

## Numerische Lösung nicht linearer Gleichungssysteme

LGS = lineares Gleichungssystem, NGS = nichtlineares Gleichungssystem

**Skalarwertige Funktionen**  $f: D \subset \mathbb{R}^n \rightarrow W \subset \mathbb{R}$   
 $(x_1, x_2, \dots, x_n) \mapsto y = f(x_1, x_2, \dots, x_n)$

$f$  mit  $n$  unabhängigen Variablen  $x_1, \dots, x_n$  und einer abhängigen Variablen  $y$ , die jedem  $(x_1, x_2, \dots, x_n)$  aus Definitionsmenge  $D \subset \mathbb{R}^n$  genau ein  $y \in W \subset \mathbb{R}$  zuordnet. Ergebnis:  $y \in \mathbb{R}$  = Skalar (eine Zahl)

**Vektorwertige Funktion** gibt einen **Vektor** zurück (statt Skalar)  
 Sei  $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^m$  eine Funktion mit  $n$  Variablen.

$$\mathbf{f}(x_1, \dots, x_n) = \begin{pmatrix} y_1 = f_1(x_1, x_2, \dots, x_n) \\ y_2 = f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ y_m = f_m(x_1, x_2, \dots, x_n) \end{pmatrix}$$

wobei die  $m$  Komponenten  $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$  für  $i = 1, 2, \dots, m$  von  $\mathbf{f}$  wieder **skalarwertige** Funktionen sind.

### Nichtlineares Gleichungssystem (NGS)

Lösungen des NGS sind Nullstellen der Funktion:

$$\mathbf{f}: \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad \mathbf{f}(x) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Ein solches System lässt sich nicht in die Form  $Ax = b$  bringen.  
 Geometrisch lassen sich die Lösungen als Schnittpunkte der beiden Funktionen interpretieren.

### Lineare Funktionen von LGS

$$A\vec{x} = \vec{b} \Rightarrow \underbrace{A\vec{x} - \vec{b}}_{\vec{f}(\vec{x})} = \vec{0} \Rightarrow \vec{f}(x_1, x_2, x_3) = 0 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\vec{f}(\vec{x}) = A\vec{x} - \vec{b} = \begin{pmatrix} 4 & -1 & 1 \\ -2 & 5 & 1 \\ 1 & -2 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \begin{pmatrix} 5 \\ 11 \\ 12 \end{pmatrix}, \quad \mathbf{f}(x_1, x_2, x_3) = \begin{pmatrix} f_1 = 4x_1 - x_2 + x_3 - 5 \\ f_2 = -2x_1 + 5x_2 + x_3 - 11 \\ f_3 = x_1 - 2x_2 + 5x_3 - 12 \end{pmatrix}$$

### Analytische Darstellung

- **Explizite Darstellung:**  $y = f(x_1, \dots, x_n)$
- **Implizite Darstellung:**  $F(x, y) = 0$
- **Parameterdarstellung:**  $x = x(t), y = y(t)$

**Darstellung durch Wertetabelle** Sei  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  eine Funktion.  
 In  $z = f(x, y)$  Werte von  $x$  und  $y$  einsetzen (der Reihe nach):

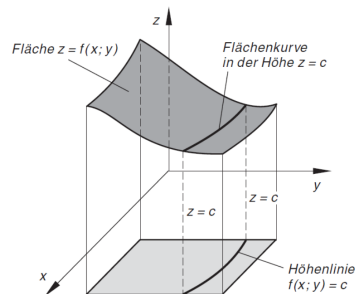
$$\begin{pmatrix} z_{11} & z_{12} & \dots & z_{1m} \\ z_{m1} & z_{m2} & \dots & z_{mn} \end{pmatrix}$$

### Funktion als Fläche im Raum

$f$  ordnet jedem Punkt  $(x, y) \in D$  in Ebene Wert  $z = f(x, y)$  zu  
 ( $\rightarrow$  Höhenkoordinate)

### Schnittkurviendiagramm

Fläche  $z = f(x, y)$  bei konstanten Höhe  $z$  schneiden: Schnittkurve.  
 Diese in  $(x, y)$ -Ebene projizieren: Höhenlinie.



hellgraue Fläche = Definitionsbereich D

## Partielle Ableitungen

**Partielle Ableitung**  $f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$

Ableitung nach  $x$ :  $f_x = \frac{\partial f}{\partial x}(x, y) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$

Ableitung nach  $y$ :  $f_y = \frac{\partial f}{\partial y}(x, y) = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$

### Partielle Ableitungen berechnen

1. Variable identifizieren: nach welcher Variable ableiten?
2. Alle anderen Variablen während Ableitung nur Konstanten
3. Standardableitungsregeln anwenden und Ergebnis korrekt notieren

**Jacobi-Matrix**  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  mit  $y = f(x)$  und  $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$   
 Jacobi-Matrix enthält alle partiellen Ableitungen 1. Ordnung von  $f$ :

$$f(x) = \begin{pmatrix} y_1 = f_1(x) \\ y_2 = f_2(x) \\ \vdots \\ y_m = f_m(x) \end{pmatrix} \rightarrow Df(x) := \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \dots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) & \dots & \frac{\partial f_m}{\partial x_n}(x) \end{bmatrix}$$

### Linearisierung Die verallgemeinerte Tangentengleichung

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

beschreibt lineare Funktion,  $f(x) \approx g(x)$  in Umgebung von  $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T \in \mathbb{R}^n$ . Man spricht von der **Linearisierung** der Funktion  $y = f(x)$  in einer Umgebung von  $x^{(0)}$  ( $x^{(k)}$  bezeichnet Vektor aus  $\mathbb{R}^n$  nach  $k$ -ter Iteration).

**Tangentialebene**  $f: \mathbb{R}^2 \rightarrow \mathbb{R}, y = f(x_1, x_2), x^{(0)} = (x_1^{(0)}, x_2^{(0)})^T \in \mathbb{R}^2$   
 Spezielle Jacobi-Matrix (nur ein Zeilenvektor mit zwei Elementen):

$$Df(x^{(0)}) = \left( \frac{\partial f}{\partial x_1}(x_1^{(0)}, x_2^{(0)}), \frac{\partial f}{\partial x_2}(x_1^{(0)}, x_2^{(0)}) \right)$$

Linearisierung  $g(x_1, x_2)$  die Gleichung der Tangentialebene:

$$\begin{aligned} &= f(x_1^{(0)}, x_2^{(0)}) + \left( \frac{\partial f}{\partial x_1}(x_1^{(0)}, x_2^{(0)}), \frac{\partial f}{\partial x_2}(x_1^{(0)}, x_2^{(0)}) \right) \cdot \begin{pmatrix} x_1 - x_1^{(0)} \\ x_2 - x_2^{(0)} \end{pmatrix} \\ &= f(x_1^{(0)}, x_2^{(0)}) + \frac{\partial f}{\partial x_1}(x_1^{(0)}, x_2^{(0)}) \cdot (x_1 - x_1^{(0)}) + \frac{\partial f}{\partial x_2}(x_1^{(0)}, x_2^{(0)}) \cdot (x_2 - x_2^{(0)}) \end{aligned}$$

Sie enthält sämtliche im Flächenpunkt  $\dot{P} = (x_1^{(0)}, x_2^{(0)}, f(x_1^{(0)}, x_2^{(0)}))$  an die Bildfläche von  $y = f(x_1, x_2)$  angelegten Tangenten.

### Jacobi-Matrix berechnen und linearisieren

Sei  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  mit  $y = f(x)$  und  $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ .

1. Identifiziere die Komponentenfunktionen  $f_1, f_2, \dots, f_m$  und Variablen  $x_1, x_2, \dots, x_n$ .
2. Berechne partielle Ableitungen  $\frac{\partial f_i}{\partial x_j}$  für  $i = 1, \dots, m, j = 1, \dots, n$ .
3. Stelle die Jacobi-Matrix  $Df(x)$  auf
4. Werte Jacobi-Matrix an Entwicklungspunkt  $x^{(0)}$  aus (Werte für  $x_1, x_2, \dots, x_n$  einsetzen)
5. Berechne Linearisierung  $g(x)$  mit Tangentengleichung

Struggle  $\in \mathbb{R}$

**Jacobi-Matrix und Linearisierung**  $f(x, y, z) = \begin{pmatrix} e^{xy} + z^2 - 3 \\ \sin(x+y) - z \\ x^2 + y^2 + z^2 - 6 \end{pmatrix}$

**Jacobi-Matrix:**  $Df(x, y, z) = \begin{bmatrix} ye^{xy} & xe^{xy} & 2z \\ \cos(x+y) & \cos(x+y) & -1 \\ 2x & 2y & 2z \end{bmatrix}$

**Linearisierung an  $(1, 0, 1)^T$ :**

$$f(1, 0, 1) = \begin{pmatrix} e^0 + 1 - 3 \\ \sin(1) - 1 \\ 1 + 0 + 1 - 6 \end{pmatrix} = \begin{pmatrix} -1 \\ \sin(1) - 1 \\ -4 \end{pmatrix}, \quad Df(1, 0, 1) = \begin{bmatrix} 0 & 1 & 2 \\ \cos(1) & \cos(1) & -1 \\ 2 & 0 & 2 \end{bmatrix}$$

Linearisierung:  $g(x, y, z) = f(1, 0, 1) + Df(1, 0, 1) \cdot \begin{pmatrix} x-1 \\ y-0 \\ z-1 \end{pmatrix}$

$$g(x, y, z) = \begin{pmatrix} -1 + y + 2(z-1) \\ (\sin(1) - 1) + \cos(1)(x-1) + \cos(1)y - (z-1) \\ -4 + 2(x-1) + 2(z-1) \end{pmatrix}$$

**Geometrische Bedeutung:** Linearisierung approximiert nichtlineare Funktion  $f$  nahe des Punktes  $(1, 0, 1)^T$  durch lineare Funktion. Entspricht der Tangentialebene an die durch  $f = 0$  definierte Fläche im 3D Raum.

## Nullstellenbestimmung für NGS

### Problemstellung zur Nullstellenbestimmung

Gegeben:  $n \in \mathbb{N}, f: \mathbb{R}^n \rightarrow \mathbb{R}^n$  Gesucht: Vektor  $\vec{x} \in \mathbb{R}^n$  mit  $f(\vec{x}) = 0$   
 Komponentenweise: Gegeben:  $n$  Funktionen  $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$  (Komponenten von  $f$ ) Gesucht: Vektor  $\vec{x} \in \mathbb{R}^n$  mit  $f_i(\vec{x}) = 0$  für  $i = 1, \dots, n$ .

