_X \leq \delta} \frac{\|f(x+z) - f(x)\|_Y}{\|z\|_X}$$



### Lemma 19.4

Sei  $f: (\mathbb{R}, \|\cdot\|) \rightarrow (\mathbb{R}, \|\cdot\|)$  differenzierbar, so gilt

$$\kappa_{\text{abs}}(f, x) = |f'(x)|$$

*Beweis.* Übungsaufgabe. □

### Definition 19.5 (Relative Kondition)

Seien  $X, Y$  normierte Räume,  $f : X \rightarrow Y$  ein Problem. Die **relative Kondition** von  $f$  in  $x \in X$  ist

$$\kappa_{\text{rel}}(f, x) := \lim_{\delta \rightarrow 0} \sup_{\|z\|_X \leq \delta} \frac{\frac{\|f(x+z) - f(x)\|_Y}{\|f(x)\|_Y}}{\frac{\|z\|_X}{\|x\|_X}}$$

d.h.  $\kappa_{\text{rel}}(f, x)$  ist die kleinste Zahl sodass

$$\underbrace{\frac{\|f(x) - f(x+z)\|_Y}{\|f(x)\|_Y}}_{\begin{array}{l} \text{relativer Fehler in der Ausgabe, wenn man} \\ \text{statt } f(x) \text{ das Problem } f(x+z) \text{ gelöst hat} \end{array}} \leq \kappa_{\text{rel}}(f, x) \underbrace{\frac{\|z\|_X}{\|x\|_X}}_{\begin{array}{l} \text{relativer Fehler in der Eingabe} \end{array}}$$

### Beispiel 19.6

Kondition der Addition:

$$f : (\mathbb{R}^2, \|\cdot\|_1) \rightarrow (\mathbb{R}, |\cdot|), (a, b) \mapsto a + b$$

$$\kappa_{\text{abs}}(f, (a, b)) = \lim_{\delta \rightarrow 0} \sup_{|\alpha|+|\beta| \leq \delta, z=(\alpha, \beta)} \frac{|a + \alpha + b + \beta - a - b|}{|\alpha| + |\beta|} = 1$$

$$\kappa_{\text{rel}}(f, (a, b)) = \dots = \frac{|a| + |b|}{|a + b|}$$

d.h. für die Addition zweier Zahlen mit gleichem Vorzeichen ist  $\kappa_{\text{rel}} = 1$ . Für Subtraktion zweier annähernd gleich großer Zahlen ist  $\kappa_{\text{rel}}$  groß.

### Definition 19.7

Ein Problem heißt **gut konditioniert**, falls  $\kappa_{\text{rel}}$  klein ist ( $< 10^3$ ) und **schlecht konditioniert**, falls  $\kappa_{\text{rel}}$  groß ist ( $> 10^8$ ).

## 20 Konditionszahl einer Matrix

Gegeben sei  $Ax = b$ . Welchen Einfluss haben Fehler in  $A$  und in  $b$  auf die Lösung  $x$ ?

In Form von §19:  $f : (A, b) \mapsto x$ . Statt  $a_{ij}$  stehen nun  $\tilde{a}_{ij} = a_{ij}(1 + \varepsilon_{ij})$  und statt  $b_i$  nun  $b_i(1 + \varepsilon_i)$  zur Verfügung. Also  $\tilde{A}\tilde{x} = \tilde{b}$ .

### Satz 20.1

Sei  $A$  invertierbar,  $A \in \mathbb{R}^{n \times n}$ ,  $Ax = b$ ,  $\tilde{A}\tilde{x} = \tilde{b}$ ,  $x \neq 0$ .  
Falls

$$\frac{\|A - \tilde{A}\|}{\|A\|} \leq \varepsilon_A, \quad \frac{\|b - \tilde{b}\|}{\|b\|} \leq \varepsilon_b$$

so gilt für die Lösung des LGS:

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \varepsilon_A \text{cond}(A)} (\varepsilon_A + \varepsilon_b)$$

falls  $\varepsilon_A * \text{cond}(A) < 1$ . Hierbei ist  $\text{cond}(A) = \|A\| \|A^{-1}\|$  die Konditionszahl von  $A$  und die Matrixnorm wird von der Vektorraumnorm induziert (vgl. 18.3).

*Beweis.*

$$\begin{aligned} b - \tilde{b} &= Ax - \tilde{A}\tilde{x} \\ &= Ax - A\tilde{x} + A\tilde{x} - \tilde{A}\tilde{x} \\ &= A(x - \tilde{x}) + (A - \tilde{A})\tilde{x} \\ \Rightarrow x - \tilde{x} &= A^{-1}(b - \tilde{b} - (A - \tilde{A})\tilde{x}) \\ \|x - \tilde{x}\| &\leq \|A^{-1}\| (\|b - \tilde{b}\| \varepsilon_b + \varepsilon_A \|A\| \|\tilde{x}\|) \\ &\stackrel{b = Ax}{\leq} \text{cond}(A) (\|x\| \varepsilon_b + \varepsilon_A (\|\tilde{x} - x\| + \|x\|)) \\ \Rightarrow (1 - \text{cond}(A)\varepsilon_A) \|x - \tilde{x}\| &\leq \text{cond}(A) \|x\| (\varepsilon_b + \varepsilon_A) \end{aligned}$$

□

### Bemerkung 20.2

Die Abschätzung aus (20.1) ist scharf, d.h. es gibt  $\tilde{A}$  und  $\tilde{b}$ , sodass Gleichheit gilt. Aber sie ist oft zu pessimistisch für Rundungsfehlerabschätzungen.

**Beispiel:** Betrachte folgendes Gleichungssystem:

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 0 & 10^{-8} \end{pmatrix}}_A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Sei des Weiteren  $|\varepsilon_j| < \text{eps}$ .

Es gilt  $\text{cond}_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty = 2 * 10^8$ . Das gestörte System ist nun:

$$(1 + \varepsilon_1)\tilde{x}_1 + (1 + \varepsilon_2)\tilde{x}_2 = \overbrace{b_1(1 + \varepsilon_3)}^{=\tilde{b}_1} \\ (1 + \varepsilon_4)10^{-8}\tilde{x}_2 = b_2(1 + \varepsilon_5)$$

Mit  $\frac{1}{1+\varepsilon} = 1 - \varepsilon + \varepsilon^2 - \varepsilon^3 + \varepsilon^4 - \dots$  folgt

$$\Rightarrow \tilde{x}_2 = 10^8 b_2 \frac{1 + \varepsilon_5}{1 + \varepsilon_4} = 10^8 b_2 (1 + \varepsilon_5 - \varepsilon_4) + \mathcal{O}(\text{eps}^2) \\ \Rightarrow \frac{|x_2 - \tilde{x}_2|}{|x_2|} \leq 2\text{eps}$$

$$\tilde{x}_1 = [b_1(1 + \varepsilon_3) - x_2(1 + \varepsilon_5 - \varepsilon_4 + \varepsilon_3)](1 - \varepsilon_1) + \mathcal{O}(\text{eps}^2) \\ = [x_1 + \underbrace{b_1}_{=x_1+x_2} \varepsilon_3 - x_2(\varepsilon_5 - \varepsilon_4 + \varepsilon_2)](1 - \varepsilon_1) + \mathcal{O}(\text{eps}^2)$$

$$\tilde{x}_1 - x_1 = x_1(-\varepsilon_1 + \varepsilon_3) - x_2(-\varepsilon_3 + \varepsilon_5 - \varepsilon_4 + \varepsilon_2) + \mathcal{O}(\text{eps}^2) \\ \frac{|\tilde{x}_1 - x_1|}{|x_1|} \leq (2 + 4 \frac{|x_2|}{|x_1|})\text{eps}$$

Dieser Wert kann sehr groß werden für  $\frac{|x_2|}{|x_1|} \rightarrow \infty$ , aber  $\frac{|\tilde{x}_1 - x_1|}{\|x_1\|_\infty} \leq 6\text{eps}$ .

