ZUSAMMENFASSUNG NUMERIK

franclu1
ZHAW School of Engineering

Fehler

Der *absolute Fehler* ist definiert durch

$$|\Delta a| := |\tilde{a} - a|$$

Der *relative Fehler* in $\%$ ist nur für $a \neq 0$ definiert und gegeben durch

$$\delta_a := \frac{|\tilde{a} - a|}{|a|}$$

a = *Ergebnis;* $\tilde{a}$ = *Näherung*

Normen

Eine Norm ordnet jedem Vektor x eines (reellen oder komplexen) Vektorraums V eine reelle Zahl $\|x\|$ zu. Eine Norm erfüllt die folgenden Bedingungen:

1. **Positivität:**

$$\|x\| \geq 0 \quad \text{und} \quad \|x\| = 0 \text{ genau dann, wenn } x = 0$$

2. **Homogenität:**

$$\|\alpha \cdot x\| = |\alpha| \cdot \|x\| \quad \text{für jede (reelle bzw. komplexe) Zahl } \alpha.$$

3. **Dreiecksungleichung:**

$$\|x + y\| \leq \|x\| + \|y\| \quad \text{für alle } x, y \in V$$

Die p-Norm eines **reellen oder komplexen Vektors**

$$x = (x_1, x_2, \ldots, x_n) \in K^n \quad \text{mit } K = \mathbb{R} \text{ oder } K = \mathbb{C}$$

ist für reelles $1 \leq p < \infty$ durch

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

definiert, wobei $|x_i|$ den Betrag der Komponente x_i bezeichnet.

$$\begin{aligned} p = 1 &\Rightarrow \text{Summennorm,} \\ p = 2 &\Rightarrow \text{euklidische Norm,} \\ p = \infty &\Rightarrow \text{Maximumsnorm: } \|x\|_\infty = \max_{i=1,\ldots,n} |x_i| \end{aligned}$$

$X = \mathbb{R}^n$, $Y = \mathbb{R}^m$, $B \in \mathbb{R}^{m \times n}$ = $(m \times n)$**-Matrix.** Stattet man sowohl X als auch Y mit der p-Norm für $1 \leq p \leq \infty$ aus, so bezeichnet man die entsprechende *Operatornorm* kurz als

$$\|B\|_p := \|B\|_{X \to Y}$$

Es gilt:

$$\|B\|_\infty = \max_{i=1,\ldots,m} \sum_{k=1}^{n} |b_{i,k}| \qquad \text{(maximale absolute Zeilensumme)}$$

$$\|B\|_1 = \max_{k=1,\ldots,n} \sum_{i=1}^{m} |b_{i,k}| \qquad \text{(maximale absolute Spaltensumme)}$$

$$\|B\|_2 = \sqrt{\lambda_{\max}(B^T B)} \qquad \text{(Spektralnorm)}$$

Dabei ist B^T die Transponierte von B und $\lambda_{\max}$ der größte Eigenwert des Produkts $B^T B$

O-Notation

Taylorreihe:

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)\,(x - x_0) + \frac{1}{2} f''(x_0)\,(x - x_0)^2 + \frac{1}{3!} f^{(3)}(x_0)\,(x - x_0)^3 + \cdots \\ &= \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}\,(x - x_0)^k = T_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}\,(x - x_0)^{n+1}, \qquad (x_0 < \xi < x) \end{aligned}$$

Seien $f, g : \mathbb{R} \to \mathbb{R}$ Funktionen und $a \in \mathbb{R}$. Man sagt, f ist *großes O von g für* $x \to a$ und schreibt

$$f(x) = \mathcal{O}(g(x)) \quad \text{für } x \to a$$

$$f(x) = T_n(x) + \mathcal{O}(|h|^{n+1}) \qquad h \to 0$$

1. Die Funktion $f(x)$ lässt sich bis auf einen Fehler durch das Polynom T_n approximieren.
2. Für $|h| < 1$ ist der Betrag des Fehlers durch eine Konstante $C > 0$ multipliziert mit der $(n+1)$-ten Potenz von $|h|$ beschränkt, d. h.