### Newton-Verfahren für NGS (Quadratische Konv.)

Gesucht: Nullstellen von  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$

$x^{(0)}$  = Startvektor nahe einer Nullstelle

**Vorbereitung:** definiere  $f(x) = 0$ , berechne  $Df(x)$ , wähle  $x^{(0)}$

**Für jede Iteration  $n$ :**

1. Linearisierung um  $x^n$ : Berechne  $f(x^{(n)})$  und  $Df(x^{(n)})$
  2. Nullstellen der Linearisierung:  $\delta^{(n)}$  als Lösung des LGS  $Df(x^{(n)}) \cdot \delta^{(n)} = -f(x^{(n)})$
  3. Setze  $x^{(n+1)} := x^{(n)} + \delta^{(n)}$  (nächste Iteration)
  4. Weiterführen bis:  $\|f(x^{(n+1)})\|_2 < \text{TOL}$  oder  $\|x^{(n+1)} - x^{(n)}\|_2 < \text{TOL}$ .
- Interpretation:** Konvergierte Lösung  $x^{(n)}$  = Näherung für Nullstelle von  $f$

### Vereinfachtes Newton-Verfahren (Lineare Konvergenz)

Lösung von  $f(x) = 0$  mit  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$  für  $n = 0, 1, 2, \dots$

1. Berechne  $f(x^{(n)})$  und  $Df(x^{(0)})$
2. Berechne  $\delta^{(n)}$  als Lösung des LGS  $Df(x^{(0)}) \cdot \delta^{(n)} = -f(x^{(n)})$
3. Setze  $x^{(n+1)} := x^{(n)} + \delta^{(n)}$
4. Weiterführen bis:  $\|f(x^{(n+1)})\|_2 < \text{TOL}$  oder  $\|x^{(n+1)} - x^{(n)}\|_2 < \text{TOL}$ .

### Fehler-Normen

- $\|f(x)\|_2 = \sqrt{\sum_{i=1}^n f_i(x)^2}$ : Euklidische Norm
- $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$ : Euklidische Norm für Vektoren
- $\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$ : Operatornorm für Matrizen

**Newton-Verfahren**  $f(x_1, x_2) = \begin{pmatrix} 20 - 18x_1 - 2x_2^2 \\ -4x_2 - 4(x_1 - 3x_2^2) \end{pmatrix}, x^{(0)} = (1.1, 0.9)^T$

**Jacobi-Matrix:**  $Df(x_1, x_2) = \begin{bmatrix} -18 & -4x_2 \\ -4x_2 & -4(x_1 - 3x_2^2) \end{bmatrix}$

**Erste Iteration:** ( $k = 0$ )  $f(1.1, 0.9) = \begin{pmatrix} -1.42 \\ -0.036 \end{pmatrix}$

$$Df(1.1, 0.9) = \begin{bmatrix} -18 & -3.6 \\ -3.6 & -5.32 \end{bmatrix}$$

LGS lösen:  $\begin{bmatrix} -18 & -3.6 \\ -3.6 & -5.32 \end{bmatrix} \delta^{(0)} = \begin{pmatrix} 1.42 \\ 0.036 \end{pmatrix} \Rightarrow \delta^{(0)} = \begin{pmatrix} -0.0822 \\ 0.0178 \end{pmatrix}$

$$x^{(1)} = \begin{pmatrix} 1.1 \\ 0.9 \end{pmatrix} + \begin{pmatrix} -0.0822 \\ 0.0178 \end{pmatrix} = \begin{pmatrix} 1.0178 \\ 0.9178 \end{pmatrix}$$

Weitere Iterationen führen zur Konvergenz.

Gedämpftes Newton-Verfahren

**Dämpfung für bessere Konvergenz** Für schlecht konditionierte Jacobi-Matrix  $Df(x^{(n)})$  kann Standard Newton-Verfahren divergieren. Dämpfung = variable Schrittweite:  $x^{(n+1)} = x^{(n)} + \frac{\delta^{(n)}}{2^p}$   $p$  = kleinstes Element aus  $\{0, 1, \dots, p_{\max}\}$  für das gilt:  $\|f(x^{(n)} + \frac{\delta^{(n)}}{2^p})\|_2 < \|f(x^{(n)})\|_2$

Gedämpftes Newton-Verfahren

Nur in der Nähe der Nullstelle ist Konvergenz des Verfahrens garantiert!

1. Berechne  $f(x^{(n)})$  und  $Df(x^{(n)})$
2. Berechne  $\delta^{(n)}$  als Lösung des lin. GS  $Df(x^{(n)}) \cdot \delta^{(n)} = -f(x^{(n)})$
3. Finde das minimale  $p \in \{0, 1, \dots, p_{\max}\}$  mit:  
$$\|f(x^{(n)} + \frac{\delta^{(n)}}{2^k})\|_2 < \|f(x^{(n)})\|_2$$

Kein minimales  $k$  gefunden  $\rightarrow k = 0$

4. Setze  $x^{(n+1)} := x^{(n)} + \frac{\delta^{(n)}}{2^k}$

Ausgleichsrechnung

**Ausgleichsproblem** ('Polyfit')  
Gegeben:  $n$  Wertepaare  $(x_i, y_i)$ ,  $i = 1, \dots, n$  mit  $x_i \neq x_j$  für  $i \neq j$ .  
Gesucht: stetige Funktion  $f: \mathbb{R} \rightarrow \mathbb{R}$ , die die Wertepaare bestmöglich annähert (es soll gelten)  $f(x_i) \approx y_i$  für alle  $i = 1, \dots, n$

**Fehlerfunktional und kleinste Fehlerquadrate**  
Eine Ausgleichsfunktion  $f$  minimiert das **Fehlerfunktional**:

$$E(f) := \|y - f(x)\|_2^2 = \sum_{i=1}^n (y_i - f(x_i))^2$$

Gefundenes  $f$  optimal im Sinne der **kleinsten Fehlerquadrate** (least squares fit).

Lineare Ausgleichsprobleme

**Lineares Ausgleichsproblem**  
Gegeben:  $n$  Wertepaare  $(x_i, y_i)$  und  $m$  Basisfunktionen  $f_1, \dots, f_m$   
Ansatzfunktion:  $f(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x) + \dots + \lambda_m f_m(x)$   
Fehlerfunktional:  $E(f) = \|y - A\lambda\|_2^2$

wobei  $A$  die  $n \times m$  Matrix ist:  $A = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \dots & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & \dots & f_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n) & f_2(x_n) & \dots & f_m(x_n) \end{bmatrix}$

**Normalgleichungen** Die Lösung des linearen Ausgleichsproblems ergibt sich aus dem **Normalgleichungssystem**:  $A^T A \lambda = A^T y$   
Nach Lösung des Normalgleichungssystems erhält man die Koeffizienten für die optimale Ausgleichsfunktion.  
Für bessere numerische Stabilität:  
QR-Zerlegung  $A = QR$  verwenden!  $R\lambda = Q^T y$

Lineare Ausgleichsrechnung durchführen

- **Basisfunktionen bestimmen**  $f_1(x), f_2(x), \dots, f_m(x)$
- **Matrix  $A$  Berechnen**  $A_{ij} = f_j(x_i) \forall i = 1, \dots, n$  und  $j = 1, \dots, m$
- **Normalgleichungssystem Berechnen**  $A^T A$  und  $A^T y$
- **LGS lösen** Löse  $A^T A \lambda = A^T y$
- **Ausgleichsfunktion**  $f(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x) + \dots + \lambda_m f_m(x)$
- **Fehlerfunktional** Berechne  $E(f) = \|y - A\lambda\|_2^2$
- **Konvergenz** Prüfe, ob  $E(f)$  klein genug ist (z.B.  $\leq 10^{-6}$ )

**Lineare Ausgleichsrechnung** Ausgleichsgerade  $f(x) = ax + b$  für:

$x_i$	1	2	3	4
$y_i$	6	6.8	10	10.5

**Basisfunktionen:**  $f_1(x) = x, f_2(x) = 1$

**Matrix  $A$ :**  $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}, y = \begin{pmatrix} 6 \\ 6.8 \\ 10 \\ 10.5 \end{pmatrix}$

**Normalgleichungen:**  $A^T A = \begin{bmatrix} 30 & 10 \\ 10 & 4 \end{bmatrix}, A^T y = \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix}$

**LGS lösen:**  $\begin{bmatrix} 30 & 10 \\ 10 & 4 \end{bmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix}$  Lösung:  $a = 1.67, b = 4.15$

Die **Ausgleichsgerade** lautet:  $f(x) = 1.67x + 4.15$

**Residuen berechnen:**  $r_i = y_i - f(x_i)$

$i$	$y_i$	$f(x_i)$	$r_i$
1	6	5.82	0.18
2	6.8	7.49	-0.69
3	10	9.16	0.84
4	10.5	10.83	-0.33

**Residuenvektor:**  $r = \begin{pmatrix} 0.18 \\ -0.69 \\ 0.84 \\ -0.33 \end{pmatrix}$

**Residuenquadrate:**  
 $r_i^2 = 0.0324, 0.4761, 0.7056, 0.1089$

**Fehlerfunktional:** (Summe der Residuenquadrate)

$E(f) = \|y - A\lambda\|_2^2 = \sum_{i=1}^n r_i^2 = 1.323$

Nichtlineare Ausgleichsprobleme

**Allgemeines Ausgleichsproblem** Gegeben:  $n$  Wertepaare  $(x_i, y_i)$  und nichtlineare Ansatzfunktion  $f_p(x, \lambda_1, \dots, \lambda_m)$  mit  $m$  Parametern  
Allgemeines Ausgleichsproblem: bestimme Parameter  $\lambda_1, \dots, \lambda_m$  so dass das Fehlerfunktional minimal wird:

$$E(\lambda) = \sum_{i=1}^n (y_i - f_p(x_i, \lambda_1, \dots, \lambda_m))^2$$

Gauss-Newton-Verfahren

löst nichtlineare Ausgleichsprobleme durch Linearisierung:

$$g(\lambda) := y - f(\lambda)$$

$\rightarrow$  Problem äquivalent zur Minimierung von  $\|g(\lambda)\|_2^2$ .