### Lemma 20.3

Sei  $A \in \mathbb{R}^{n \times n}$  invertierbar. Dann gilt:

- i)  $\text{cond}(A) \geq 1$
- ii)  $\forall \alpha \in \mathbb{R} \setminus \{0\} : \text{cond}(\alpha A) = \text{cond}(A)$
- iii)  $\text{cond}(A) = \frac{\max_{\|y\|=1} \|Ay\|}{\min_{\|x\|=1} \|A^{-1}x\|}$

*Beweis.* Übungsaufgabe. □

### Beispiel 20.4

1) Matrizen mit kleiner Konditionszahl:

- $I$  mit  $\text{cond}(I) = 1$
- orthogonale Matrizen  $Q$  ( $Q^T Q = I$ )

$$\|Qx\|_2^2 = x^T Q^T Q x = x^T x = \|x\|_2^2 \Rightarrow \|Q\|_2 = 1$$

$$Q^{-1} = Q^T \Rightarrow \|Q^{-1}\|_2 = 1 \Rightarrow \text{cond}_2(Q) = 1$$

- Splineinterpolationsmatrix ( $h_i = h$ )

$$A = \frac{1}{h} \begin{bmatrix} 4 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & 1 & \\ 0 & & 1 & 4 \end{bmatrix}, \quad \|A\|_\infty = 6$$

$$A = 4(I + N) \quad \text{mit } N = \begin{bmatrix} 0 & 1/4 & & 0 \\ 1/4 & \ddots & \ddots & \\ & \ddots & 1/4 & \\ 0 & & 1/4 & 0 \end{bmatrix}$$

$$A^{-1} = 1/4(I + N)^{-1} = 1/4(I - N + N^2 - N^3 + \dots), \quad \|N\|_\infty = 1/2$$

$$\|A^{-1}\| \leq 1/4(\|I\| + \|N\| + \|N\|^2 + \dots) = 1/2$$

2) Matrizen mit großer Konditionszahl:

- Hilbertmatrix  $H = \left( \frac{1}{i+j-1} \right)_{i,j=1}^n$

$$H = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots \\ 1/2 & 1/3 & 1/4 & & \\ 1/3 & 1/4 & & & \\ 1/4 & & & & \\ \vdots & & & & \end{bmatrix}$$

Für  $n \in \{1, \dots, 10\}$  ergibt sich folgende Tabelle:

$n$	$\text{cond}_2(A)$
1	1
2	27
3	740
4	2300
$\vdots$	$\vdots$
10	$35 * 10^{13}$

## 21 Stabilität von Verfahren

### Definition 21.1

Ein Verfahren zur Auswertung eines Problems  $f$  ist die Hintereinanderausführung von elementaren Operationen  $\tilde{f}_k$   
 $\tilde{f} = \tilde{f}_n \circ \tilde{f}_{n-1} \circ \dots \circ \tilde{f}_1, \quad \tilde{f}_k \in \{+, -, *, /, fl, \sqrt{\cdot}, \dots\}$

### Definition 21.2

Ein Verfahren zur Auswertung des Problems  $f$  ist stabil im Sinne der Vorwärtsanalyse, falls

$$\|\tilde{f}(x) - f(x)\| < C * \text{eps} * \|f(x)\|$$

für eine nicht zu große Konstante  $C$ .

### Beispiel 21.3

Berechnung von  $\frac{1}{x(x-1)}$  für  $x = 10^4$ .

$$\begin{aligned} 1) \quad x &\xrightarrow{\quad fl(x) \quad} fl(x(x-1)) \rightarrow fl\left(\frac{1}{x(x-1)}\right) \\ &\xrightarrow{\quad fl(x-1) \quad} \\ 2) \quad x &\xrightarrow{\quad fl(x) \quad} fl\left(\frac{1}{x}\right) \rightarrow fl\left(-\frac{1}{x} + \frac{1}{x-1}\right) \\ &\xrightarrow{\quad fl(x-1) \quad} fl\left(\frac{1}{x-1}\right) \end{aligned}$$

Verfahren 2) ist nicht stabil, da  $\frac{1}{x} \approx \frac{1}{x-1}$ , falls  $x = 10^4$  gilt und die Subtraktion im letzten Schritt schlecht konditioniert ist.

### Definition 21.4

Ein Verfahren  $\tilde{f}$  zur Auswertung eines Problems  $f$  ist stabil im Sinne der

Rückwärtsanalyse, falls für jedes  $x \in X$  ein  $\tilde{x} \in X$  existiert, sodass

$$\tilde{f}(x) = f(\tilde{x}) \quad \text{mit} \quad \frac{\|\tilde{x} - x\|}{\|x\|} \leq C * \text{eps}$$

für eine nicht zu große Konstante  $C$ . Die berechnete Lösung  $\tilde{f}(x)$  kann als exakte Lösung eines benachbarten Problems  $f(\tilde{x})$  aufgefasst werden.

### Beispiel 21.5

Zur Berechnung von  $x_1x_2 + x_3x_4$  verwendet man:

$$\begin{array}{ccccc} & \nearrow & x_1x_2 & \searrow & \\ (x_1, x_2, x_3, x_4) & & + & & x_1x_2 + x_3x_4 \\ & \searrow & x_3x_4 & \nearrow & \end{array}$$

und erhält unter Berücksichtigung von Rundungsfehlern

$$[(x_1(1 + \varepsilon_1)x_2(1 + \varepsilon_2))(1 + \eta_1) + (x_3(1 + \varepsilon_3)x_4(1 + \varepsilon_4))(1 + \eta_2)](1 + \eta_3) \quad \text{für } |\varepsilon_j|, |\eta_j| \leq \text{eps}$$

Das ist das exakte Ergebnis für

$$\begin{aligned} \tilde{x}_1 &= x_1(1 + \varepsilon_1)(1 + \eta_1)(1 + \eta_3) \\ \tilde{x}_2 &= x_2(1 + \varepsilon_2)(1 + \eta_1)(1 + \eta_3) \\ \tilde{x}_3 &= x_3(1 + \varepsilon_3)(1 + \eta_2)(1 + \eta_3) \\ \tilde{x}_4 &= x_4(1 + \varepsilon_4)(1 + \eta_2)(1 + \eta_3) \end{aligned}$$

Die Konstante  $C$  in (21.4) ist etwa 3, wenn man Produkte von  $\varepsilon_j$  und  $\eta_j$  vernachlässigt. Das Verfahren ist also rückwärtsstabil, auch wenn evtl. das Problem schlecht konditioniert ist.

### Satz 21.6 (Stabilität der Gaußelimination (LR-Zerlegung))

Sei  $A \in \mathbb{R}^{n \times n}$  invertierbar und  $\hat{L}\hat{R}$  das rundungsfehlerbehaftete Ergebnis der Gaußelimination mit Pivotisierung, sodass  $|\hat{l}_{ij}| \leq 1$  für alle  $i, j \in \{1, \dots, n\}$ .

Dann gilt für  $\hat{A} = (\hat{a}_{ij})_{i,j=1}^n = \hat{L}\hat{R}$ :

$$|a_{ij} - \hat{a}_{ij}| \leq 2 \max_{i,j,k} |a_{ij}^{(k)}| * \min\{i-1, j\} * \text{eps}$$

für Maschinengenauigkeit  $\text{eps}$ .

*Beweis.* Im k-ten Schritt berechnet man ausgehend von  $\hat{a}_{ij}^{(k-1)}$

$$\begin{aligned} \hat{a}_{ij}^{(k)} &= \left( \hat{a}_{ij}^{(k-1)} - \hat{l}_{ik}\hat{a}_{kj}^{(k-1)}(1 + \varepsilon_{ijk}) \right) (1 + \eta_{ijk}) \\ &= \hat{a}_{ij}^{(k-1)} - \hat{l}_{ik}\hat{a}_{kj}^{(k-1)} + \mu_{ijk} \quad (*) \end{aligned}$$

mit  $|\varepsilon_{ijk}|, |\eta_{ijk}| \leq \text{eps}$  und  $\mu_{ijk} \leq |\hat{a}_{ij}^{(k-1)}| |\eta_{ijk}| + |\hat{l}_{ik}| |\hat{a}_{kj}^{(k-1)}| |\varepsilon_{ijk}| + \mathcal{O}(\text{eps}^2)$ . (\*\*)

Nach Definition von  $\hat{A}$  ist

$$\hat{a}_{ij} = \sum_{k=1}^{\min\{i,j\}} \hat{l}_{ik} \hat{r}_{kj} = \sum_{k=1}^{\min\{i,j\}} \hat{l}_{ik} \hat{a}_{kj}^{(k-1)}$$

Verwendet man für  $i > j$  (\*), so erhält man

$$\hat{a}_{ij} = \sum_{k=1}^j \left( \hat{a}_{ij}^{(k-1)} - \hat{a}_{ij}^{(k)} + \mu_{ijk} \right) = a_{ij} + \sum_{k=1}^j \mu_{ijk}, \quad \text{da } a_{ij}^{(j)} = 0$$

Für  $i \leq j$  erhält man

$$\hat{a}_{ij} = \sum_{k=1}^{i-1} \left( \hat{a}_{ij}^{(k-1)} - \hat{a}_{ij}^{(k)} + \mu_{ijk} \right) + \hat{l}_{ii} a_{ij}^{(i-1)} = a_{ij} + \sum_{k=1}^{i-1} \mu_{ijk}, \quad \text{da } \hat{l}_{ii} = 1$$

Zusammen mit (\*\*) folgt die Behauptung.  $\square$

### Bemerkung

Aus dem Satz (21.6) kann man entnehmen, dass die Gaußelimination im Sinne der Rückwärtsanalyse stabil ist, falls

$$\frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}$$

nicht zu groß wird. Dieser Quotient ist meistens klein. Mehr kann man nicht beweisen.

## 22 QR-Zerlegung mit Hilfe der Householdertransformationen

Ziel: Konstruiere zu einer gegebenen Matrix  $A \in \mathbb{R}^{m \times n}$  ( $m \geq n$ ) eine Zerlegung  $A = QR$  mit einer orthogonalen Matrix  $Q \in \mathbb{R}^{m \times m}$  ( $Q^T Q = I_m$ ) und

$$R = \underbrace{\begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}}_n \quad \begin{array}{l} \}n \\ \}m-n \end{array}$$

Dabei ist  $\tilde{R} \in \mathbb{R}^{n \times n}$  eine obere Dreiecksmatrix.