$$|f(x) - T_n(x)| \leq C\,|h|^{n+1}$$

Die $\mathcal{O}$-Notation erfasst das *qualitative Verhalten* des absoluten Fehlers, wenn h gegen 0 geht

h = Schrittweite, dx

Numerisches Differenzieren

Grundlage numerischer Ableitung:

$$f(x_0 + h) = f(x_0) + f'(x_0)\,h + \frac{f''(x_0)}{2}\,h^2 + \frac{f^{(3)}(x_0)}{6}\,h^3 + O(h^4)$$

$$f(x_0 - h) = f(x_0) - f'(x_0)\,h + \frac{f''(x_0)}{2}\,h^2 - \frac{f^{(3)}(x_0)}{6}\,h^3 + O(h^4)$$

Vorwärtsdifferenzenquotient (FO=1):

$${}^1D_{f\rightarrow}(x_0, h) = \frac{f(x_0 + h) - f(x_0)}{h} \approx f'(x_0)$$

Rückwärtsdifferenzenquotient (FO=1):

$${}^1D_{f\leftarrow} = \frac{f(x_0) - f(x_0 - h)}{h} \approx f'(x_0)$$

absoluter Fehler des Differenzenquotienten (Diskretisierungsfehler):

$$\left| {}^1D_f(x_0, h) - f'(x_0) \right|$$

Fehlerordnung des Differenzenquotienten:

$$\left| {}^1D_f(x_0, h) - f'(x_0) \right| = \mathcal{O}(h^k)$$

symmetrischer Differenzenquotienten (FO=2):

$${}^1D_f(x_0, h) := \frac{f(x_0 + h) - f(x_0 - h)}{2h} \approx f'(x_0)$$

symmetrischer Differenzenquotienten für f" (FO=2):

$${}^2D_f(x_0, h) := \frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} = f''(x_0) + \mathcal{O}(h^2)$$

Fehlerordnung (FO)
FO 1: h halbieren = Fehler halbiert sich
FO 2: h halbieren = Fehler wird viermal kleiner

Konditionszahl

Konditionszahl = maximale Verstärkung des Fehlers
Relative Fehlerverstärkung:

$$\left| \frac{f(\tilde{x}) - f(x_0)}{f(x_0)} \right| \approx k_{\text{rel}}(x_0) \cdot \left| \frac{\tilde{x} - x_0}{x_0} \right|$$

rel. Fehler Ausgabe = Fehlerverstärkung mal rel. Fehler Eingabe

$$\text{mit } k_{\text{rel}}(x_0) = \left| f'(x_0) \cdot \frac{x_0}{f(x_0)} \right|$$

absolute Fehlerverstärkung:

$$|f(\tilde{x}) - f(x_0)| \approx k_{\text{abs}}(x_0) \cdot |\tilde{x} - x_0|$$

$$\text{mit } k_{\text{abs}}(x_0) = |f'(x_0)|$$

Zahlendarstellung | Fehler | Stabilität

Jede reelle Zahl x kann als Dezimalbruchentwicklung

$$x = d_0 + d_1 \cdot 10^{-1} + d_2 \cdot 10^{-2} + d_3 \cdot 10^{-3} + \ldots$$

mit $d_0 \in \mathbb{Z}$ und $d_1, d_2, d_3, \cdots \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ dargestellt werden.
x genau in Q, wenn Dezimalbruchentwicklung abbricht oder d_i periodisch werden.

Maschinenzahl: Menge der exakt dargestellten Zahlen auf dem Rechner.
Gleitpunktdarstellung:

$$x = f \cdot b^e = \pm \left(\sum_{j=1}^{m} d_j\, b^{-j} \right) \cdot b^e$$

wobei;

1. $b \in \mathbb{N} \setminus \{1\}$die Basis des Zahlensystems ist
2. die Mantisse f eine feste Anzahl m von Stellen besitzt

---

relative Maschinengenauigkeit:

$$\varepsilon := b^{1-m}$$

relative Rundungsfehler (Rundungseinheit):

$$u_M = \frac{1}{2} b^{1-m} = \frac{\varepsilon}{2}$$

Grundrechenarten der relativen Fehler:

1. *Addition und Subtraktion (problematisch = Auslöschung)*

$$\frac{\Delta(x_1 \pm x_2)}{x_1 \pm x_2} = \frac{x_1}{x_1 \pm x_2} \frac{\Delta x_1}{x_1} \pm \frac{x_2}{x_1 \pm x_2} \frac{\Delta x_2}{x_2}$$