In jeder Iteration  $g(\lambda)$  linearisieren:

$$g(\lambda) \approx g(\lambda^{(k)}) + Dg(\lambda^{(k)}) \cdot (\lambda - \lambda^{(k)})$$

Gauss-Newton-Verfahren

**Funktionen definieren**  $g(\lambda) := y - f(\lambda)$  und  $Dg(\lambda)$  berechnen

**Iterationsschleife** Für  $k = 0, 1, \dots$ :

- Löse das lineare Ausgleichsproblem:

$$\min \|g(\lambda^{(k)}) + Dg(\lambda^{(k)}) \cdot \delta^{(k)}\|_2^2$$

- Das ergibt:

$$Dg(\lambda^{(k)})^T Dg(\lambda^{(k)}) \delta^{(k)} = -Dg(\lambda^{(k)})^T g(\lambda^{(k)})$$

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

Dämpfung (optional)

Bei Konvergenzproblemen:  $\lambda^{(k+1)} = \lambda^{(k)} + \frac{\delta^{(k)}}{2^p}$  mit geeignetem  $p$ .

Konvergenzprüfung

Abbruch wenn  $\|\delta^{(k)}\| < \text{TOL}$  oder  $\|g(\lambda^{(k+1)})\| < \text{TOL}$ .

Wahl zwischen linearer und nichtlinearer Ausgleichsrechnung:

- **Linear:** Wenn die Ansatzfunktion linear in den Parametern ist
- **Nichtlinear:** Wenn Parameter "verwoben" mit der Funktionsgleichung
- **Stabilität:** Gedämpfte Verfahren sind robuster, aber aufwendiger

Gauss-Newton-Verfahren

**Aufgabe:** Fitten Sie die Funktion  $f(x) = a \cdot e^{-bx} + c$  an die Datenpunkte:

$x$	0	1	2	3
$y$	5.2	3.8	3.1	2.9

**Funktionen definieren:**  $g(\lambda) = y - f(\lambda) = \begin{pmatrix} 5.2 \\ 3.8 \\ 3.1 \\ 2.9 \end{pmatrix} - \begin{pmatrix} ae^{-b \cdot 0} + c \\ ae^{-b \cdot 1} + c \\ ae^{-b \cdot 2} + c \\ ae^{-b \cdot 3} + c \end{pmatrix}$

Jacobi-Matrix von  $g$ :  $\frac{\partial g_i}{\partial a} = -e^{-bx_i}, \frac{\partial g_i}{\partial b} = ax_i e^{-bx_i}, \frac{\partial g_i}{\partial c} = -1$

$$Dg(\lambda) = \begin{bmatrix} -e^{-b \cdot 0} & a \cdot 0 \cdot e^{-b \cdot 0} & -1 \\ -e^{-b \cdot 1} & a \cdot 1 \cdot e^{-b \cdot 1} & -1 \\ -e^{-b \cdot 2} & a \cdot 2 \cdot e^{-b \cdot 2} & -1 \\ -e^{-b \cdot 3} & a \cdot 3 \cdot e^{-b \cdot 3} & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & -1 \\ -e^{-b} & ae^{-b} & -1 \\ -e^{-2b} & 2ae^{-2b} & -1 \\ -e^{-3b} & 3ae^{-3b} & -1 \end{bmatrix}$$

**Gauss-Newton-Schritt mit  $\lambda^{(0)} = (2, 0.5, 2.5)^T$ :**

$$f(\lambda^{(0)}) = \begin{pmatrix} 2e^{-0.5} + 2.5 \\ 2e^{-1} + 2.5 \\ 2e^{-1.5} + 2.5 \end{pmatrix} = \begin{pmatrix} 4.5 \\ 3.71 \\ 2.95 \end{pmatrix}$$

$$g(\lambda^{(0)}) = \begin{pmatrix} 5.2 \\ 3.8 \\ 3.1 \\ 2.9 \end{pmatrix} - \begin{pmatrix} 4.5 \\ 3.71 \\ 3.24 \\ 2.95 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 0.09 \\ -0.14 \\ -0.05 \end{pmatrix}$$

$$Dg(\lambda^{(0)}) = \begin{bmatrix} -1 & 0 & -1 \\ -0.606 & 1.213 & -1 \\ -0.368 & 1.472 & -1 \\ -0.223 & 1.340 & -1 \end{bmatrix}$$

Normalgleichungssystem:  $Dg^T Dg \delta = -Dg^T g$

Nach Lösung:  $\delta^{(0)} = \begin{pmatrix} 0.32 \\ -0.18 \\ 0.61 \end{pmatrix} \lambda^{(1)} = \lambda^{(0)} + \delta^{(0)} = \begin{pmatrix} 2.32 \\ 0.32 \\ 3.11 \end{pmatrix}$

**Physikalische Interpretation:** Die Funktion  $f(x) = ae^{-bx} + c$  beschreibt einen exponentiellen Abfall mit:

- $a = 2.32$ : Anfangsamplitude des abfallenden Anteils
- $b = 0.32$ : Abfallkonstante (je größer, desto schneller der Abfall)
- $c = 3.11$ : Asymptotischer Grenzwert für  $x \rightarrow \infty$

Dies könnte z.B. einen Abkühlungsprozess, radioaktiven Zerfall oder Entladung eines Kondensators beschreiben.

Interpolation

Interpolation: Spezialfall der linearen Ausgleichsrechnung. Suche zu einer Menge von vorgegebenen Punkten eine Funktion, die exakt durch diese Punkte verläuft.

**Interpolationsproblem** Gegeben:  $n + 1$  Wertepaare  $(x_i, y_i)$

$i = 0, \dots, n$ , mit  $x_i \neq x_j$  für  $i \neq j$ .

Gesucht: stetige Funktion  $g$  mit Eigenschaft  $g(x_i) = y_i \forall i = 0, \dots, n$ .

**Stützpunkte:**  $n + 1$   $(x_i, y_i)$ , **Stützstellen:**  $x_i$ , **Stützwerte:**  $y_i$

Interpolation vs. Ausgleichsrechnung

- **Interpolation:** Gesuchte Funktion geht **exakt** durch alle Datenpunkte
- **Ausgleichsrechnung:** Funktion **approximiert** die Datenpunkte möglichst gut
- **Interpolation:** Spezialfall der Ausgleichsrechnung ( $m = n, E(f) = 0$ )

Lagrange Interpolationsformel

Durch  $n + 1$  Stützpunkte mit verschiedenen Stützstellen  $\exists$  genau ein Polynom  $P_n(x)$  vom Grade  $\leq n$ , das alle Stützpunkte interpoliert.  $P_n(x)$  lautet in der Lagrangeform:  $P_n(x) = \sum_{i=0}^n l_i(x)y_i$   
 $l_i(x)$  (Lagrangepolynome vom Grad  $n$ ):  $l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$

Fehlerabschätzung

$y_i$  = Funktionswerte einer genügend oft stetig differenzierbaren Funktion  $f$  (also  $y_i = f(x_i)$  ), dann Interpolationsfehler an Stelle  $x$ :  
 $|f(x) - P_n(x)| \leq \frac{|(x-x_0)(x-x_1)\dots(x-x_n)|}{(n+1)!} \max_{x_0 \leq \xi \leq x_n} f^{(n+1)}(\xi)$

Lagrange-Interpolation durchführen

Gegeben:  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  und gesuchter Punkt  $x$ .  
Lagrangepolynome  $l_i(x)$ : Für  $i = 0, 1, \dots, n$  berechne:

$$l_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}$$

Interpolationspolynom:  $P_n(x) = y_0 \cdot l_0(x) + y_1 \cdot l_1(x) + \cdots + y_n \cdot l_n(x)$   
Funktionswert berechnen: Setze gewünschten  $x$ -Wert ein:  
 $P_n(x)$  = gesuchter Interpolationswert

Lagrange-Interpolation Bestimme Atmosphärendruck bei 3750m:

Höhe [m]	0	2500	5000	10000
Druck [hPa]	1013	747	540	226

Stützpunkte  $(0, 1013), (2500, 747), (5000, 540)$  für  $x = 3750$ .  
Lagrangepolynome:

$$l_0(3750) = \frac{(3750 - 2500)(3750 - 5000)}{(0 - 2500)(0 - 5000)} = \frac{1250 \cdot (-1250)}{(-2500) \cdot (-5000)} = -0.125$$
  
$$l_1(3750) = \frac{(3750 - 0)(3750 - 5000)}{(2500 - 0)(2500 - 5000)} = \frac{3750 \cdot (-1250)}{2500 \cdot (-2500)} = 0.75$$
  
$$l_2(3750) = \frac{(3750 - 0)(3750 - 2500)}{(5000 - 0)(5000 - 2500)} = \frac{3750 \cdot 1250}{5000 \cdot 2500} = 0.375$$

Interpolationswert:  
 $P(3750) = 1013 \cdot (-0.125) + 747 \cdot 0.75 + 540 \cdot 0.375 = 636.0 \text{ hPa}$

Probleme der Polynominterpolation Polynome mit hohem Grad oszillieren stark, besonders an den Rändern des Interpolationsintervalls. Für viele Stützpunkte ist Polynominterpolation daher ungeeignet. Lösung: Spline-Interpolation verwendet stückweise kubische Polynome mit glatten Übergängen.

Natürliche kubische Splinefunktion

Natürliche kubische Splinefunktion  $S(x)$  ist in jedem Intervall  $[x_i, x_{i+1}]$  durch kubisches Polynom dargestellt:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

mit den Randbedingungen  $S''_0(x_0) = 0$  und  $S''_{n-1}(x_n) = 0$ .

Natürliche kubische Splinefunktion berechnen

Parameter initialisieren:  $a_i = y_i$  und  $h_i = x_{i+1} - x_i$   
Randbedingungen setzen:  $c_0 = 0$  und  $c_n = 0$  (natürliche Spline).  
Gleichungssystem für  $c_i$  lösen Für  $i = 1, \dots, n - 1$ :

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_ic_{i+1} = 3(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}})$$

Restliche Koeffizienten:

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i), d_i = \frac{1}{3h_i}(c_{i+1} - c_i)$$

Kubische Splinefunktion Stützpunkte:

$x_i$	4	6	8	10
$y_i$	6	3	9	0

Parameter:  $a_0 = 6, a_1 = 3, a_2 = 9, h_0 = h_1 = h_2 = 2$   
Randbedingungen:  $c_0 = 0, c_3 = 0$  (natürliche Spline)  
Gleichungssystem: für  $c_1, c_2$ :

$$2 \cdot 8 \cdot c_1 + 2 \cdot c_2 = 3(3 - (-1.5)) = 13.5$$

$$2 \cdot c_1 + 2 \cdot 8 \cdot c_2 = 3((-4.5) - 3) = -22.5$$

Lösung:  $c_1 = 1.2, c_2 = -1.8$

Restliche Koeffizienten:

$$b_0 = -2.8, b_1 = 2.2, b_2 = -7.2, d_0 = 0.6, d_1 = -1.5, d_2 = 0.9$$

Die Splinefunktionen sind:

$$S_0(x) = 6 - 2.8(x - 4) + 0.6(x - 4)^3$$
  
$$S_1(x) = 3 + 2.2(x - 6) + 1.2(x - 6)^2 - 1.5(x - 6)^3$$
  
$$S_2(x) = 9 - 7.2(x - 8) - 1.8(x - 8)^2 + 0.9(x - 8)^3$$

Numerische Integration

Numerische Integration (Quadratur)

Für  $f : \mathbb{R} \rightarrow \mathbb{R}$  soll das bestimmte Integral  $I(f) = \int_a^b f(x)dx$  auf einem Intervall  $[a, b]$  numerisch berechnet werden.