Anwendungen:

- a)  $m = n$ ,  $Ax = b$ ,  $Qc = b$ ,  $Rx = c$ ,  $Q^{-1} = Q^T$ . Besonders stabiler Algorithmus (stabiler als Gauß). Dafür doppelt so teuer.
- b)  $m > n \rightarrow$  lineare Ausgleichsrechnung (siehe §23)
- c) QR-Algorithmus zur Berechnung von Eigenwerten (siehe Numerik II)

**Definition 22.1**

Für  $v \in \mathbb{R}^m$ ,  $\|v\|_2 = 1$  heißt

$$Q = I - 2vv^T$$

**Householderreflexion** zum Vektor  $v$ .

**Satz 22.2**

Für eine Householderreflexion  $v \in \mathbb{R}^m$ ,  $\|v\|_2 = 1$ ,  $Q = I - 2vv^T$  gilt:

- i)  $Q$  ist symmetrisch
- ii)  $Q$  ist orthogonal
- iii)  $Qv = -v$
- iv)  $Qw = w$  für alle  $w \in \mathbb{R}^m$  mit  $w^T v = 0$

Mit iii) und iv) erhält man, dass  $Q$  eine Spiegelung an der Hyperebene  $\{x \in \mathbb{R}^m : x^T v = 0\}$  ist.

*Beweis.*

- i)  $Q^T = (I - 2vv^T)^T = I - 2vv^T = Q$
- ii)  $Q^T Q \stackrel{i)}{=} (I - 2vv^T)(I - 2vv^T) = I - 4vv^T + 4v \underbrace{(v^T v)}_{=1} v^T = I$
- iii)  $Qv = (I - 2vv^T)v = v - 2v \underbrace{v^T v}_{=1} = -v$
- iv)  $Qw = (I - 2vv^T)w = w - 2v \underbrace{v^T w}_{=0} = w$

□

### 22.3 (Algorithmus der QR-Zerlegung)

Sei  $A = \begin{bmatrix} \vdots & \vdots & & \vdots \\ a_1 & a_2 & \dots & a_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$  mit  $a_j$  als die j-te Spalte von  $A$ .

$$1) \text{ Suche } Q_1 a_1 = \begin{bmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha_1 e_1$$

Da  $Q_1$  orthogonal ist, gilt  $\|a_1\|_2^2 = \|Q_1 a_1\|_2^2 = |\alpha_1|^2$

$\Rightarrow \alpha_1 = \pm \|a_1\|_2$  (Vorzeichen noch nicht fest)

$$Q_1 a_1 = \alpha_1 e_1$$

$$Q_1 a_1 = (I - 2u_1 u_1^T) a_1 = a_1 - 2u_1 \underbrace{u_1^T a_1}_{\in \mathbb{R}}$$

$\Rightarrow u_1$  ist ein Vielfaches von  $a_1 - \alpha_1 e_1$

Mit der Forderung  $\|u_1\|_2 = 1$  ergibt sich

$$u_1 = \frac{a_1 - \alpha_1 e_1}{\|a_1 - \alpha_1 e_1\|_2}$$

$$\text{Dabei ist } \|a_1 - \alpha_1 e_1\|_2^2 = \underbrace{\|a_1\|_2^2}_{=\alpha_1^2} - 2\alpha_1 \underbrace{e_1^T a_1}_{a_{11}} + \alpha_1^2 = 2\alpha_1(\alpha_1 - a_{11}).$$

Man wählt das Vorzeichen von  $\alpha_1$  so, dass keine Stellenauslösung bei der Berechnung von  $\alpha_1 - a_{11}$  auftritt

$$\alpha_1 = -\text{sgn}(a_{11}) \|a_1\|_2$$

Es ist dann

$$Q_1 A = A^{(1)} = \left[ \begin{array}{c|c} \alpha_1 & * \\ \hline 0 & \\ \vdots & B \\ 0 & \end{array} \right]$$

wobei die j-te Spalte von  $A^{(1)}$  (Nenne diese  $a_j^{(1)}$ ) durch

$$a_j^{(1)} = Q_1 a_j = a_j - \frac{2v_1^T a_j}{v_1^T v_1} v_1, \quad j = 2, \dots, n$$

für  $v_1 = a_1 - \alpha_1 e_1$  gegeben ist. Weiter gilt

$$\frac{v_1^T v_1}{2} = \frac{1}{2} \|a_1 - \alpha_1 e_1\|_2^2 = \alpha_1(\alpha_1 - a_{11})$$

Bemerkung: Zur Berechnung von  $Q_1 A$  reicht es  $v_1$  und  $\alpha$  zu kennen.  
Die Matrix  $Q_1 = I - 2u_1 u_1^T$  wird nicht aufgestellt.

- 2) Suche nun  $\tilde{Q}_2 = I_{m-1} - 2u_2 u_2^T$  mit  $u_2 \in \mathbb{R}^{m-1}$ , sodass  $\tilde{Q}_2 b_1 = \alpha_2 e_1$  ( $e_1 \in \mathbb{R}^{m-1}$ ).

Für  $Q_2 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \tilde{Q}_2 & \\ 0 & & & \end{bmatrix}$  ist dann  $Q_2 A^{(1)} = \left[ \begin{array}{cc|c} \alpha_1 & * & * \\ 0 & \alpha_2 & \\ \vdots & 0 & \\ 0 & 0 & C \end{array} \right]$

usw.

im k-ten Schritt hat man dann

$$Q_k = \begin{bmatrix} I_{k-1} & \\ & \tilde{Q}_k \end{bmatrix}$$

mit dem Ergebnis  $Q_n Q_{n-1} \cdots Q_1 A = R$  mit

$$R = \underbrace{\begin{bmatrix} \alpha_1 & & * \\ & \ddots & \\ 0 & & \alpha_n \\ & & 0 \end{bmatrix}}_n}_{m-n} \quad n$$

## 22.4 (Rechenaufwand)

**Schritt 1:**  $\approx 2 * m * n (*, +)$  Operationen

Gesamtaufwand:  $2(m * n + (m-1)(n-1) + \dots + 1)$

Falls  $m = n \rightarrow \frac{2}{3} n^3$

Falls  $m \gg n \rightarrow 2m(n + (n-1) + \dots + 1) \approx mn^2$

## 22.5 (Stabilität)

$$A = QR$$

Da  $Q$  orthogonal ist, gilt  $\|A\|_2 = \|QR\|_2 = \|R\|_2$   
 $\Rightarrow$  Einträge von  $R$  können nicht groß werden.

Für das berechnete  $\hat{Q}$  gilt  $\|\hat{Q}^T \hat{Q} - I\| < c * \text{eps}$  für eine kleine Konstante  $c$ , d.h.  $\hat{Q}$  ist fast orthogonal.

Man kann zeigen, dass

$$\|A - \hat{Q}\hat{R}\|_2 < c \|A\|_2 \text{eps}$$

## Bemerkung 22.6

QR-Zerlegung ist bis zum Ende durchführbar, falls  $\text{rang}(A) = n$  (dann ist  $\alpha_1, \alpha_2, \dots, \alpha_n \neq 0$ , da  $\text{rang}(A) = \text{rang}(R)$ )

Falls  $\text{rang}(A) = k < n$  wäre bei der Rechnung ohne Rundungsfehler  $\alpha_l = 0$  für ein  $l$  und das Verfahren bricht ab. Modifizierte den Algorithmus deswegen:

- Schritt:** Berechne  $\|a_1\|_2, \dots, \|a_n\|_2$  die Spaltennormen und vertausche die Spalten, sodass  $\|a_1\|_2$  maximal wird:

$$Q_1 A P_1 = \begin{bmatrix} \alpha_1 & & \\ 0 & & \\ \vdots & * & \\ 0 & & \end{bmatrix}, \quad |\alpha_1| = \|a_1\|_2$$

usw. mit Spaltenvertauschungen in weiteren Schritten.

$\Rightarrow |\alpha_1| \geq |\alpha_2| \geq \dots \geq |\alpha_n| > 0$ . Falls  $\text{rang}(A) = k < n$  erhält man  $\alpha_{k+1} = 0$  und

$$\left[ \begin{array}{cc|c} \alpha_1 & R_1 & \\ \ddots & & R_2 \\ 0 & \alpha_k & \end{array} \right]$$

$AP = QR$  (numerische Rangentscheidung)

falls  $\frac{|\alpha_{k+1}|}{|\alpha_1|} < 100\text{eps}$  setze  $\text{Rang}(A) = k$ .

## 23 Lineare Ausgleichsrechnung

### Problem 23.1

Zu gegebenen Messdaten  $(t_j, y_j)_{j=1}^m$  suche  $y = f(t)$

$$f(t) = \sum_{i=1}^n x_i \varphi_i(t),$$

sodass  $y_j \approx f(t_j)$  für  $j = 1, \dots, m$ .

Hierbei sind die  $\varphi_1, \dots, \varphi_n$  gegebene Funktionen und  $x_1, \dots, x_n$  unbekannte Parameter, wobei  $m \gg n$ .