2. *Multiplikation*

$$\frac{\Delta(x_1 x_2)}{x_1 x_2} \approx \frac{\Delta x_1}{x_1} + \frac{\Delta x_2}{x_2}$$

3. *Division*

$$\frac{\Delta(x_1 / x_2)}{x_1 / x_2} \approx \frac{\Delta x_1}{x_1} + \frac{\Delta x_2}{x_2}$$

Stabilität des Algorithmus:
Ein Algorithmus ist stabil, wenn seine Fehler nicht größer als der konditionsbedingte unvermeidbare Fehler sind.

Konditionierung für LGS

$$A\vec{x} = \vec{b} \iff CA\vec{x} = C\vec{b}$$
$$A\vec{x} - \vec{b} = 0$$

Relative Kondition:

$$\frac{\|\mathcal{L}(\tilde{x}) - \mathcal{L}(x)\|_Y}{\|\mathcal{L}(x)\|_Y} \leq \kappa(\mathcal{L}) \frac{\|\tilde{x} - x\|_X}{\|x\|_X},$$

wobei

$$\kappa(\mathcal{L}) = \|\mathcal{L}\|_{X\to Y} \|\mathcal{L}^{-1}\|_{Y\to X} \quad \text{Norm mal Norm}$$

Sei $\vec{x} + \Delta\vec{x}$ die Lösung von $A \cdot (\vec{x} + \Delta\vec{x}) = \vec{b} + \Delta\vec{b}$

$$\frac{\|\Delta\vec{x}\|}{\|\vec{x}\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \frac{\|\Delta\vec{b}\|}{\|\vec{b}\|}$$

Eigenschaften Konditionszahl:

1. Konditionszahlen von Matrizen hängen von der jeweiligen Norm $\|\cdot\|$ ab
2. Es gilt stets $1 \leq \kappa(A)$
3. $\kappa(A) = \kappa(A^{-1})$

Residuum

$$\vec{r} = \vec{b} - A \cdot \vec{x}$$

Rückwärtseinsetzen

Rückwärtseinsetzen: GLS-Variablen von unten nach oben auflösen.
Vorwärtseinsetzen: GLS-Variablen von oben nach unten auflösen.

$$x_n = \frac{b_n}{r_{n,n}}$$

$$x_{n-1} = \frac{b_{n-1} - r_{n-1,n}\, x_n}{r_{n-1,n-1}}$$

Allgemein: Für $j = n, n-1, \ldots, 1$ berechne

$$x_j = \frac{b_j - \sum_{k=j+1}^{n} r_{j,k}\, x_k}{r_{j,j}}$$

Gauss-Elimination/LR-Zerlegung

Verfahren:
Subtrahiere in $(A \mid \mathbf{b})$ Zeile j mit Faktor

$$\ell_{i,j} := \frac{a_{i,j}^{(j)}}{a_{j,j}^{(j)}}$$

von Zeile i.
Gilt im Gauss-Algorithmus stets $a_{j,j}^{(j)} \neq 0$, dann erhält man

$$A = LR$$

$$L = \begin{pmatrix} 1 & & & 0 \\ \ell_{2,1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \ell_{n,1} & \cdots & \ell_{n,n-1} & 1 \end{pmatrix}$$

Pivotierung:
verbessert Stabilität der Gausselemination / LR-Zerlegung
Skalierung: verbessert Konditionszahl der Matrix

Permutationsmatrix P:

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

Zeile 2 und 4 vertauschen:

$$P_{2,4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

LR-Zerlegung $((n^3 - 3)/3)$**:**
Ziel: $Ax = b$

$$A = LR$$

- L: untere Dreiecksmatrix
- R: obere Dreiecksmatrix

Es gilt:

$$PA = LR$$

und damit:

$$PAx = Pb$$

Setze:

$$Rx = y$$

Es entstehen zwei Dreieckssysteme (Rechenaufwand: n^2):