Quadraturverfahren allgemeine Form: 
$$I(f) = \sum_{i=1}^n a_i f(x_i)$$

wobei  $x_i$  = Stützstellen oder Knoten und  $a_i$  = Gewichte

Newton-Cotes Formeln

Einfache Rechteck- und Trapezregel

Die Rechteckregel (Mittelpunktsregel) und die Trapezregel zur Approximation von  $\int_a^b f(x)dx$  sind definiert als:

Rechteckregel: 
$$Rf = f(\frac{a+b}{2}) \cdot (b-a)$$

Trapezregel: 
$$Tf = \frac{f(a) + f(b)}{2} \cdot (b-a)$$

Geometrische Interpretation

- Rechteckregel: Approximiert die Fläche durch ein Rechteck mit Höhe  $f(\frac{a+b}{2})$
- Trapezregel: Approximiert die Fläche durch ein Trapez zwischen  $(a, f(a))$  und  $(b, f(b))$

Summierte Rechteck- und Trapezregel  $f : [a, b] \rightarrow \mathbb{R}$  stetig

- $n \in \mathbb{N}$  = Anzahl Subintervalle
- Schrittweite  $h = \frac{b-a}{n}$
- Stützstellen  $x_i = a + ih$  für  $i = 0, 1, \dots, n$

Summierte Rechteckregel: 
$$Rf(h) = h \cdot \sum_{i=0}^{n-1} f(x_i + \frac{h}{2})$$

Summierte Trapezregel: 
$$Tf(h) = h \cdot (\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i))$$

Trapezregel für nicht-äquidistante Stützstellen

$$Tf_{neq} = \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2} \cdot (x_{i+1} - x_i)$$

Simpson-Regel approximiert  $f(x)$  durch Polynom 2. Grades an den Stellen  $x_1 = a, x_2 = \frac{a+b}{2}$  und  $x_3 = b$ .

Einfache Simpson-Regel: 
$$Sf = \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$$

Summierte Simpson-Regel:

$$Sf(h) = \frac{h}{3}(\frac{1}{2}f(a) + \sum_{i=1}^{n-1} f(x_i) + 2 \sum_{i=1}^n f(\frac{x_{i-1} + x_i}{2}) + \frac{1}{2}f(b))$$

Simpson-Regel als gewichtetes Mittel

Die summierte Simpson-Regel kann als gewichtetes Mittel der summierten Trapez- und Rechteckregel interpretiert werden:

$$Sf(h) = \frac{1}{3}(Tf(h) + 2Rf(h))$$

Fehlerabschätzung für summierte Quadraturformeln

Für genügend glatte Funktionen gelten folgende Fehlerabschätzungen:

Rechteckregel: 
$$\left| \int_a^b f(x)dx - Rf(h) \right| \leq \frac{h^2}{24}(b-a) \cdot \max_{x \in [a,b]} |f''(x)|$$

Trapezregel: 
$$\left| \int_a^b f(x)dx - Tf(h) \right| \leq \frac{h^2}{12}(b-a) \cdot \max_{x \in [a,b]} |f''(x)|$$

Simpson-Regel: 
$$\left| \int_a^b f(x)dx - Sf(h) \right| \leq \frac{h^4}{2880}(b-a) \cdot \max_{x \in [a,b]} |f^{(4)}(x)|$$

Schrittweite für gewünschte Genauigkeit bestimmen

Maximaler absoluter Fehler:  $\epsilon$

Höchste Ableitung abschätzen:

Berechne  $\max_{x \in [a,b]} |f^{(k)}(x)|$  für entsprechendes  $k$ .

Schrittweite berechnen:

Für Trapezregel: 
$$h \leq \sqrt{\frac{12\epsilon}{(b-a) \max |f''(x)|}}$$

Für Simpson-Regel: 
$$h \leq \sqrt[4]{\frac{2880\epsilon}{(b-a) \max |f^{(4)}(x)|}}$$

Anzahl Intervalle bestimmen:  $n = \frac{b-a}{h}$  (aufrunden auf ganze Zahl)

Anwendung Newton-Cotes Formeln

**Aufgabe:** Ein Teilchen mit Masse  $m = 10\text{ kg}$  bewegt sich durch eine Flüssigkeit mit Widerstand  $R(v) = -v\sqrt{v}$ . Für die Verlangsamung von  $v_0 = 20\text{ m/s}$  auf  $v = 5\text{ m/s}$  gilt:

$$t = \int_5^{20} \frac{m}{R(v)} dv = \int_5^{20} \frac{10}{-v\sqrt{v}} dv$$

Berechnen Sie das Integral mit  $n = 5$

**Parametrisation:**  $h = \frac{20-5}{5} = 3$ , Stützstellen: 5, 8, 11, 14, 17, 20

**Rechteckregel:**

$$Rf(3) = 3 \cdot \sum_{i=0}^4 f(x_i + 1.5)$$

Mittelpunkte: 6.5, 9.5, 12.5, 15.5, 18.5

$$Rf(3) = 3 \cdot (-0.154 - 0.108 - 0.090 - 0.081 - 0.076) = -1.527$$

**Trapezregel:**

$$Tf(3) = 3 \cdot \left( \frac{f(5) + f(20)}{2} + \sum_{i=1}^4 f(x_i) \right)$$

$$Tf(3) = 3 \cdot \left( \frac{-0.179 - 0.056}{2} + (-0.125 - 0.096 - 0.082 - 0.072) \right) = -1.477$$

**Simpson-Regel:**

$$Sf(3) = \frac{1}{3} (Tf(3) + 2Rf(3)) = \frac{1}{3} (-1.477 + 2(-1.527)) = -1.510$$

**Exakter Wert:**  $\int_5^{20} \frac{-10}{v^{3/2}} dv = \left[ \frac{20}{\sqrt{v}} \right]_5^{20} = -1.506$

**Absolute Fehler:**

- Rechteckregel:  $|-1.527 - (-1.506)| = 0.021$
- Trapezregel:  $|-1.477 - (-1.506)| = 0.029$
- Simpson-Regel:  $|-1.510 - (-1.506)| = 0.004$

**Schrittweite für gewünschte Genauigkeit**

**Aufgabe:** Bestimmen Sie die Schrittweite  $h$ , um  $I = \int_0^{0.5} e^{-x^2} dx$  mit der summierten Trapezregel auf einen absoluten Fehler von maximal  $10^{-5}$  genau zu berechnen.

**Parameter:**  $\epsilon = 10^{-5}$ ,  $a = 0$ ,  $b = 0.5$

**Zweite Ableitung bestimmen:** für  $f(x) = e^{-x^2}$

$$f'(x) = -2xe^{-x^2}$$

$$f''(x) = -2e^{-x^2} + 4x^2e^{-x^2} = e^{-x^2}(4x^2 - 2)$$

Auf  $[0, 0.5]$ :  $\max |f''(x)| = \max |e^{-x^2}(4x^2 - 2)| = 2$  (bei  $x = 0$ )

**Schrittweite berechnen:**

$$h \leq \sqrt{\frac{12 \cdot 10^{-5}}{0.5 \cdot 2}} = \sqrt{0.00012} \approx 0.011$$

**Anzahl Intervalle:**  $n = \frac{0.5}{0.011} \approx 46$  Intervalle

Romberg-Extrapolation

Idee der Romberg-Extrapolation

Die Romberg-Extrapolation verbessert systematisch die Genauigkeit der Trapezregel durch Verwendung mehrerer Schrittweiten und anschließende Extrapolation.

Basis: Trapezregel mit halbierten Schrittweiten  $h_j = \frac{b-a}{2^j}$  für  $j = 0, 1, 2, \dots, m$ .

Romberg-Extrapolation

Für die summierte Trapezregel  $Tf(h)$  gilt:

Sei  $T_{j0} = Tf(\frac{b-a}{2^j})$  für  $j = 0, 1, \dots, m$ . Dann sind durch die Rekursion

$$T_{jk} = \frac{4^k \cdot T_{j+1,k-1} - T_{j,k-1}}{4^k - 1}$$

für  $k = 1, 2, \dots, m$  und  $j = 0, 1, \dots, m-k$  Näherungen der Fehlerordnung  $2k + 2$  gegeben.

Die verwendete Schrittweitenfolge  $h_j = \frac{b-a}{2^j}$  heißt **Romberg-Folge**.

Romberg-Extrapolation durchführen

**Schritt 1:** Trapezwerte für erste Spalte berechnen

Berechne  $T_{j0}$  mit der summierten Trapezregel

für  $h_j = \frac{b-a}{2^j}$ ,  $j = 0, 1, \dots, m$ .

**Schritt 2:** Extrapolationsschema aufstellen

$T_{00}$			
$T_{10}$	$T_{01}$		
$T_{20}$	$T_{11}$	$T_{02}$	
$T_{30}$	$T_{21}$	$T_{12}$	$T_{03}$

**Schritt 3:** Rekursionsformel anwenden

$$T_{jk} = \frac{4^k \cdot T_{j+1,k-1} - T_{j,k-1}}{4^k - 1}$$

**Schritt 4:** Genaueste Näherung

Der Wert rechts unten im Schema ist die genaueste Approximation.

Romberg-Extrapolation anwenden

Berechne  $\int_0^\pi \cos(x^2) dx$  mit Romberg-Extrapolation für  $m = 4$  (d.h.  $j = 0, 1, 2, 3, 4$ ).

**Schritt 1:** Erste Spalte berechnen  $T_{00} = Tf(\pi)$  mit  $h_0 = \pi$  (1 Intervall)  $T_{10} = Tf(\pi/2)$  mit  $h_1 = \pi/2$  (2 Intervalle)  $T_{20} = Tf(\pi/4)$  mit  $h_2 = \pi/4$  (4 Intervalle)  $T_{30} = Tf(\pi/8)$  mit  $h_3 = \pi/8$  (8 Intervalle)  $T_{40} = Tf(\pi/16)$  mit  $h_4 = \pi/16$  (16 Intervalle)

**Beispielrechnung für  $T_{00}$ :**

$$T_{00} = \pi \cdot \frac{\cos(0) + \cos(\pi^2)}{2} = \frac{\pi}{2} (1 + \cos(\pi^2))$$

**Schritt 2:** Extrapolationsschema:

$T_{00}$				
$T_{10}$	$T_{01}$			
$T_{20}$	$T_{11}$	$T_{02}$		
$T_{30}$	$T_{21}$	$T_{12}$	$T_{03}$	
$T_{40}$	$T_{31}$	$T_{22}$	$T_{13}$	$T_{04}$

Der Wert  $T_{04}$  liefert die beste Approximation des Integrals.