Genauer:  $\sum_{j=1}^m (y_j - f(t_j))^2$  soll minimal werden.

### 23.2 (Matrix-Vektor Formulierung)

Es sei

$$A = \begin{bmatrix} \varphi_1(t_1) & \cdots & \varphi_n(t_1) \\ \varphi_1(t_2) & \cdots & \varphi_n(t_2) \\ \vdots & & \vdots \\ \varphi_1(t_m) & \cdots & \varphi_n(t_m) \end{bmatrix} \in \mathbb{R}^{m \times n}$$

$$b = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$$

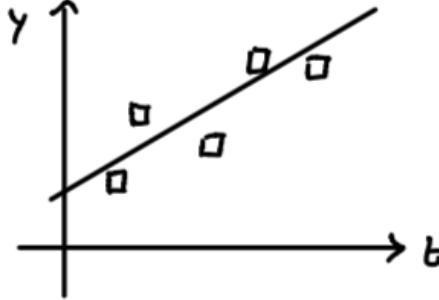
Setze

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Nun gilt es  $\|Ax - b\|_2$  zu minimieren

$$Ax = b \Leftrightarrow \begin{bmatrix} \varphi_1(t_1) & \cdots & \varphi_n(t_1) \\ \varphi_1(t_2) & \cdots & \varphi_n(t_2) \\ \vdots & & \vdots \\ \varphi_1(t_m) & \cdots & \varphi_n(t_m) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

**Beispiel 23.3** (Ausgleichsproblem)



Wähle  $n = 2$  mit  $\varphi_1(t) = 1$  und  $\varphi_2(t) = t$ . Dann ergibt sich

$$f(t) = x_2 t + x_1.$$

Bezeichne nun  $x_2$  mit  $m$  und  $x_1$  mit  $c$ , um die übliche Geradedarstellung zu erhalten:

$$f(t) = mt + c,$$

wobei  $c$  der y-Achsenabschnitt und  $m$  die Steigung ist. Dann ergibt sich:

$$Ax = b \Leftrightarrow$$

$$\begin{bmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_m & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

**Satz 23.4** (von Gauß)

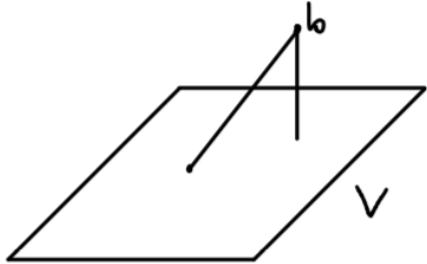
Sei  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $m \geq n$ . Äquivalent sind:

- i)  $\|Ax - b\|_2 = \min_{v \in \mathbb{R}^n} \|Av - b\|_2$
- ii)  $A^T Ax = A^T b$  "Normalengleichung"

**Bemerkung**

$V = \{Ax : x \in \mathbb{R}^n\} \leq \mathbb{R}^n$  Unterraum

Das gesuchte  $Ax$  ist die orthogonale Projektion von  $b$  in  $V$ .



*Beweis.*

i)  $\Leftrightarrow$

Für beliebiges  $y \in \mathbb{R}^n$  gilt:

$$\begin{aligned}
 \|Ax - b\|_2^2 &\leq \|A(x + y) - b\|_2^2 \\
 &= (A(x + y) - b)^T(A(x + y) - b) \\
 &= (A(x - b) + Ay)^T((Ax - b) + Ay) \\
 &= \|Ax - b\|_2^2 + 2\underbrace{(Ay)^T(Ax - b)}_{=0} + \underbrace{\|Ay\|_2^2}_{\geq 0}
 \end{aligned}$$

$$\Leftrightarrow (Ay)^T(Ax - b) = 0 \text{ für alle } y \in \mathbb{R}^n$$

$$\Leftrightarrow Ax - b \text{ ist orthogonal auf } V$$

$$\Leftrightarrow y^T A^T Ax - y^T A^T b = 0 \text{ für alle } y \in \mathbb{R}^n$$

$$\Leftrightarrow A^T Ax - A^T b = 0$$

□

### Eigenschaften 23.5

- $A^T A \in \mathbb{R}^{n \times n}$  symmetrisch positiv semidefinit  $[\forall x \in \mathbb{R}^n : \|Ax\|_2^2 = x^T A^T Ax \geq 0]$
- $A^T A$  ist positiv definit  $\Leftrightarrow \text{Rang}(A) = n$   $[x^T A^T Ax = \|Ax\|_2^2 = 0 \Leftrightarrow Ax = 0 \text{ falls } \text{Rang}(A) = n \Rightarrow x = 0]$

### Algorithmus 23.6

#### 1. Algorithmus

Berechne  $A^T A$  ( $\frac{1}{2}mn^2$  Operationen)  
und  $A^T b$  ( $mn$  Operationen)

Löse  $A^T Ax = A^T b$  mit der Choleskyzerlegung ( $\frac{1}{6}n^3$  Operationen)

## 2. Algorithmus

Berechne QR-Zerlegung von  $A$  ( $mn^2$  Operationen)

Nun gilt  $A = QR$  und damit lässt sich  $\|Ax - b\|_2^2 = \|QRx - b\|_2^2 = \|Rx - Q^T b\|_2^2 = \|\tilde{R}x - c\|_2^2 + \|d\|_2^2$  mit  $R = \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix}$  und  $Q^T b = \begin{bmatrix} c \\ d \end{bmatrix}$

umschreiben.

Dabei ist  $\tilde{R} = (r_{ij})_{i,j=1}^n$  eine quadratische  $n \times n$  rechte obere Dreiecksmatrix und  $c$  der Vektor aus den ersten  $n$  Einträgen von  $Q^T b$ .

$\|Ax - b\|_2 = \min! \Leftrightarrow \tilde{R}x = c$ , dann ist  $\|Ax - b\|_2^2 = \|d\|_2^2$

Berechne  $Q^T b = Q_n Q_{n-1} \dots Q_1 b$  ( $2nm$  Operationen)

Löse  $\tilde{R}x = c$  ( $\frac{1}{2}n^2$  Operationen)

Der 2. Algorithmus mit der QR-Zerlegung ist etwa doppelt so teuer wie der 1. Algorithmus dafür aber deutlich stabiler.

### Beispiel 23.7

Sei  $A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix}$ ,  $b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$  und  $\varepsilon^2 < \text{eps}$ .

Es gilt  $A^T A = \begin{bmatrix} 1 + \varepsilon^2 & 1 \\ 1 & 1 + \varepsilon^2 \end{bmatrix}$  und  $A^T b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ .

Die exakte Lösung ist  $x_1 = x_2 = \frac{1}{2+\varepsilon^2} \approx \frac{1}{2}$

In Gleitkommaarithmetik ist  $A^T A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$  singulär

Die QR-Zerlegung mit Householder (in Gleitkommaarithmetik) liefert das

exakte Ergebnis:  $\alpha_1 = 1$ ,  $v_1 = \begin{bmatrix} 2 \\ \varepsilon \\ 0 \end{bmatrix}$ ,  $R = \begin{bmatrix} -1 & -1 \\ 0 & \sqrt{2}\varepsilon \\ 0 & 0 \end{bmatrix}$ ,  $Q^T b = \begin{bmatrix} 1 \\ \frac{\sqrt{2}}{2}\varepsilon \\ \frac{\sqrt{2}}{2}\varepsilon \end{bmatrix}$

**Alg. 1:** Lösung von  $A^T Ax = A^T b$   $\text{cond}_2(A^T A) = \text{cond}_2(A)^2 \geq \text{cond}_2(A) = \frac{\max_y \|Ay\|}{\min_z \|Az\|}$  Der abschließende Bruch funktioniert auch für nicht inv. Matrizen.

**Alg. 2:** Lösung von  $\tilde{R}x = c$ ,  $\text{cond}_2(\tilde{R}) = \text{cond}_2(R) = \text{cond}_2(A)$

## IV Nichtlineare Gleichungssysteme

Problemstellung:

Zu einer Funktion  $f: U \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $U$  offen, suche  $x \in U$  mit  $f(x) = 0$ ,

$$\text{d.h. } \begin{cases} f_1(x_1, \dots, x_n) = 0 \\ f_2(x_1, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, \dots, x_n) = 0 \end{cases}.$$

Eventuell ex. keine Lösungen

$$f(x) = e^x$$

oder es ex. mehrere Lösungen

$$f(x) = x^2 - 1 \quad \text{oder} \quad f(x) = \tan(x) - x$$

Erinnerung/Wiederholung:

**Definition:**

Sei  $\Omega \subset \mathbb{R}^n$  abgeschlossen. Eine Abbildung  $\Phi: \Omega \rightarrow \mathbb{R}^n$  heißt kontrahierend, falls

$$\|\Phi(x) - \Phi(y)\| \leq \Theta \|x - y\| \quad \text{für } \Theta \in (0, 1) \text{ und alle } x, y \in \Omega.$$

Eine Abbildung heißt Selbstabbildung, falls  $\Phi(x) \in \Omega$  für alle  $x \in \Omega$ .