1. Vorwärtseinsetzen:
$$Ly = Pb$$
2. Rückwärtseinsetzen:
$$Rx = y$$

Cholesky-Verfahren

Jede symmetrisch positiv definite Matrix $A \in \mathbb{R}^{n\times n}$ besitzt eine eindeutige Zerlegung

$$A = LDL^T$$

-D = Diagonalmatrix mit Diagonaleinträgen ist

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \ell_{2,1} & 1 & 0 \\ \ell_{3,1} & \ell_{3,2} & 1 \end{pmatrix}, \qquad D = \begin{pmatrix} d_{1,1} & 0 & 0 \\ 0 & d_{2,2} & 0 \\ 0 & 0 & d_{3,3} \end{pmatrix}$$

$j = 1$:
$(1,1)$-Element: $d_{1,1} = a_{1,1}$
$(2,1)$-Element: $l_{2,1}\, d_{1,1} = a_{2,1} \quad \Rightarrow \quad l_{2,1} = \frac{a_{2,1}}{d_{1,1}}$
$(3,1)$-Element: $l_{3,1}\, d_{1,1} = a_{3,1} \quad \Rightarrow \quad l_{3,1} = \frac{a_{3,1}}{d_{1,1}}$
$j = 2$:
$(2,2)$-Element: $l_{2,1}^2 d_{1,1} + d_{2,2} = a_{2,2}$
$(3,2)$-Element: $l_{3,1} d_{1,1} l_{2,1} + l_{3,2} d_{2,2} = a_{3,2}$
$j = 3$:
$(3,3)$-Element: $l_{3,1}^2 d_{1,1} + l_{3,2}^2 d_{2,2} + d_{3,3} = a_{3,3} \quad \Rightarrow \quad d_{3,3} = a_{3,3} - l_{3,1}^2 d_{1,1} - l_{3,2}^2 d_{2,2}$

Vorgehen Cholesky

1. Vorwärtseinsetzen:
$$Ly = b$$
2. Diagonalsystem lösen:
$$Dz = y$$
3. Rückwärtseinsetzen:
$$L^T x = z$$

Hinweis: y und z sind Hilfsvektoren

Lineare Ausgleichsrechnung

Gesucht ist eine Ausgleichsgerade in einer definierten Form.
vorhandenes Wertepaar:

x_i	x_1	x_2	x_3	x_4
y_i	y_1	y_2	y_3	y_4

$z.B. \quad y = ax + b \qquad \to f_1 = \lambda \cdot x \quad | \quad f_2 = \lambda \cdot 1$

Lineares GLS in Matrix-Vektor-Form:

$$\underbrace{\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n f(x_i)\, x_i \\ \sum_{i=1}^n f(x_i) \end{pmatrix}}_{A^T A \cdot \lambda = A^T y}$$

Lineares Problem mit Fehlergleichungssystem lösen:

$$A \cdot \lambda = y$$

$$A = \begin{pmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_m(x_n) \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix}$$

Lösen mit Normalgleichungssystem:

$$A^T \cdot A \cdot \lambda = A^T \cdot y$$

Summe der Fehlerquadrate:

$$E = \sum_{i=1}^{m} (y_i - f(x_1))^2$$

Nichtlineare Gleichungen

Fixpunktiteration
Fixpunkt von f: x = f(x)

$$x_1 = f(x_0)$$
$$x_2 = f(x_1)$$
$$\vdots$$
$$x_n = f(x_{n-1}), \quad n = 1, 2, \ldots$$

Banachscher Fixpunktsatz:

$$|f(x_1) - f(x_2)| \leq K \cdot |x_1 - x_2|$$

Wichtig: Ableitung von f(x) < 1

$$\left|f'(x)\right| \leq K < 1$$

Intervallhalbierungsverfahren - Bisektion (Konvergenz=1):
Intervalllänge / Anzahl n:

$$b_n - a_n = \frac{b-a}{2^n} \qquad \frac{b_0 - a_0}{2^{n+1}} < \epsilon \quad (\cdot Fehlertoleranz \quad 1/2)$$

Mittepunktrechnung:

$$m_n = \frac{a_n + b_n}{2}$$

$$\text{Falls } f(m_n) \begin{cases} = 0, & \text{Abbruch, da } m_n \text{ Nullstelle,} \\ > 0, & a_{n+1} = a_n,\ b_{n+1} = m_n \quad (links) \\ < 0, & a_{n+1} = m_n,\ b_{n+1} = b_n \quad (rechts) \end{cases}$$

gilet nur, wenn f(a)<0 und f(b)>0 ist = sonst links/rechts vertauscht

---

Newton Tangentenverfahren:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (n = 0, 1, 2, \ldots)$$