Gauss-Formeln

Optimale Stützstellen

Bei Newton-Cotes Formeln sind die Stützstellen äquidistant gewählt. Gauss-Formeln wählen sowohl Stützstellen  $x_i$  als auch Gewichte  $a_i$  optimal, um die Fehlerordnung zu maximieren.

**Gauss-Formeln** für  $n = 1, 2, 3$

Die Gauss-Formeln für  $\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{i=1}^n a_i f(x_i)$  lauten:

$$n = 1: G_1 f = (b-a) \cdot f\left(\frac{b+a}{2}\right)$$

$$n = 2: G_2 f = \frac{b-a}{2} \left[ f\left(-\frac{1}{\sqrt{3}} \cdot \frac{b-a}{2} + \frac{b+a}{2}\right) + f\left(\frac{1}{\sqrt{3}} \cdot \frac{b-a}{2} + \frac{b+a}{2}\right) \right]$$

$$n = 3: G_3 f = \frac{b-a}{2} \left[ \frac{5}{9} f(x_1) + \frac{8}{9} f\left(\frac{b+a}{2}\right) + \frac{5}{9} f(x_3) \right]$$

wobei  $x_1 = -\sqrt{0.6} \cdot \frac{b-a}{2} + \frac{b+a}{2}$  und  $x_3 = \sqrt{0.6} \cdot \frac{b-a}{2} + \frac{b+a}{2}$ .

**Erdmasse berechnen** mit der nicht-äquidistanten Dichteverteilung:

$$m = \int_0^{6370} \rho(r) \cdot 4\pi r^2 dr$$

$r$ [km]	0	800	1200	1400	2000	...
$\rho$ [kg/m³]	13000	12900	12700	12000	11650	...

Stützstellen nicht äquidistant

→ speziell summierte Trapezregel für nicht-äquidistante Daten:

$$\int_0^{6370} \rho(r) \cdot 4\pi r^2 dr \approx \sum_{i=0}^{n-1} \frac{[\rho(r_i) \cdot 4\pi r_i^2] + [\rho(r_{i+1}) \cdot 4\pi r_{i+1}^2]}{2} \cdot (r_{i+1} - r_i)$$

**Wichtig:** Umrechnung der Einheiten:  $r$  in km  $\rightarrow$  m,  $\rho$  in kg/m³

Ergebnis:  $m_{Erde} \approx 5.94 \times 10^{24}\text{ kg}$

**Vergleich mit Literaturwert:**  $5.97 \times 10^{24}\text{ kg}$  Relativer Fehler:  $\approx 0.5\%$

Wahl des Integrationsverfahrens:

- Trapezregel:** Einfach, für glatte Funktionen ausreichend
- Simpson-Regel:** Höhere Genauigkeit (polynomähnliche Funktionen!)
- Romberg-Extrapolation:** Sehr hohe Genauigkeit (systematische Verbesserung)
- Gauss-Formeln:** Für begrenzte Anzahl von Funktionsauswertungen
- Nicht-äquidistante Daten:** Spezielle Trapezregel

Kombinierte Anwendung mit DGL

Bewegungsgleichung einer Rakete:  $a(t) = h''(t) = v_{rel} \cdot \frac{\mu}{m_A - \mu \cdot t} - g$

mit  $v_{rel} = 2600\text{ m/s}$ ,  $m_A = 300000\text{ kg}$ ,  $m_E = 80000\text{ kg}$ ,  $t_E = 190\text{ s}$ . Berechne Geschwindigkeit und Höhe als Funktion der Zeit.

**Parameter:**  $\mu = \frac{m_A - m_E}{t_E} = \frac{220000}{190} = 1158\text{ kg/s}$

**System 1. Ordnung:**  $z'_1 = z_2$  (Höhe)

$$z'_2 = 2600 \cdot \frac{1158}{300000 - 1158t} - 9.81$$
 (Geschwindigkeit)

**Anfangsbedingungen:**  $z_1(0) = 0$ ,  $z_2(0) = 0$

**Numerische Lösung mit Trapezregel:**

$$v(t) = \int_0^t a(\tau) d\tau \text{ und } h(t) = \int_0^t v(\tau) d\tau$$

**Ergebnisse nach 190s:** Geschwindigkeit:  $\approx 2500\text{ m/s}$ , Höhe:  $\approx 180\text{ km}$ , Beschleunigung:  $\approx 2.5g$



## Differentialgleichungen

**Differentialgleichung n-ter Ordnung** ist eine Gleichung, in der Ableitungen einer unbekannten Funktion  $y = y(x)$  bis zur  $n$ -ten Ordnung auftreten. Explizite Form:

$$y^{(n)}(x) = f(x, y(x), y'(x), \dots, y^{(n-1)}(x))$$

Gesucht sind die Lösungen  $y = y(x)$  dieser Gleichung, wobei die Lösungen  $y$  auf einem Intervall  $[a, b]$  definiert sein sollen.

Implizite Form: nicht nach  $y^{(n)}$  aufgelöst, sondern in Form  $F(x, y, y', y'', \dots, y^{(n)}) = 0$  gegeben.

### Arten von DGL

- **Separierbar:**  $y' = g(x) \cdot h(y)$   
→  $F(x, y)$  kann als Produkt eines  $x$ - &  $y$ -Anteils geschrieben werden
- **Autonom:**  $y' = f(y)$  →  $F(x, y)$  hängt nur von  $y$  ab
- **Linear:** falls die Variabel welche abgeleitet wird, nur in der ersten Potenz vorkommt und nicht multipliziert miteinander oder mit der unabhängigen Variabel wird.

### Homogenität von DGL

- Homogene DGL:  $F(x, y, y', y'', \dots, y^{(n)}) = 0$
- Inhomogene DGL:  $F(x, y, y', y'', \dots, y^{(n)}) = g(x)$  →  $g(x)$  ist die Störfunktion

### Allgemeine Lösung der inhomogenen DGL $y' + f(x)y = g(x)$

$$y = e^{-F(x)} \cdot \int g(x) e^{F(x)} dx$$

wobei  $F(x)$  eine Stammfunktion von  $f(x)$  ist.

**Anfangswertproblem (AWP)** für Differentialgleichung  $n$ -ter Ordnung werden der Lösungsfunktion  $y = y(x)$  noch  $n$  Werte vorgeschrieben:

#### DGL 1. Ordnung:

Gegeben ist  $y'(x) = f(x, y(x))$  und der Anfangswert  $y(x_0) = y_0$ .

#### DGL 2. Ordnung:

Gegeben ist  $y''(x) = f(x, y(x), y'(x))$  und

Anfangswerte  $y(x_0) = y_0, y'(x_0) = y'_0$ .

#### DGL n-ter Ordnung:

Gegeben ist  $y^{(n)}(x) = f(x, y(x), y'(x), \dots, y^{(n-1)}(x))$  und die Anfangswerte  $y(x_0) = y_0, y'(x_0) = y'_1, \dots, y^{(n-1)}(x_0) = y_{n-1}$ .

### Lösen von Separierbaren DGL $\frac{dy}{dx} = g(x) \cdot h(y)$

Für  $g(x), h(y)$  stetige Funktionen und  $\begin{cases} y' &= g(x)h(y) \\ y(x_0) &= y_0 \end{cases}$  mit  $h(y_0) \neq 0$  ist AWP:

**Trennung aller x- und y-Terme:**  $\frac{1}{h(y)} \cdot dy = g(x) \cdot dx$

**Integration beider Seiten, auflösen nach y:**  $\int \frac{1}{h(y)} dy = \int g(x) dx$

**Anfangsbedingungen einsetzen:**  $\int_{y_0}^y \frac{1}{h(s)} ds = \int_{x_0}^x g(t) dt$

**Spezialfall:**  $h(y_0) = 0$  →  $y = y_0$  eine Lösung der DGL.

**Radioaktiver Zerfall:**  $\frac{dn}{dt} = -\lambda n$  → DGL 1. Ordnung, Lösung:  $n(t) = n_0 e^{-\lambda t}$

**Freier Fall:**  $s''(t) = -g$  → DGL 2. Ordnung, Lösung:  $s(t) = -\frac{1}{2}gt^2 + v_0t + s_0$

**Harmonische Schwingung (Federpendel):**  $mx'' = -cx \Rightarrow x'' + \frac{c}{m}x = 0$

DGL 2. Ordnung, Lösung:  $x(t) = A \sin(\omega_0 t + \varphi)$  mit  $\omega_0 = \sqrt{\frac{c}{m}}$

## Richtungsfelder

**Richtungsfeld** = geometrisches Verständnis von expliziten

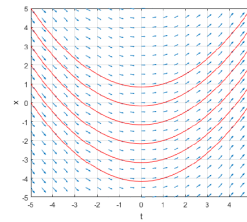
DGL 1. Ordnung, d.h. DGL der Form:  $y' = f(x, y)$

- $y'$  = Steigung der Lösungskurve am Punkt  $(x, y(x))$
- Richtungsfeld = Pfeil an jedem Punkt  $(x, y)$ , der die Steigung  $f(x, y)$  angibt
- Jeder Punkt ist somit die Tangente einer spezifischen Lösungskurve (verläuft tangential zu den Pfeilen)

### Richtungsfelder von Speziellen DGL

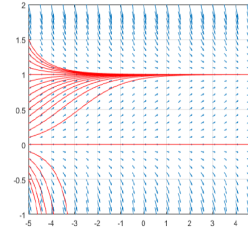
Unbestimmtes Integral:  $y' = f(x)$

unabhängig von  $y$ , Verschiebung in  $y$ -Richtung durch Konstante  $C$



Autonome DGL:  $y' = f(y)$

unabhängig von  $x$ , Verschiebung in  $x$ -Richtung durch  $C$



### Richtungsfeld zeichnen und interpretieren

**Steigungen  $f(x_i, y_j)$**  für verschiedene Punkte  $(x_i, y_j)$  berechnen

**Richtungspfeile zeichnen:**  $\forall (x_i, y_j)$ , Steigung  $f(x_i, y_j)$

**Lösungskurven:** Von Anfangspunkt  $(x_0, y_0)$  ausgehend folge Richtungspfeilen, um Lösungskurve zu approximieren.

**Python-Implementierung** Verwende `numpy.meshgrid()` und `pyplot.quiver()` zur automatischen Darstellung.

## Numerische Lösungsverfahren

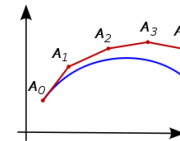
### Idee des Euler-Verfahrens

Euler-Verfahren folgt Tangente im Punkt  $(x_i, y_i)$

mit Steigung  $f(x_i, y_i)$  um Schrittweite  $h$ .