**Satz: (Spezialfall von BFS)**

Sei  $\Omega \subset \mathbb{R}^n$  abgeschlossen,  $\Phi: \Omega \rightarrow \Omega$  eine kontrahierende Selbstabbildung. Dann gilt:

- i) Es existiert ein Fixpunkt  $X^*$  von  $\Phi$ , d.h.  $\Phi(X^*) = X^*$
- ii) Für alle  $x^{(0)} \in \Omega$  konvergiert die Folge  $(x^{(k)})_{k \in \mathbb{N}}$  definiert durch  $x^{(k+1)} = \Phi(x^{(k)})$  gegen  $x^*$  mit

$$\|x^* - x^{(k)}\| \leq \frac{L}{1-L} \|x^{(k)} - x^{(k-1)}\| \quad \text{und} \quad (\text{a posteriori Schranke})$$

$$\|x^* - x^{(k)}\| \leq \frac{L^k}{1-L} \|x^{(1)} - x^{(0)}\| \quad \text{für ein } L < 1 \quad (\text{a priori Schranke})$$

*Beweis.* Ana II

□

**Bemerkung:**

$$f(x) \stackrel{!}{=} 0$$

Kann man das nichtlineare Gleichungssystem  $f(x) = 0$  in eine äquivalente Fixpunktgleichung umwandeln?

$$x = \Phi(x)$$

Mit Hilfe der Fixpunktgleichung konstruiert man eine Folge  $(x_n)_n$  ausgehend von  $x_0$  durch  $x_{n+1} = \Phi(x_n)$ , die hoffentlich gegen  $x^*$  mit  $x^* = \Phi(x^*)$  konvergiert.

**Beispiel:**

$$f(x) = 2x - \tan(x) \stackrel{!}{=} 0$$

Fixpunktgleichungen:

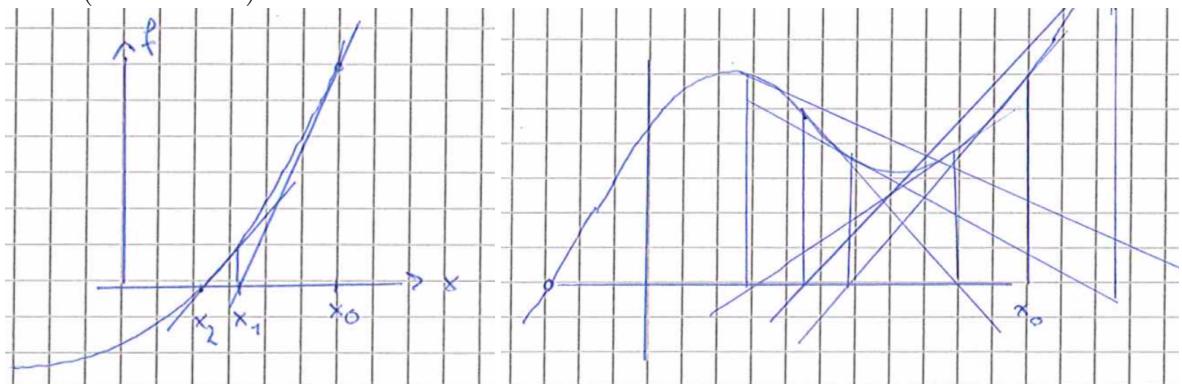
$$x = \frac{1}{2} \tan(x) \Rightarrow \Phi_1(x) = \frac{1}{2} \tan(x)$$

$$x = \arctan(2x) \Rightarrow \Phi_2(x) = \arctan(2x)$$

$$\begin{pmatrix} 2x - \tan(x) - x = -x \\ x = \tan(x) - x \end{pmatrix}$$

## 24 Newton-Verfahren

### 24.1 (Illustration)



Startwert  $x_0$ ,  $x_1$  Schnitt der Tangente von  $f$  im Punkt  $(x_0, f(x_0))$  mit der x-Achse,  $x_2$  Schnitt der Tangente in  $(x_1, f(x_1))$  mit x-Achse, usw.

### 24.2 (Herleitung Newton-Verfahren)

Sei  $x_0$  in der Nähe einer Nullstelle  $x^*$  von  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ . Taylorentwicklung liefert

$$0 = f(x^*) = f(x_0 + (x^* - x_0)) = f(x_0) + f'(x_0)(x^* - x_0) + \mathcal{O}(\|x^* - x_0\|^2)$$

Dabei ist  $f'(x_0)$  die Jacobimatrix an der Stelle  $x_0$ .  
 Näherungsweise gilt damit, falls  $f'(x_0)$  invertierbar ist

$$x^* - x_0 \approx -f'(x_0)^{-1}f(x_0)$$

Setze nun

$$x_1 = x_0 - f'(x_0)^{-1}f(x_0)$$

Gewöhnliches Newtonverfahren:

```

 $x_0$  ist gegeben
for  $k = 0, 1, \dots$  do
    Löse  $f'(x_k)\Delta x_k = -f(x_k)$  LGS (z.B. mit LR-Zerlegung)
     $x_{k+1} = x_k + \Delta x_k$ 
end for

```

### Satz 24.3

Sei  $f$  dreimal stetig differenzierbar,  $f(x^*) = 0$ ,  $f'$  invertierbar in einer Umgebung von  $x^*$  und die Folge  $(x_k)_k$  definiert durch das gewöhnliche Newtonverfahren. Dann gilt für den Fehler  $e_k = x_k - x^*$

$$e_k = \frac{1}{2}f'(x_k)^{-1}f''(x_k)[e_k, e_k] + \mathcal{O}(\|e_k\|^3)$$

Insbesondere gilt  $\|e_{k+1}\| \leq C\|e_k\|^2$ , d.h. das Newtonverfahren konvergiert quadratische (Ordnung 2), falls  $\|e_k\|$  genügend klein ist.

Dabei ist

$$f''(x)[y, z] = \sum_{k=1}^n z_k \sum_{j=1}^n \frac{\delta^2}{\delta_{x_k} \delta_{x_j}} f(x) y_j \in \mathbb{R}^n$$

*Beweis.*

$$\begin{aligned}
0 &= f(x^*) = f(x_k - e_k) \\
&\stackrel{\text{Taylor}}{=} f(x_k) - f'(x)_k e_k + \frac{1}{2} f''(x_k)[e_k, e_k] + \mathcal{O}(\|e_k\|^3) \\
&= -f''(x_k)(x_{k+1} + x^* - x^* - x_k) - f'(x_k)e_k + \frac{1}{2} f''(x_k)[e_k, e_k] + \mathcal{O}(\|e_k\|^3) \\
&= -f'(x_k)(e_{k+1}) + \frac{1}{2} f''(x_k)[e_k, e_k] + \mathcal{O}(\|e_k\|^3) \\
\Rightarrow e_{k+1} &= \frac{1}{2} f'(x_k)^{-1} f''(x_k)[e_k, e_k] + \mathcal{O}(\|e_k\|^3) \\
\Rightarrow \|e_{k+1}\| &\leq \frac{1}{2} C \|e_k\|^2
\end{aligned}$$

□

#### Definition 24.4

Eine Folge  $(x_k)_k$  konvergiert mit Ordnung  $p$  für  $p \geq 1$  gegen  $x^*$  falls ein  $C \geq 0$  existiert mit

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^p,$$

wobei  $C < 1$ , falls  $p = 1$ .

#### Satz 24.5 (Newton Mysovskii)

$f \in \mathcal{C}^1(D, \mathbb{R}^n)$ ,  $D \subset \mathbb{R}^n$  offene Teilmenge,  $f'(x)$  invertierbar für bel.  $x \in D$  und es gelte:

- i)  $\|\Delta x_0\| \leq \alpha$  (Definition von  $\alpha$ )
- ii)  $\|f'(x)^{-1}(f'(y) - f'(z))(y - z)\| \leq \omega \|y - z\|^2$  (Definiton von  $\omega$ ) für bel.  $x, z \in D$  und festes  $y \in \overline{xz}$
- iii)  $y := \frac{1}{2}\alpha\omega < 1$
- iv) Für  $\rho := \frac{\alpha}{1-y}$  ist  $B_\rho(x_0) \subset D$

Dann gilt für die Folge der Iteration des Newtonverfahrens  $(x_k)_k$ :

- $(x_k)_k \subset B_\rho(x_0)$
- $x_k \rightarrow x^*$  für  $k \rightarrow \infty$  mit  $f(x^*) = 0$ , genauer:
- $\|x_{k+1} - x_k\| \leq \frac{\omega}{2} \|x_k - x_{k-1}\|^2$

Hierbei ist  $\overline{xz}$  die Strecke zwischen  $x$  und  $z$  und  $B_\rho(x_0) = \{x \in \mathbb{R}^n, \|x-x_0\| < \rho\}$  die offene Kugel um  $x_0$  mit Radius  $\rho$ .

*Beweis.* mühsam □

## 24.6 (Praktische Durchführung)

$x_0$  sei gegeben.