Nichtlineare Gleichungssysteme

Fixpunktgleichung:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1(x_1,\dots,x_n) \\ \vdots \\ f_n(x_1,\dots,x_n) \end{pmatrix}$$

Nullstellengleichung:

$$\begin{pmatrix} g_1(x_1,\dots,x_n) \\ \vdots \\ g_n(x_1,\dots,x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Jacobi-Matrix:
$\mathbb{R}^n \to \mathbb{R}^m$ in $x_0 \in D$. Es gilt $J_f(x_0) \in \mathbb{R}^{m\times n}$

$$J_f(x_0) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Beschriebene Form:

$$f(x) = f(x_0) + J_f(x_0)\cdot(x-x_0) + \mathcal{O}(\|x-x_0\|^2)$$

Newton-Verfahren (Konvergenz=2) für Systeme:
Gesucht ist die Lösung von:

$$f(x^{(0)}) + J_f(x^{(0)})\cdot(x-x^{(0)}) = 0$$

Daraus folgt:

$$x^{(k+1)} = x^{(k)} - J_f(x^{(k)})^{-1}\cdot f(x^{(k)}) \qquad mit \qquad \underbrace{J_f(x^{(k)})^{-1}\cdot f(x^{(k)})}_{=\delta^{(k)}}$$

$$x^{(k+1)} = x^{(k)} - \delta^{(k)}$$

mit:

$$\delta^{(k)} = J_f(x^{(k)})^{-1}\cdot f(x^{(k)})$$

Vorgehen:

- Berechne
$$b^{(k)} = f(x^{(k)}), \quad A^{(k)} = J_f(x^{(k)})$$
- Löse das lineare Gleichungssystem in $\delta^{(k)}$
$$A^{(k)}\cdot\delta^{(k)} = b^{(k)}$$
- Setze (Newton-Schritt)
$$x^{(k+1)} = x^{(k)} - \delta^{(k)}$$

nichtlineare Ausgleichsrechnung

$$\vec{F}(\vec{\lambda}) = \Big(f(t_i,\vec{\lambda}) - b_i\Big)_{i=1,\dots,m} = \begin{pmatrix} F_1(\vec{\lambda}) \\ F_2(\vec{\lambda}) \\ \vdots \\ F_m(\vec{\lambda}) \end{pmatrix}$$

und Jacobi-Matrix von $\vec{F}(\vec{\lambda})$ als Funktion $J_{\vec{F}}(\vec{\lambda})$. Für $k = 0,1,2,\dots$:

- Berechne für k = 0,1,2...
$$\vec{b}^{(k)} = J_{\vec{F}}^{\top}(\vec{\lambda}^{(k)})\cdot\vec{F}(\vec{\lambda}^{(k)}), \qquad A^{(k)} = J_{\vec{F}}^{\top}(\vec{\lambda}^{(k)})\cdot J_{\vec{F}}(\vec{\lambda}^{(k)})$$
- Löse das lineare Gleichungssystem in $\vec{s}^{(k)}$
$$A^{(k)}\cdot\vec{s}^{(k)} = \vec{b}^{(k)}$$
- Setze (Newton-Schritt)
$$\vec{\lambda}^{(k+1)} = \vec{\lambda}^{(k)} - \vec{s}^{(k)}$$

Interpolation

Interpolation nach Lagrange:
$P_n(x)$ gesucht mit Grad n bei n+1 Stützpunkten.

$$P_n(x) = \sum_{i=0}^{n} y_i\cdot L_i(x)$$

mit Lagrange-Basispolynome (Achtung $j \neq i$):

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

Fehlerrechnung:

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

Newtonsche Interpolationsformel:
neue Stützpunkte können einfach darunter eingefügt werden.

x_i	f_i	$f[x_i,x_{i+1}]$	$f[x_i,x_{i+1},x_{i+2}]$
x_0	$f[x_0]$		
		$\searrow\nearrow\ \frac{f[x_1]-f[x_0]}{x_1-x_0} = f[x_0,x_1]$	
x_1	$f[x_1]$		$\searrow\nearrow\ \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} = f[x_0,x_1,x_2]$
		$\searrow\nearrow\ \frac{f[x_2]-f[x_1]}{x_2-x_1} = f[x_1,x_2]$	
x_2	$f[x_2]$		

daraus folgt von oben oder von unten:

$$p(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \dots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