→ einfachstes Einschrittverfahren mit

Konvergenzordnung  $p = 1$ .



**Eulerverfahren** Findet Geraden mit Steigung  $f(x_i, y_i)$  für Punkte  $(x_i, y_i)$

**DGL am Punkt  $(x_i, y_i)$ :**  $y' = f(x_i, y_i) \rightarrow y = y_i + f(x_i, y_i) \cdot (x - x_i)$

**AWP:**  $y' = f(x, y)$  mit  $y(a) = y_0$  auf dem Intervall  $[a, b]$

**Euler-Verfahren** mit Schrittweite  $h = \frac{b-a}{n}$ :

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + h \cdot f(x_i, y_i)$$

Gegebene Anfangswerte:  $x_0 = a, x_i = a + ih$  für  $i = 0, \dots, n-1$  und  $y_0$  Problem: Steigung wird nur am linken Ende des Intervalls berücksichtigt!

**Anfangswerte berechnen** Für  $i = 0$  und  $x = x_0$ :

$$\underbrace{y_1}_{\approx y(x_1)} = y_0 + f(x_0, y_0) \cdot \underbrace{(x_1 - x_0)}_{=h} \rightarrow x_i = x_0 + i \cdot h$$

**Euler-Verfahren anwenden** AWP  $y' = f(x, y), y(a) = y_0$ , Intervall  $[a, b]$

**Parameter:**  $n$  = Anzahl Schritte, berechne:  $h = \frac{b-a}{n}$

**Startwerte:**  $x_0 = a, y_0$  = gegebener Anfangswert

**Iteration** für  $i = 0, 1, \dots, n-1$ :

- Berechne  $f(x_i, y_i)$
- Setze  $x_{i+1} = x_i + h$  und  $y_{i+1} = y_i + h \cdot f(x_i, y_i)$

**Absoluter Fehler:**  $|y(x_n) - y_n|$  ( $y(x_n)$  = exakte Lösung an Stelle  $x_n$ )

**Interpretation:** Punkte  $(x_i, y_i)$  approximieren Lösung  $y(x)$  an den Stützstellen

**Euler-Verfahren**  $\frac{dy}{dx} = \frac{x^2}{y}$  mit  $y(0) = 2$  auf Intervall  $[0, 1.4]$

**Parameter:**  $n = 2, h = 0.7$

**Iteration 0:**

- $i = 0: x_0 = 0, y_0 = 2 \rightarrow f(0, 2) = \frac{0^2}{2} = 0$
- $x_1 = 0 + 0.7 = 0.7$  und  $y_1 = 2 + 0.7 \cdot 0 = 2$

**Iteration 1:**

- $i = 1: x_1 = 0.7, y_1 = 2 \rightarrow f(0.7, 2) = \frac{0.7^2}{2} = 0.245$
- $x_2 = 0.7 + 0.7 = 1.4$  und  $y_2 = 2 + 0.7 \cdot 0.245 = 2.1715$

**Exakte Lösung:**  $y(x) = \sqrt{\frac{2x^3}{3}} + 4$   $y(1.4) = \sqrt{\frac{2 \cdot 1.4^3}{3}} + 4 = 2.253$

**Absoluter Fehler:**  $|2.253 - 2.1715| = 0.0815$

## Verbesserte Euler-Verfahren

**Mittelpunkt-Verfahren** berechnet Steigung in der Mitte des Intervalls:

$$x_{h/2} = x_i + \frac{h}{2} \text{ und } y_{h/2} = y_i + \frac{h}{2} \cdot f(x_i, y_i)$$

$$x_{i+1} = x_i + h \text{ und } y_{i+1} = y_i + h \cdot f(x_{h/2}, y_{h/2})$$

Konvergenzordnung:  $p = 2$

### Modifiziertes Euler-Verfahren (Heun-Verfahren)

verwendet den Durchschnitt zweier Steigungen:

$$k_1 = f(x_i, y_i) \text{ und } k_2 = f(x_i + h, y_i + h \cdot k_1)$$

$$x_{i+1} = x_i + h \text{ und } y_{i+1} = y_i + h \cdot \frac{k_1 + k_2}{2}$$

Konvergenzordnung:  $p = 2$

**Vergleich der Euler-Verfahren** AWP aus vorigem Bsp

**Mittelpunkt-Verfahren:**

- $x_{1/2} = 0.35, y_{1/2} = 2, f(0.35, 2) = 0.061$
- $y_1 = 2 + 0.7 \cdot 0.061 = 2.043$
- $x_{3/2} = 1.05, y_{3/2} = 2.128, f(1.05, 2.128) = 0.518$
- $y_2 = 2.043 + 0.7 \cdot 0.518 = 2.406$

**Modifiziertes Euler-Verfahren:**

- $k_1 = 0, k_2 = f(0.7, 2) = 0.245$
- $y_1 = 2 + 0.7 \cdot \frac{0 + 0.245}{2} = 2.086$
- $k_1 = 0.245, k_2 = f(1.4, 2.257) = 0.866$
- $y_2 = 2.086 + 0.7 \cdot \frac{0.245 + 0.866}{2} = 2.475$

**Fehlervergleich bei  $x = 1.4$ :**

- Exakt:  $y(1.4) = 2.253$
- Euler:  $|2.253 - 2.172| = 0.081$
- Mittelpunkt:  $|2.253 - 2.406| = 0.153$
- Modifiziert:  $|2.253 - 2.475| = 0.222$

**Runge-Kutta Verfahren** Das klassische Runge-Kutta Verfahren verwendet vier Steigungen und hat Konvergenzordnung  $p = 4$ :  
**Steigungen berechnen:**

$$k_1 = f(x_i, y_i), \quad k_2 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1)$$
$$k_3 = f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2), \quad k_4 = f(x_i + h, y_i + hk_3)$$

**Gewichtetes Mittel bilden:** Steigung =  $\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$   
**Nächsten Punkt berechnen:**  $x_{i+1} = x_i + h$  und  $y_{i+1} = y_i + h \cdot$  Steigung  
**Iteration fortsetzen:** Wiederhole bis zum Ende des Intervalls.

**Butcher-Schema** Charakterisiert Runge-Kutta Verfahren:

0			
$\frac{1}{2}$	$\frac{1}{2}$		
$\frac{1}{2}$	0	$\frac{1}{2}$	
1	0	0	1
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$

**Interpretation:** Die erste Spalte gibt die Stufen  $c_i$ , die zweite Spalte die Koeffizienten  $a_{ij}$  für die Steigungen  $k_j$  und die letzte Zeile die Gewichtung der Steigungen für die nächste Iteration an.

**Vergleich numerischer Verfahren** AWP:  $\frac{dy}{dx} = x + y^2, \quad y(0) = 0$  auf dem Intervall  $[0, 1]$  mit Schrittweite  $h = 0.2$ .

**Euler-Verfahren**

$f(x, y) = x + y^2, h = 0.2, n = 5$

**Iteration 0  $\rightarrow$  1:**  $x_0 = 0, y_0 = 0 \quad f(0, 0) = 0 + 0^2 = 0$   
 $x_1 = 0.2, y_1 = 0 + 0.2 \cdot 0 = 0$

**Iteration 1  $\rightarrow$  2:**  $x_1 = 0.2, y_1 = 0 \quad f(0.2, 0) = 0.2 + 0^2 = 0.2$   
 $x_2 = 0.4, y_2 = 0 + 0.2 \cdot 0.2 = 0.04$   
usw... **Euler-Ergebnis:**  $y(1) \approx 0.4150$

**Runge-Kutta Verfahren**

**Schritt 0  $\rightarrow$  1:**  $x_0 = 0, y_0 = 0$   
 $k_1 = f(0, 0) = 0, k_2 = f(0.1, 0) = 0.1$   
 $k_3 = f(0.1, 0.01) = 0.1 + 0.01^2 = 0.1001$   
 $k_4 = f(0.2, 0.02002) = 0.2 + 0.02002^2 = 0.2004$   
 $y_1 = 0 + \frac{0.2}{6}(0 + 2 \cdot 0.1 + 2 \cdot 0.1001 + 0.2004) = 0.0200$

**Schritt 1  $\rightarrow$  2:**  $x_1 = 0.2, y_1 = 0.0200$   
 $k_1 = f(0.2, 0.0200) = 0.2 + 0.0004 = 0.2004$   
 $k_2 = f(0.3, 0.0400) = 0.3 + 0.0016 = 0.3016$   
 $k_3 = f(0.3, 0.0502) = 0.3 + 0.0025 = 0.3025$   
 $k_4 = f(0.4, 0.0805) = 0.4 + 0.0065 = 0.4065$   
 $y_2 = 0.0200 + \frac{0.2}{6}(0.2004 + 2 \cdot 0.3016 + 2 \cdot 0.3025 + 0.4065) = 0.0801$   
usw...

**Runge-Kutta Ergebnisse:**  
 $y_1 = 0.0200, y_2 = 0.0801, y_3 = 0.1806, y_4 = 0.3214, y_5 = 0.5027$

**Runge-Kutta-Ergebnis:**  $y(1) \approx 0.5027$

**Genauigkeitsvergleich:**

Genau Lösung:  $y(1) \approx 0.5463$   
Fehler:

- Euler:  $|0.5463 - 0.4150| = 0.1313$
- Runge-Kutta:  $|0.5463 - 0.5027| = 0.0436$

Das Runge-Kutta Verfahren ist etwa 3-mal genauer.

**Konvergenzordnung:**

- Euler-Verfahren: Konvergenzordnung  $p = 1$
- Runge-Kutta 4: Konvergenzordnung  $p = 4$

Bei Halbierung der Schrittweite erwarten wir:

- Euler: Fehler halbiert sich
- RK4: Fehler wird um Faktor 16 kleiner

**DGL höherer Ordnung  $\rightarrow$  System 1. Ordnung**  
Jede DGL  $n$ -ter Ordnung kann in ein System von  $n$  DGL 1. Ordnung umgewandelt werden durch Einführung von Hilfsvariablen für die Ableitungen.