**for**  $k = 0, 1, 2, \dots$  **do**

Löse  $f'(x_k)\Delta x_k = -f(x_k)$  mit LR-Zerlegung

$x_{k+1} = x_k + \Delta x_k$

**if**  $\|\Delta_k\| \leq \text{TOL}$  or  $k \geq k_{\max}$  **then**

$x_{k+1}$  ist die Lösung

Warnung, falls  $k \geq k_{\max}$

**end if**

**end for**

$\|f(x_k)\| \leq \text{TOL}$  ist **kein** geeignetes Abbruchkriterium. Ersetzt man die nichtlineare Gleichung  $f(x) = 0$  durch  $\tilde{f}(x) = Af(x) = 0$  für  $A$  invertierbare Matrix, so ändern sich die Iterierten nicht.

$f \mapsto Af \Rightarrow f'(x)^{-1}f(x) \mapsto f'(x)^{-1}A^{-1}Af(x)$

Man sagt, das Newtonverfahren ist affin invariant. Deswegen sollte sich auch das Abbruchkriterium nicht ändern.

$\|f(x)\| \mapsto \|af(x)\|$  statt  $\|x_k\| \leq \text{TOL}$  oft  $\|x_k\| \leq \frac{\text{TOL}}{1 - \frac{\|\Delta_k\|}{\|\Delta_k^{-1}\|}}$

## 24.7 (Vereinfachtes Newtonverfahren)

### Algorithmus:

$A \approx f'(x_0)$  ( $LR = A$ )

**for**  $k = 0, 1, \dots$  **do**

Löse  $A\Delta x_k = -f(x_k)$  (mit LR von A)

$x_{k+1} = x_k + \Delta x_k$

**end for**

### Satz 1:

Sei  $f \in \mathcal{C}^2(D, \mathbb{R}^n)$ ,  $f(x^*) = 0$  und  $A$  invertierbar. Dann gilt für  $e_k = x_k - x^*$

$$e_{k+1} = (I - A^{-1}f'(x_k))e_k + \mathcal{O}(\|e_k\|^2)$$

*Beweis.*

$$\begin{aligned}
0 &= f(x^*) = f(x_k - e_k) \stackrel{\text{Taylor}}{=} f(x_0) - f'(x_k)e_k + \mathcal{O}(\|e_k\|^2) \\
f(x_k) &= -A(x_{k+1} - x^* + x^* - x_k) = -A(e_{k+1} - e_k) \\
0 &= -A(e_{k+1} - e_k) - f'(x_k)e_k + \mathcal{O}(\|e_k\|^2) \\
\Rightarrow Ae_{k+1} &= (A - f'(x_k))e_k + \mathcal{O}(\|e_k\|^2)
\end{aligned}$$

□

**Satz 2:**

Sei  $f \in \mathcal{C}^1(D, \mathbb{R}^n)$ ,  $A$  invertierbar,  $x_0 \in D$  mit

- i)  $\|\Delta x_0\| \leq \alpha$
- ii)  $\|I - A^{-1}f'(x)\| \leq y < 1$  für bel.  $x \in D$
- iii)  $B_\rho(x_0) \subset D$  mit  $\rho = \frac{\alpha}{1-y}$

Dann konvergiert  $x_k$  aus dem Algorithmus gegen  $x^*$  mit  $f(x^*) = 0$   $\|x_{k+1} - x_k\| \leq y\|x_k - x_{k-1}\|$ , d.h. das vereinfachte Newtonverfahren konvergiert lokal linear.

*Beweis.* Das vereinfachte Newtonverfahren ist Fixpunktiteration zu  $\Phi(x) = x - A^{-1}f(x)$

- $\Phi$  ist kontrahierend wegen ii)
- $\Phi$  ist eine Selbstabbildung

$\Rightarrow$  BFS liefert die Behauptung. □

## V Gewöhnliche Differentialgleichungen

Problem:

Suche Lösung  $y: [t_0, T] \rightarrow \mathbb{R}^d$  der Anfangswertaufgabe

$$\frac{d}{dt}y(t) = y'(t) = f(t, y(t)) \quad \text{für } t \in (t_0, T)$$

d.h. eine Funktion, die die obige Gleichung erfüllt.

$f: \mathcal{U} \rightarrow \mathbb{R}^d$ ,  $\mathcal{U} \subset \mathbb{R} \times \mathbb{R}^d$  offen,  $y(t_0) = y_0$  Anfangswert,  $(t_0, y_0) \in \mathcal{U}$ .

## 25 Beispiele für gewöhnliche Differentialgleichungen

**Beispiel 25.1** (Harmonischer Oszillator)

$$\begin{aligned}\frac{d}{dt}p(t) &= -q(t) \\ \frac{d}{dt}q(t) &= p(t)\end{aligned}$$

Dabei ist  $\frac{d}{dt}p(t)$  die Änderungsrate zur Zeit  $t$  von  $p$ ,  
 $q(t)$  die Position zur Zeit  $t$  und  
 $p(t)$  die Geschwindigkeit zur Zeit  $t$ .

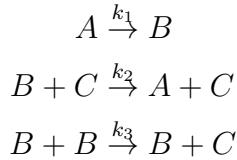
**Beispiel 25.2** (Pendel)

$$\begin{aligned}ms''(t) &= -mg \sin(\phi(t)), \quad s(t) = l\phi(t) \\ \phi''(t) &= -\frac{g}{l} \sin(\phi(t)) \\ y(t) &:= \begin{pmatrix} \phi(t) \\ \phi'(t) \end{pmatrix}, \quad y'(t) = \underbrace{\begin{pmatrix} \phi'(t) \\ -\frac{g}{l} \sin(\phi(t)) \end{pmatrix}}_{=f(t,y(t))}\end{aligned}$$

Dabei ist  $\phi(t)$  der Winkel zur Zeit  $t$ ,  $s(t)$  die Position zur Zeit  $t$ ,  $l$  die Länge des Pendels,  $m$  die Masse und  $g$  die Erdbeschleunigung.

**Beispiel 25.3** (Chemische Reaktionen)

Hier möchte man den Verlauf chemischer Reaktionen simulieren. Weiß man etwa, dass die Substanzen  $A, B, C$  gemäß



mit Reaktionskonstanten  $k_1, k_2, k_3$  reagieren, dann liefert das Massenwirkungsgesetz für die Konzentrationen  $a(t), b(t), c(t)$  der Substanzen  $A, B, C$  zur Zeit  $t$ .

$$\begin{aligned}a' &= -k_1a + k_2bc \\ b' &= k_1a - k_2bc - k_3b^2 \\ c' &= k_3b^2\end{aligned}$$

Zusätzlich müssen Anfangskonzentrationen  $a(0)$ ,  $b(0)$  und  $c(0)$  gegeben sein.

**Beispiel 25.4** (Räuber Beute Modell)

Die Anzahl  $y(t)$  von Speisefischen zur Zeit  $t$  und die Anzahl  $z(t)$  von Raubfischen mit Hilfe des Populationsmodells

$$\begin{aligned} y' &= ay - byz \\ z' &= -cz \end{aligned}$$

berechnet werden.

Hierbei ist  $a$  die Geburtenrate der Speisefische,  $b$  die Effizienz der Raubfische,  $c$  die Sterberate der Raubfische und  $d$  die nahrungsabhängige Geburtenrate der Raubfische.

## 26 Erinnerung an die Theorie gewöhnlicher DGLs

**Bemerkung 26.1**

Jeder Differentialgleichung  $k$ -ter Ordnung

$$y^{(k)} = f(t, y, y', \dots, y^{(k-1)})$$

kann in ein System erster Ordnung umgeschrieben werden:

Mit der Setzung

$$\begin{aligned} y_1 &= y & y'_1 &= y_2 \\ y_2 &= y' & y'_2 &= y_3 \\ &\vdots &&\vdots \\ y_{k-1} &= y^{(k-2)} & y'_{k-1} &= y_k \\ y_k &= y^{(k-1)} & y'_k &= f(t, y_1, \dots, y_k) \end{aligned}$$

erhält man für  $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$  das System  $Y' = F(t, Y)$ , wenn man  $F$  durch

$$F(t, Y) = \begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ y_k \\ f(t, y_1) \end{bmatrix} \text{ setzt.}$$

### Definition 26.2

Hängt die rechte Seite  $f$  nicht explizit von  $t$  ab, so heißt die Differentialgleichung autonom.

### Bemerkung

Jede nichtautonome Differentialgleichung

$$y' = f(t, y) \quad y(t_0) = y_0$$

ist äquivalent zu einem autonomen System

$$Y' = F(Y) \text{ mit } Y = \begin{bmatrix} y \\ t \end{bmatrix} \text{ und } F(Y) = \begin{bmatrix} f(t, y) \\ 1 \end{bmatrix}$$

### Bemerkung (Allgemeine Voraussetzungen)

Als nächstes wiederholen wir einige theoretische Resultate zur Existenz, Eindeutigkeit und Stabilität von Lösungen von Anfangswertproblemen. Dazu sei ab jetzt

- $U \subset \mathbb{R} \times \mathbb{R}^d$  offen und zusammenhängend
- $f : U \rightarrow \mathbb{R}^d$  stetig und erfülle folgende lokale Lipschitz-Bedingung:  
 $\exists L \forall K \subset U \text{ kompakt } \forall (t, y), (t, z) \in K : \|f(t, y) - f(t, z)\| \leq L \|y - z\|$

Die lokale Lipschitz-Bedingung ist erfüllt, falls  $f$  stetig differenzierbar ist. Dann kann  $L = \max_{(t, y) \in K} \|f_y(t, y)\|$  gewählt werden.