Newton-cotes-Quadraturformel:
Bei geradem m: Exaktheit bis Grad m+1

Name	m	w_i						
Trapezregel	1	$\frac{1}{2}$	$\frac{1}{2}$					
Simpsonregel	2	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$				
$3/8-$Regel	3	$\frac{3}{8}$	$\frac{9}{8}$	$\frac{9}{8}$	$\frac{3}{8}$			
Milneregel	4	$\frac{14}{45}$	$\frac{64}{45}$	$\frac{8}{15}$	$\frac{64}{45}$	$\frac{14}{45}$		
	5	$\frac{95}{288}$	$\frac{125}{96}$	$\frac{125}{144}$	$\frac{125}{144}$	$\frac{125}{96}$	$\frac{95}{288}$	
Weddleregel	6	$\frac{41}{140}$	$\frac{54}{35}$	$\frac{9}{70}$	$\frac{68}{35}$	$\frac{9}{70}$	$\frac{54}{35}$	$\frac{41}{140}$

Trapezregel:

$$I = \int_a^b f(x)\,dx \approx h\cdot\left(\frac{1}{2}f(a)+\frac{1}{2}f(b)\right) = Q_h^T(f) \quad \text{mit} \quad h = b-a$$

Simpsonregel:

$$I = \int_a^b f(x)\,dx \approx h\cdot\left(\frac{1}{3}f(a)+\frac{4}{3}f(a+h)+\frac{1}{3}f(b)\right) = Q_h^S(f) \quad \text{mit} \quad h = \frac{b-a}{2}$$

summierte Trapezregel:

$$\frac{h}{2}\left(f(a)+2\sum_{k=1}^{n-1}f(a+kh)+f(b)\right)$$

Gauss Quadraturformel:
Transformation von f(x) ϵ[a,b] zu f(x) ϵ[-1,1]

$$x = \frac{b-a}{2}\cdot t + \frac{a+b}{2} \qquad dx = \frac{b-a}{2}dt$$

Stützstellen gemäss Legendrepolynom:

n	Stützstelle t_i	Gewichtung ω_i
1	0	2
2	$-\frac{1}{\sqrt{3}}$ $\quad$ $\frac{1}{\sqrt{3}}$	1 $\quad$ 1
3	$-\sqrt{\frac{3}{5}}$ $\quad$ 0 $\quad$ $\sqrt{\frac{3}{5}}$	$\frac{5}{9}$ $\quad$ $\frac{8}{9}$ $\quad$ $\frac{5}{9}$

Einsetzen:

$$\int_{-1}^{1} f(t)\,\mathrm{d}t \approx \sum_{k=1}^{n}\omega_k\,f(t_k)$$

Spline-Funktionen:

- periodisch interpolierender Spline, gilt:
$$s(a) = s(b), \quad s'(a) = s'(b), \quad s''(a) = s''(b)$$
- natürlich interpolierender Spline, gilt:
$$s''(a) = s''(b) = 0$$

Satz der Momente: Bei $x_i - x_{i-1} =$ const gilt $\mu = 0.5$

$$\mu_i M_{i-1} + 2M_i + (1-\mu_i)M_{i+1} = 6f[x_{i-1},x_i,x_{i+1}]$$

für $i = 1,\dots,n-1$, wobei $\mu_i = \frac{x_i - x_{i-1}}{x_{i+1}-x_{i-1}}$.