**DGL höherer Ordnung auf System 1. Ordnung zurückführen**  
**Nach höchster Ableitung auflösen:**  
Bringe die DGL in die Form  $y^{(n)} = f(x, y, y', \dots, y^{(n-1)})$ .

**Hilfsvariablen einführen**

$$z_1(x) = y(x)$$
$$z_2(x) = y'(x)$$
$$\dots$$
$$z_n(x) = y^{(n-1)}(x)$$

**System aufstellen**

$$z'_1 = z_2$$
$$\dots$$
$$z'_{n-1} = z_n$$
$$z'_n = f(x, z_1, z_2, \dots, z_n)$$

**Vektorielle Schreibweise:**  $z' = f(x, z)$  mit  $z(x_0) = \begin{pmatrix} y(x_0) \\ y'(x_0) \\ \vdots \\ y^{(n-1)}(x_0) \end{pmatrix}$

**Landende Boeing - DGL 2. Ordnung** Eine Boeing 737-200 landet mit  $v_0 = 100$  m/s und erfährt die Bremskraft  $F = -5x^2 - 570000$ . Die Bewegungsgleichung ist:  $mx'' = -5x'^2 - 570000$  mit  $m = 97000$  kg. Formen Sie in ein System 1. Ordnung um.

**Nach  $x''$  auflösen:**  $x'' = \frac{-5x'^2 - 570000}{97000}$

**Hilfsvariablen:**  
 $z_1(t) = x(t) =$  Position  
 $z_2(t) = x'(t) = v(t) =$  Geschwindigkeit

**System 1. Ordnung:**  $z'_1 = z_2$  und  $z'_2 = \frac{-5z_2^2 - 570000}{97000}$

**Anfangsbedingungen:**  $z(0) = \begin{pmatrix} 0 \\ 100 \end{pmatrix}$

Das System kann nun mit Runge-Kutta gelöst werden.

**Kombinierte Anwendung mit Ausgleichsrechnung**  
Umsatzentwicklung modelliert durch DGL:  $\frac{dU}{dt} = 0.1U(100 - U) - S(t)$   
 $U(t)$  = Umsatz in Millionen €,  $S(t) = 20 \sin(2\pi t)$  = saisonale Schwankungen

**Klassifikation:**

- Nichtlineare DGL 1. Ordnung (wegen  $U(100 - U)$  Term)
- Zeitabhängiger Störterm  $S(t)$

**Gleichgewichtspunkte für  $S = 0$ :**  $\frac{dU}{dt} = 0.1U(100 - U) = 0$   
Lösungen:  $U^* = 0$  oder  $U^* = 100$

Stabilität:  $\frac{d}{dU}(0.1U(100 - U)) = 0.1(100 - 2U)$

- Bei  $U^* = 0$ :  $\frac{df}{dU} = 10 > 0 \rightarrow$  instabil
- Bei  $U^* = 100$ :  $\frac{df}{dU} = -10 < 0 \rightarrow$  stabil

**Numerische Lösung:**  
Mit Runge-Kutta ( $h = 0.1$ ):  
System:  
 $\frac{dU}{dt} = 0.1U(100 - U) - 20 \sin(2\pi t)$   
 $\frac{dS}{dt} = 20 \cos(2\pi t)$

$t$	$U(t)$	$S(t)$	$\frac{dU}{dt}$
0	30.0	0	210.0
1	45.2	0	247.6
2	62.8	0	233.9
3	76.1	0	182.0
4	85.3	0	125.4
5	91.1	0	81.2

Typische Ergebnisse (Tabelle):

**Ausgleichsrechnung:** Fitten der numerischen Daten mit logistischer Funktion:  $U(t) = \frac{K}{1 + Ae^{-rt}}$   
Linearisierung durch:  $\ln(\frac{K-U}{U}) = \ln(A) - rt$   
Nach linearer Regression:  $K \approx 100, A \approx 2.33, r \approx 0.693$   
Güte des Fits:  $R^2 > 0.98$   
(sehr gute Anpassung an ungestörtes Wachstum)

Bei der numerischen Lösung von DGL kann es vorkommen, dass der numerische Fehler unbeschränkt wächst. Dies führt zu **instabilen** Lösungen. **Tipp:** beginne mit einer kleinen Schrittweite und prüfe die Stabilität der Lösung.

**Stabilitätsfunktion**  
Für DGL  $y' = -\alpha y$  (mit  $\alpha > 0$ ) kann die numerische Lösung in der Form  $y_{i+1} = g(h\alpha) \cdot y_i$  geschrieben werden.  
Die Funktion  $g(z)$  heißt **Stabilitätsfunktion** des Verfahrens.  
Das offene Intervall  $z \in (0, \alpha)$ , in dem  $|g(z)| < 1$  gilt, bezeichnet man als **Stabilitätsintervall**.

**Stabilität des Euler-Verfahrens**  $y' = -2.5y$  mit  $y(0) = 1$ .  
**Euler-Verfahren:**  $y_{i+1} = y_i - h \cdot 2.5y_i = y_i(1 - 2.5h)$   
**Stabilitätsfunktion:**  $g(z) = 1 - z$  mit  $z = 2.5h$   
**Stabilitätsbedingung:**  $|1 - 2.5h| < 1 \Rightarrow 0 < 2.5h < 2 \Rightarrow 0 < h < 0.8$   
**Verhalten:**

- $h = 0.2$ : Stabile Lösung (exponentieller Abfall)
- $h = 0.85$ : Instabile Lösung (Oszillation mit wachsender Amplitude)

**Exakte Lösung:**  $y(x) = e^{-2.5x}$  (streng monoton fallend)

**Lokaler und globaler Fehler**  
**Lokaler Fehler:** Fehler nach einer Iteration  $\varphi(x_i, h) := y(x_{i+1}) - y_{i+1}$   
**Globaler Fehler:** Der Fehler nach  $n$  Iterationen  $y(x_n) - y_n$

**Konsistenzordnung  $p$ :**  
Ein Verfahren hat Konsistenzordnung  $p$ , falls  $|\varphi(x_i, h)| \leq C \cdot h^{p+1}$

**Konvergenzordnung  $p$ :**  
Ein Verfahren hat Konvergenzordnung  $p$ , falls  $|y(x_n) - y_n| \leq C \cdot h^p$

**Konvergenzordnungen der Verfahren**

- Euler-Verfahren:** Konvergenzordnung  $p = 1$
- Mittelpunkt-Verfahren:** Konvergenzordnung  $p = 2$
- Modifiziertes Euler-Verfahren:** Konvergenzordnung  $p = 2$
- Klassisches Runge-Kutta:** Konvergenzordnung  $p = 4$

**Praktische Bedeutung:** Bei Halbierung der Schrittweite  $h$  reduziert sich der Fehler um den Faktor  $2^p$ .

Konvergenzverhalten für  $\frac{dy}{dx} = \frac{x^2}{y}$  mit  $y(0) = 2$  auf  $[0, 10]$

**Exakte Lösung:**  $y(x) = \sqrt{\frac{2x^3}{3} + 4}$   
**Fehler bei  $x = 10$  für verschiedene  $h$ :**

$h$	Euler	Mittelpunkt	Mod. Euler	Runge-Kutta
0.1	$10^{-1}$	$10^{-2}$	$10^{-2}$	$10^{-5}$
0.05	$5 \times 10^{-2}$	$2.5 \times 10^{-3}$	$2.5 \times 10^{-3}$	$6 \times 10^{-7}$
0.025	$2.5 \times 10^{-2}$	$6 \times 10^{-4}$	$6 \times 10^{-4}$	$4 \times 10^{-8}$

**Beobachtung:** Bei Halbierung von  $h$ :

- Euler: Fehler halbiert sich (Ordnung 1)
- Mittelpunkt/Mod. Euler: Fehler viertelt sich (Ordnung 2)
- Runge-Kutta: Fehler wird um Faktor 16 kleiner (Ordnung 4)

**Wann welches Verfahren?**

- Euler:** Einfachste Implementierung, Lernzwecke, grobe Näherungen
- Mittelpunkt/Modifiziert:** Bessere Genauigkeit als Euler, moderater Aufwand
- Runge-Kutta 4:** Standard für die meisten Probleme, gute Balance zwischen Genauigkeit und Aufwand

Mitternachtsformel 
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Polynomdivision ! Vorzeichen von Nullstellen umdrehen 
$$\frac{P(x)}{q(x)} = S(x) + \frac{r(x)}{q(x)} \quad P, q, S, r \text{ Polynome}$$

Funktionen

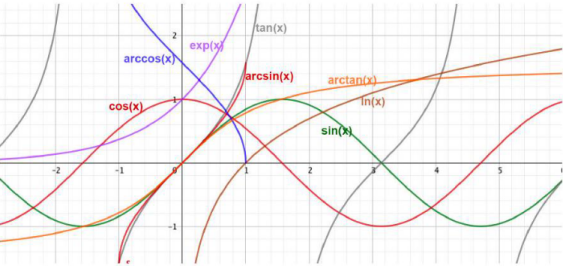
Komposition/Verkettung 
$$(g \circ f)(x) = g(f(x))$$

Stetigkeit Funktion ist stetig, falls Kurve keine Sprünge macht und man Graphen der Funktion zeichnen kann, ohne Stift abzusetzen.