**Satz 26.3** (Satz von Picard-Lindelöf zur lokalen Existenz und Eindeutigkeit)) Unter obigen Voraussetzungen gilt: Es gibt ein offenes Intervall  $I$  mit  $t_0 \in I$ , sodass genau eine Lösung  $y : I \rightarrow \mathbb{R}^d$  existiert, mit

$$y'(t) = f(t, y(t)), \text{ für } t \in I \text{ und } y(t_0) = y_0$$

Diese Lösung kann bis an den Rand von  $U$  fortgesetzt werden.

*Beweis.* Ana II □

### Beispiel 26.4

Die rechte Seite der Differentialgleichung

$$y' = y^2, \quad y(0) = 1, \quad U = \mathbb{R} \times \mathbb{R}$$

ist lokal Lipschitz-stetig, erfüllt also die Voraussetzungen von Picard-Lindelöf. Die eindeutige Lösung  $y(t) = (1-t)^{-1}$  existiert auf dem offenen Intervall  $I = (-\infty, 1)$  und  $t_0 = 0 \in I$ . Es ist jedoch  $\lim_{t \nearrow 1} y(t) = +\infty$ .

Für numerische Verfahren ist es wichtig wie sich Störungen der Anfangswerte auf die Lösung auswirken.

### Satz 26.5

Zusätzlich zu den allgemeinen Voraussetzungen erfülle für  $\langle \cdot, \cdot \rangle$ , ein Skalarprodukt auf  $\mathbb{C}^d$ , und  $\|\cdot\|$ , die induzierte Norm, die rechte Seite  $f: [t_0, T] \rightarrow \mathbb{C}^d$   $f$  für ein  $l$  folgende einseitige Lipschitz-Bedingung

$$\Re \langle f(t, y) - f(t, z), y - z \rangle \leq l \|y - z\|^2 \quad \text{für alle } y, z \in U$$

Sind  $y$  und  $z$  zwei Lösungen von  $y' = f(t, y)$  zu verschiedenen Anfangswerten  $y_0$  bzw.  $z_0$ , so gilt

$$\|y(t) - z(t)\| \leq e^{l(t-t_0)} \|y_0 - z_0\| \quad \text{für alle } t \in [t_0, T].$$

*Beweis.*

$$\begin{aligned} \frac{d}{dt} \|y(t) - z(t)\|^2 &= 2 \Re \langle y'(t) - z'(t), y(t) - z(t) \rangle \\ &= 2 \Re \langle f(t, y(t)) - f(t, z(t)), y(t) - z(t) \rangle \\ &\leq 2l \|y(t) - z(t)\|^2 \end{aligned}$$

Falls  $y(t_0) \neq z(t_0)$  so gilt wegen der Eindeutigkeit der Lösung auch  $y(t) \neq z(t)$  für alle  $t$ .

Mit

$$\varphi(t) := \|y(t) - z(t)\|^2 \neq 0$$

erhalten wir

$$\frac{\varphi'(t)}{\varphi(t)} = \frac{d}{dt} \log \varphi(t) \leq 2l.$$

Integration ergibt

$$\begin{aligned} \log(\varphi(t)) - \log(\varphi(t_0)) &= \int_{t_0}^t \frac{d}{ds} \log(\varphi(s)) ds \leq \int_{t_0}^t 2l ds = 2l(t - t_0) \\ \Rightarrow \log \left( \frac{\varphi(t)}{\varphi(t_0)} \right) &\leq 2l(t - t_0) \end{aligned}$$

”Exponieren“ liefert  $\varphi(t) \leq e^{2l(t-t_0)} \varphi(t_0)$

□

### Bemerkung 26.6

- i) Fehler in den Anfangsdaten können maximal mit dem Faktor  $e^{l(t-t_0)}$  verstärkt werden.
- ii) Da  $f$  lokal Lipschitz-stetig ist, ist die Voraussetzung des Satzes mit  $l = L$  erfüllt. Für das bestmögliche (kleinste)  $l$  kann aber  $l \ll L$  gelten.
- iii)  $l < 0$  ist möglich, hingegen ist immer  $L > 0$

### Beispiel 26.7 (Testgleichung)

$$y' = \lambda y, \quad \lambda \in \mathbb{C}, \quad y(t_0) = y_0$$

Lösung

$$y(t) = e^{(t-t_0)\lambda} y_0$$

$$l = \Re \lambda, \quad L = |\lambda|$$

- $\Re \lambda < 0$ : Fehler werden gedämpft 0, ist asymptotisch stabil
- $\Re \lambda = 0$ : keine Fehlerverstärkung
- $\Re \lambda > 0$ : Fehler wachsen exponentiell

## 27 Euler-Verfahren

Einfachstes und ältestes Verfahren zur näherungsweisen Lösung von

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0$$

Idee: Ersetze lokal die (unbekannte) Lösung durch die bekannte Tangente an der Stelle  $t_0$ , so erhält man  $y_1 = y_0 + hf(t_0, y_0)$ , usw.

Allgemeine Iterationsvorschrit (explizites Eulerverfahren):

$$t_n = t_0 + nh$$

$$y_i = y_{i-1} + hf(t_{i-1}, y_{i-1}) \quad \text{für } i \in \mathbb{N}.$$

Ersetzt man lokal die (unbekannte) Lösung durch die Tangente an der ebenfalls unbekannten Stelle  $(t_1, y_1)$ , so erhält man  $y_1 = y_0 + hf(t_1, y_1)$ , usw.

Allgemein Iterationsvorschrit (implizites Eulerverfahren):

$$\begin{aligned} t_n &= t_0 + nh \\ y_i &= y_{i-1} + hf(t_i, y_i) \text{ für } i \in \mathbb{N}. \end{aligned}$$

Hierbei muss in jedem Schritt ein nicht-lineares Gleichungssystem gelöst werden (etwa mit Newton-Verfahren oder Fixpunktiteration).

Approximationsfehler beim expliziten Eulerverfahren:

Sei  $I = [t_0, T]$  ein Intervall und  $f: I \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  stetig differenzierbar und (global) Lipschitz-stetig, d.h.:

$$\forall y, z \in \mathbb{R}^d \forall t \in I: \|f(t, y) - f(t, z)\| \leq L\|y - z\|$$

Ist  $y: I \rightarrow \mathbb{R}^d$  Lösung des Anfangswertproblems  $y' = f(t, y), y(t_0) = y_0$ , dann ist  $y$  jetzt zweimal stetig differenzierbar, denn

$$y'' = \delta_t f + D_y f \cdot y' = \delta_t f + D_y f \cdot f$$

$D_y f = D_y f(t, y)$  bezeichnet die Ableitung ( $d \times d$  Matrix) nach  $y$ . Die Folge  $(y_n)_{n \in \mathbb{N}}$  ist mit  $t_n = t_0 + nh \in I$  durch das explizite Eulerverfahren definiert.

**Satz 27.1** (Fehlerabschätzung für das explizite Eulerverfahren)

Mit den eben gemachten Voraussetzungen gilt für den Fehler des expliziten Euler-Verfahrens

$$\|y_n - y(t_n)\| \leq M \cdot h,$$

mit

$$M = \frac{e^{L(T-t_0)} - 1}{L} \frac{1}{2} \max_{t \in I} \|y''(t)\|$$

Insbesondere gilt

$$\lim_{h \rightarrow 0} \max_{n \in \mathbb{N}} \|y_n - y(t_n)\| = 0,$$

d.h. die Näherungslösung konvergiert gleichmäßig gegen die exakte Lösung der Anfangswertaufgabe, falls  $h$  gegen Null geht.

*Beweis.* Beweis erfolgt in 3 Schritten:

- 1) Abschätzung für den lokalen Fehler  
Fehler nach einem Schritt des Verfahrens mit Startwert auf der exakten

Lösung:

$$\begin{aligned}
 & \underbrace{y(t_{n+1})}_{\substack{\text{exakte Lsg.} \\ \text{bei } t_{n+1}}} - \underbrace{(y(t_n) + hf(t_n, y(t_n)))}_{\substack{\text{expl. Eulerverfahren mit} \\ \text{Startwert } y(t_n)}} = y(t_{n+1}) - y(t_n) - hy'(t_n) \\
 & \qquad \qquad \qquad \text{Taylorentwicklung} \\
 & \qquad \qquad \qquad \text{von } y(t_{n+1}) \text{ im } t_n \text{ } h^2 \int_0^1 (1-\theta)y''(t_n + \theta h) d\theta \\
 \Rightarrow & \|y(t_{n+1}) - (y(t_n) + hf(t_n, y(t_n)))\| \leq Ch^2 \\
 \text{für } C = & \frac{1}{2} \max_{t \in I} \|y''(t)\|.
 \end{aligned}$$

2) Fehlerfortpflanzung

Ausgehend von Anfangswerten  $v_n$  bzw.  $w_n$  ergeben sich durch einen Eulerschritt die Näherungen

$$\begin{aligned}
 v_{n+1} &= v_n + hf(t_n, v_n) \\
 w_{n+1} &= w_n + hf(t_n, w_n)
 \end{aligned}$$

Bildet man die Norm der Differenz, so gilt:

$$\begin{aligned}
 \|v_{n+1} - w_{n+1}\| &\leq \|v_n - w_n\| + h\|f(t_n, v_n) - f(t_n, w_n)\| \\
 &\leq (1 + Lh)\|v_n - w_n\|
 \end{aligned}$$

3) Fehlerakkumulation

Bezeichne im Folgenden mit  $y_n^k$  die Näherung an  $y(t_n)$  zum Anfangswert  $y(t_k)$  nach  $(n-k)$  Schritten.