Berechnung Splines:

$$s(x) = M_{i-1}\frac{(x_i-x)^3}{6h_i} + M_i\frac{(x-x_{i-1})^3}{6h_i} + C_i\left(x - \frac{x_{i-1}+x_i}{2}\right) + D_i$$

wobei $h_i := x_i - x_{i-1}$ und

$$C_i := \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1})$$

$$D_i := \frac{y_i + y_{i-1}}{2} - \frac{h_i^2}{12}(M_i + M_{i-1})$$

numerische Verfahren für AWP

$$h = \frac{b-a}{n}$$
$$y' = f(x_i,y_i)$$

explizites Eulerverfahren:
Gegeben:

$$y'(x) = f(x,y(x)) \quad \text{mit} \quad y(x_0) = y_0$$

schrittweise berechnen von $y_k \approx y(x_k)$ durch

$$y_{k+1} = y_k + h\cdot f(x_k,y(x_k)) \quad \text{für } k = 0,1,\dots,n-1$$

implizites Eulerverfahren:
Gegeben:

$$y'(x) = f(x,y(x)) \quad \text{mit} \quad y(x_0) = y_0$$

schrittweise berechnen von $y_k \approx y(x_k)$ durch

$$y_{k+1} = y_k + h\cdot f(x_{k+1},y_{k+1}) \quad \text{für } k = 0,1,\dots,n-1$$

Verfahren nach Runge (p=2)

$$r_1 = f(x_k, y_k)$$
$$r_2 = f\left(x_k + \frac{h}{2},\, y_k + \frac{h}{2} r_1\right)$$
$$y_{k+1} = y_k + h\, r_2$$

Verfahrensfunktion:

$$\phi(x_k, y_k, h) = f\left(x_k + \frac{h}{2},\, y_k + \frac{h}{2} f(x_k, y_k)\right)$$

Heun-Verfahren (p=2)

$$r_1 = f(x_k, y_k)$$
$$r_2 = f(x_k + h, y_k + h r_1)$$
$$y_{k+1} = y_k + \frac{h}{2}(r_1 + r_2)$$

---

klassische Runge-Kutte-Verfahren (p=4)

$$r_1 = f(x_k, y_k)$$
$$r_2 = f\left(x_k + \frac{1}{2}h,\, y_k + \frac{1}{2} h r_1\right)$$
$$r_3 = f\left(x_k + \frac{1}{2}h,\, y_k + \frac{1}{2} h r_2\right)$$
$$r_4 = f(x_k + h,\, y_k + h r_3)$$
$$y_{k+1} = y_k + \frac{h}{6}(r_1 + 2r_2 + 2r_3 + r_4)$$

Butcher-Schema

Euler-Verfahren

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

Runge-Verfahren (Mittelpunktsverfahren)

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$$

Verfahren von Heun

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Runge–Kutta-Verfahren (4-stufig)

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & & \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

links: welcher Faktor für h in x

rechts: welcher Faktor für y

Verfahrensanalyse

Stabilität: Qualitatives Verhalten von y_k für $k \to \infty$

Konvergenz: Für $h \to 0$ strebt y_k gegen die exakte Lösung

$$y_k \to y(t_k)$$

Konvergenzordnung: Beschreibt, wie schnell y_k gegen $y(t_k)$ konvergiert.
Ein Verfahren hat die *Konvergenzordnung* p, wenn

$$|u(t_k) - u_k| < Ch^p = \mathcal{O}(h^p), \qquad C = \text{const}$$

Lösen von AWP

Gegeben sei ein Anfangswertproblem der Form

$$x^{(n)} = f\left(t, x, x', x'', \ldots, x^{(n-1)}\right)$$

1. Nach der höchsten Ableitung auflösen

$$x^{(n)} = f\left(t, x, x', \ldots, x^{(n-1)}\right)$$

Ziel ist die Form:

$$y' = f(t, y)$$

2. Umwandlung in ein System erster Ordnung

Neue Variablen einführen:

$$y_1 = x \qquad y_2 = x' \qquad y_3 = x''$$

Dann ergibt sich:

$$y_1' = y_2 \qquad y_2' = y_3$$

Somit erhält man ein System:

$$y' = \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ f(t, y) \end{pmatrix}$$

3. Anfangswerte umschreiben

$$y_1(t_0) = x(t_0) \qquad y_2(t_0) = x'(t_0)$$

4. Schrittweite festlegen

Die Schrittweite lautet:

$$h = \frac{b - a}{N}$$

mit Intervall $[a, b]$ und Anzahl der Schritte N.
Die Stützstellen sind:

$$t_n = t_0 + nh$$

5. Anwendung

Einsetzen in numerisches Verfahren

Merksatz

Auflösen → Variablen einführen → System bilden → Verfahren anwenden → iterativ rechnen

$$h_{max} = \frac{2}{\lambda_{max}}$$