Dann ist  $y_k = y_k^0$  und  $y(t_k) = y_k^k$

Nach Schritt 1:  $\|y_{k+1}^k - y_{k+1}^{k+1}\| < Ch^2$

Nach Schritt 2:  $\|y_m^k - y_m^{k+1}\| \leq (1 + hL)\|y_{m-1}^k - y_{m-1}^{k+1}\|$

induktiv erhält man also:

$$\begin{aligned}
 \|y_n^k - y_n^{k+1}\| &\leq (1 + hL)\|y_{n-1}^k - y_{n-1}^{k+1}\| \\
 &\leq \dots \\
 &\leq (1 + hL)^{n-k-1}\|y_{k+1}^k - y_{k+1}^{k+1}\| \\
 &\leq (1 + hL)^{n-k-1}Ch^2
 \end{aligned}$$

Damit ist

$$\begin{aligned}
\|y_n - y(t_n)\| &= \|y_n^0 - y_n^n\| \\
&\leq \|y_n^0 - y_n^1\| + \|y_n^1 - y_n^2\| + \dots + \|y_n^{n-1} - y_n^n\| \\
&= \sum_{l=1}^n \|y_n^{l-1} - y_n^l\| \\
&\leq Ch^2 \sum_{l=1}^n (1 + hL)^{n-l} \\
&= Ch^2 \frac{(1 + hL)^n - 1}{1 + hL - 1} = Ch \frac{(1 + hL)^n - 1}{L} \\
&\leq Ch \frac{e^{nhL} - 1}{L} \leq Ch \frac{e^{(T-t_0)L} - 1}{L}
\end{aligned}$$

Die vorletzte Ungleichung folgt aus  $(1 + x) \leq e^x$ , die letzte aus  $nh \leq T - t_0$ .

□

## 28 Runge-Kutta Verfahren

Ziel: Verfahren höherer Ordnung.

Das (explizite) Eulerverfahren hatte nur Ordnung 1. D.h. der Fehler ist aus  $\mathcal{O}(h)$  für  $h \rightarrow 0$ .

Die exakte Lösung  $y$  erfüllt für  $t_1 = t_0 + h$

$$y(t_1) = y_0 + \int_{t_0}^{t_1} y'(t) dt = y_0 + \int_{t_0}^{t_1} f(t, y(t)) dt$$

Anwendung einer s-stufigen Quadraturformel mit Knoten  $c_1, \dots, c_s$  und Gewichten  $b_1, \dots, b_s$  ergibt

$$y(t_1) \approx y_0 + h \sum_{i=1}^s b_i f(t_0 + c_i h, y(t_0 + c_i h))$$

**Beispiel 28.1** (Mittelpunktsregel)

Sei  $s = 1$ ,  $b_1 = 1$ ,  $c_1 = 1/2$

$$y(t_1) \approx y(t_0) + h f(t_0 + \frac{1}{2}h, y(t_0 + \frac{1}{2}h))$$

Wie berechnet man  $y(t_0 + \frac{1}{2}h)$ ? Zum Beispiel mit einem Schritt des expliziten Eulerverfahrens

$$y(t_0 + \frac{1}{2}h) \approx y(t_0) + \frac{1}{2}h f(t_0, y(t_0))$$

### Bemerkung

In der Quadraturformel treten also die Werte der unbekannten Lösung an den Stellen  $t_0 + c_i h$  auf.

Zur Approximation von  $y(t_0 + c_i h)$  verwenden wir daher erneut die Integraldarstellung der Lösung

$$y(t_0 + c_i h) = y_0 + \int_{t_0}^{t_0 + c_i h} f(t, y(t)) dt$$

und approximieren die Integrale  $\int_{t_0}^{t_0 + c_i h} f(t, y(t)) dt$  mit Quadraturformeln mit **denselben** Knoten  $t_0 + c_i h$  und Gewichten  $a_{ij}$  passend zu  $\int_0^{c_i}$

$$y(t_0 + c_i h) \approx y_0 + h \sum_{j=1}^s a_{ij} f(t_0 + c_j h, y(t_0 + c_j h))$$

Zusammenfassend:

### Definition 28.2

Ein Schritt eines Runge-Kutta Verfahrens zur Lösung von  $y' = f(t, y)$ ,  $y(t_0) = y_0$  ist durch

$$\begin{aligned} y_1 &= y_0 + h \sum_{i=1}^s b_i Y'_i \\ Y'_i &= f(t_0 + c_i h, Y_i), \quad i = 1, \dots, s \\ Y_i &= y_0 + h \sum_{j=1}^s a_{ij} Y'_j, \quad i = 1, \dots, s \end{aligned}$$

gegeben mit Koeffizienten  $a_{ij}$ ,  $b_i$  und  $c_i$  ( $i, j = 1, \dots, s$ ).

### Bemerkung

Üblicherweise stellt man ein Runge-Kutta Verfahren für  $i, j \in \{1, \dots, s\}$  in einem sogenannten Butcher Tableau dar:

$\vdots$			
$c_i$	$a_{ij}$		
$\vdots$			
$\dots$	$b_j$	$\dots$	

Die  $Y_i$ 's sind im Allgemeinen Lösungen eines nichtlinearen Gleichungssystems.

Falls  $a_{ij} = 0$  für  $j \geq i$ , können die  $Y_i$ 's nacheinander explizit berechnet werden:

```

for  $i = 1, \dots, s$  do
     $Y_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} Y'_j$ 
     $Y'_i = f(t_0 + c_i h, Y_i)$ 
end for
 $y_1 = y_0 + h \sum_{i=1}^s b_i Y'_i$ 
```

**Beispiel 28.3** (Explizites Eulerverfahren)

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

$Y_1 = y_0$ ,  $Y'_1 = f(t_0, Y_1)$  und  $y_1 = y_0 + hY'_1$ . Der lokale Fehler des expliziten Euler-Verfahren ist aus  $\mathcal{O}(h^2)$  (Fraglich, warum der Fehler plötzlich kleiner im Vergleich zur Einleitung des Kapitels geworden ist).

**Beispiel 28.4** (Runge-Verfahren 1895)

Mittelpunktsregel + explizites Eulerverfahren

$$y_1 = y_0 + h f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(t_0, y_0)\right)$$

oder

$$\begin{aligned} Y_1 &= y_0, & Y_2 &= y_0 + \frac{h}{2} Y'_1 \\ Y'_1 &= f(t_0, y_0), & Y'_2 &= f\left(t_0 + \frac{h}{2}, Y_2\right) \\ y_1 &= y_0 + h Y'_2 \end{aligned}$$

Butcher Tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

lokaler Fehler (Taylorentwicklung von  $y_1 = y_0 + h f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(t_0, y_0)\right)$  um  $(t_0, y_0)$ )

$$y_1 = y_0 + h f(t_0, y_0) + \frac{h^2}{2} (\partial_t f(t_0, y_0) + D_y f(t_0, y_0) f(t_0, y_0)) + \mathcal{O}(h^3)$$

(Taylorentwicklung von  $y(t_0 + h)$  um  $t_0$ )

$$\begin{aligned} y(t_0 + h) &= y(t_0) + hy'(t_0) + \frac{h^2}{2}y''(t_0) + \mathcal{O}(h^3) \\ &= y(t_0) + hf(t_0, y_0) + \frac{h^2}{2}(\partial_t f(t_0, y_0) + D_y f(t_0, y_0)f(t_0, y_0)) + \mathcal{O}(h^3) \end{aligned}$$

Bildet man die Differenz, so erhält man

$$\|y_1 - y(t_0 + h)\| \in \mathcal{O}(h^3)$$

**Beispiel 28.5** (Kutta-Verfahren 1901)

Simpsonregel mit doppeltem Knoten bei  $\frac{1}{2}$

$$\begin{array}{ll} y_1 = y_0 + h(1/6Y'_1 + 1/3Y'_2 + 1/3Y'_3 + 1/6Y'_4) \\ Y'_1 = f(t_0, Y_1) & Y_1 = y_0 \\ Y'_2 = f(t_0 + h/2, Y_2) & Y_2 = y_0 + h/2f(t_0, Y_1) \\ Y'_3 = f(t_0 + h/2, Y_3) & Y_3 = y_0 + h/2f(t_0, Y_2) \\ Y'_4 = f(t_0 + h, Y_4) & Y_4 = y_0 + hf(t_0, Y_3) \end{array}$$