Gleichmässige Konvergenz  $\forall x$  Linksseitiger Grenzwert = Rechtsseitiger GW

Symmetrie gerade  $f(-x) = f(x)$ , ungerade  $f(-x) = -f(x)$

Trigonometrische Funktionen



$$\tan x = \frac{\sin x}{\cos x} \text{ und } \cot x = \frac{\cos x}{\sin x}$$

Differentialrechnung

- Ableitungsregeln Seien  $f, g : D \rightarrow \mathbb{R}$  differenzierbar. Dann gelten:
- Summe/Differenz:  $(f + g)'(x) = f'(x) + g'(x)$
  - Faktor:  $f(x) = a \cdot g(x) \rightarrow f'(x) = a \cdot g'(x)$
  - Produkt:  $(f \cdot g)'(x) = f'(x)g(x) + f(x)g'(x)$
  - Quotient:  $(\frac{f}{g})'(x) = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$
  - Kettenregel:  $(g \circ f)'(x) = g'(f(x))f'(x)$
  - Umkehrfunktion:  $(f^{-1})'(y_0) = \frac{1}{f'(x)}$
  - Potenz/Logarithmus 1:  $(a^{f(x)})' = \ln(a) \cdot a^{f(x)} \cdot f'(x)$   
 $(f(x)^{g(x)})' = f(x)^{g(x)} \cdot (\ln(f(x)) \cdot g(x))' = f(x)^{g(x)} \cdot (\ln(f(x)) \cdot g(x) \cdot \frac{f'(x)}{f(x)})$

Sekanten-Steigung/Differentialquotient 
$$\frac{\Delta f}{\Delta x} = \frac{f(x_0+h) - f(x_0)}{h}$$

- Krümmung Zusammenhang zwischen 2. Ableitung und Krümmung:
- $f''(x_0) > 0$  Konvex (Nach links/oben gekrümmt)
  - $f''(x_0) < 0$  Konkav (Nach rechts/unten gekrümmt)
  - $f''(x_0) = 0$  Keine eindeutige Krümmung
- Bmk: Die Summe zweier konvexer Funktionen ist konvex. (konkav analog)

- Kurvendiskussion  $f(x) = y$
- $f'(x) = 0 \Rightarrow x$  lokales Extremum (NST)
  - Bestimme  $f''(x)$  (Zweite Ableitung)
    - $f''(x) = 0 \Rightarrow$  siehe Vorgehen Wende- und Sattelpunkte
    - $f''(x) < 0 \Rightarrow$  relatives Maximum
    - $f''(x) > 0 \Rightarrow$  relatives Minimum

- Vorgehen Wende- und Sattelpunkte:
- Wendepunkt:  $f^{(3)}(x_0) \neq 0$
  - Sattelpunkt: zusätzlich  $f'(x_0) = 0$
- Punkt bestimmen: In Gleichung  $f(x) = y$  einsetzen  $\rightarrow P(x, y)$

Integralrechnung

- Integralregeln  $(\lambda_1, \lambda_2 \in \mathbb{R})$
- Addition/Subtraktion:  $\int f(x - k)dx = F(x - k) + C$
  - Multiplikation:  $\int f(x \cdot k)dx = \frac{1}{k}F(x \cdot k) + C$
  - Skalarmultiplikation:  $\int \lambda_1 f(x) + \lambda_2 g(x)dx = \lambda_1 F(x) + \lambda_2 G(x) + C$
  - Linearkombination:  $\int (\lambda_1 f(x) + \lambda_2 g(x)) = \lambda_2 F(x) + \lambda_2 G(x) + C$
  - Verschiebung um k in x-Richt.:  $\int f(x - k)dx = F(x - k) + C$
  - Streckung um k in x-R.:  $\int f(k \cdot x)dx = \frac{1}{k}F(k \cdot x) + C(k \neq 0)$

Partielle Integration 
$$\int_a^b f(x)g'(x)dx = f(b)g(b) - f(a)g(a) - \int_a^b f'(x)g(x)dx$$
  
Unbestimmte:  $\int (f(x) \cdot g'(x))dx = f(x) \cdot g(x) - \int (f'(x) \cdot g(x))dx$

Substitution 
$$\int_{g(b)}^{g(a)} f(x)dx = \int_a^b f(g(t))g'(t)dt$$

Unbestimmte Integrale: 
$$\int f(g(t)) \cdot g'(t)dt = \int f(x)dx \Big|_{x=g(t)}$$

Nützliche Regeln Sei  $I \subseteq \mathbb{R}$  ein Intervall und  $f : I \rightarrow \mathbb{R}$  stetig.  
 $\int_{a+c}^{b+c} f(x)dx = \int_a^b f(t+c)dt$  und  $\int_a^b f(ct)dt = \frac{1}{c} \int_{ac}^{bc} f(x)dx$

- Substitution  $u = g(x), \frac{du}{dx} = g'(x), dx = \frac{du}{g'(x)}$
- Durchführen der Substitution  $u = g(x)$  und  $dx = \frac{du}{g'(x)}$ :  
 $\int f(x)dx = \int r(u)du$  bzw.  $\int_a^b f(x)dx = \int_{g(a)}^{g(b)} r(u)du$
  - Berechnen des Integrals mit Variable u:  
 $\int r(u)du = R(u) + C$  bzw.  $\int_{g(a)}^{g(b)} r(u)du = R(u) + C \Big|_{g(a)}^{g(b)}$
  - Rücksubstitution:  
 $R(u) + C = R(g(x)) + C$  bzw.  $R(u) + C \Big|_{g(a)}^{g(b)} = R(g(x)) + C \Big|_{g(a)}^{g(b)}$

- Uneigentliche Integrale
- $I = \int_a^\infty f(x)dx = \lim_{\lambda \rightarrow \infty} I(\lambda) = \lim_{\lambda \rightarrow \infty} (\int_a^\lambda f(x)dx)$
  - $I = \int_{-\infty}^b f(x)dx = \int_{-\infty}^c f(x)dx + \int_c^b f(x)dx = \lim_{\lambda \rightarrow -\infty} I(\lambda) +$
  - $I = \int_a^b f(x)dx = \lim_{\epsilon \rightarrow 0} I(\epsilon) = \lim_{\epsilon \rightarrow 0} (\int_{a+\epsilon}^b f(x)dx)$

Flächeninhalt zwischen zwei Kurven  $f(x)$  und  $g(x)$

$$\left| \int_a^{x_1} (f(x) - g(x))dx \right| + \left| \int_{x_1}^{x_2} (f(x) - g(x))dx \right| + \dots + \left| \int_{x_n}^b (f(x) - g(x))dx \right|$$

Mittelwert einer Funktion  $\mu = \frac{1}{b-a} \int_a^b f(x)dx$   
 $\rightarrow$  Höhe des Rechtecks mit  $l = b - a$  und Flächeninhalt  $A = \int_a^b f(x)dx$ .

- Wichtige Integrale
- Rotationsvolumen:  $V = \pi \int_a^b (f(x))^2 dx$
  - Bogenlänge:  $L = \int_a^b \sqrt{1 + (f'(x))^2} dx$
  - Mantelfläche:  $M = 2\pi \int_a^b f(x) \cdot \sqrt{1 + (f'(x))^2} dx$

- Trick Ungerade Funktionen  $\int_{-a}^+a f(x)dx = 0$
- Summe/Komposition: ungerade und ungerade  $\rightarrow$  ungerade
  - Produkt/Quotient: ungerade und gerade  $\rightarrow$  ungerade
  - Ableitung: gerade  $\rightarrow$  ungerade
- Ungerade:  $f(x) = -x, x, \sin(x), \tan(x)$ , Polynomfunkt. (ungerader Exponent)  
Gerade:  $1, x^2, \cos(x), \sec(x)$ , Polynomfunkt. mit geradem Exponent  
Gerade und Ungerade:  $f(x) = 0, f(x) = C$  (Konstante)

Ableitung   f'(x)	Funktion   f(x)	Integral   F(x)
0	C	x + C
1	x	$\frac{1}{2}x^2 + C$
$-\frac{1}{x^2}$	$\frac{1}{x}$	$\ln x  + C$
$ax^{a-1}$	$x^a$ with $a \in \mathbb{R}$	$\frac{x^{a+1}}{a+1} + C$
$x^x \cdot (1 + \ln x)$ $x > 0$	$x^x$	
$\frac{1}{2\sqrt{x}}$	$\sqrt{x}$	$\frac{2}{3}x^{\frac{3}{2}}$
$\frac{1}{n}x^{\frac{1}{n}-1}$	$\sqrt[n]{x}$	$\frac{n}{n+1}x^{\frac{1}{n}+1}$
$\ln(a) \cdot a^x$	$a^x$	$\frac{a^x}{\ln(a)} + C$
$a^{bx} \cdot c \ln a$	$a^{bx}$	$\frac{1}{b \ln a}a^{bx}$
$e^x$	$e^x$	$e^x + C$
$\frac{1}{x}$	$\ln(x)$	$x \ln(x) - x + C$
$\frac{1}{x \ln(a)}$	$\log_a(x)$	$x \log_a(x) - \frac{x}{\ln(a)} + C$
$\cos(x)$	$\sin(x)$	$-\cos(x) + C$
$-\sin(x)$	$\cos(x)$	$\sin(x) + C$
$1 + \tan^2(x) = \frac{1}{\cos^2(x)}$	$\tan(x)$	$-\ln \cos(x)  + C$
$-1 - \cot^2(x) = -\frac{1}{\sin^2(x)}$	$\cot(x)$	$\ln(\sin(x)) + C$
$\frac{1}{\sqrt{1-x^2}}$	$\arcsin(x)$	$x \arcsin(x) + \sqrt{1-x^2} + C$
$-\frac{1}{\sqrt{1-x^2}}$	$\arccos(x)$	$x \arccos(x) - \sqrt{1-x^2} + C$
$\frac{1}{1+x^2}$	$\arctan(x)$	$x \arctan(x) - \frac{1}{2} \ln(1+x^2) + C$
$\sin^2(x)$	$\frac{1}{2}(x - \sin(x) \cos(x))$	$\sin(x) \cos(x) + C$
$\cos^2(x)$	$\frac{1}{2}(x + \sin(x) \cos(x))$	$\cos(x) \sin(x) + C$
$\tan^2(x)$	$\tan(x) - x$	$\tan(x) + C$
$\cot^2(x)$	$-\cot(x) - x$	$\cot(x) + C$
$\frac{f'(x)}{f(x)}$	$\ln f(x) $	$x \cdot (\ln x  - 1) + C$
$\frac{-f'(x)}{(f(x))^2}$	$\frac{1}{f(x)}$	
$(ax + b)^n$	$\frac{1}{a \cdot (n+1)}(ax + b)^{n+1}$	

Approximations- und Rundungsfehler

- Fehlerarten Sei  $\tilde{x}$  eine Näherung des exakten Wertes  $x$ :
- Absoluter Fehler:  $|\tilde{x} - x|$
- Relativer Fehler:  $\left| \frac{\tilde{x} - x}{x} \right|$  bzw.  $\frac{|\tilde{x} - x|}{|x|}$  für  $x \neq 0$

Konditionierung Die Konditionszahl  $K$  beschreibt die relative Fehlervergrösserung bei Funktionsauswertungen:

$K := \frac{|f'(x)| \cdot |x|}{|f(x)|}$

- $K \leq 1$ : gut konditioniert
- $K > 1$ : schlecht konditioniert
- $K \gg 1$ : sehr schlecht konditioniert

Fehlerfortpflanzung Für  $f$  (differenzierbar) gilt näherungsweise:

Absoluter Fehler:  $|f(\tilde{x}) - f(x)| \approx |f'(x)| \cdot |\tilde{x} - x|$

Relativer Fehler:  $\frac{|f(\tilde{x}) - f(x)|}{|f(x)|} \approx K \cdot \frac{|\tilde{x} - x|}{|x|}$

Fehlerabschätzung für Nullstellen

- So schätzen Sie den Fehler einer Näherungslösung ab:
- Sei  $x_n$  der aktuelle Näherungswert
  - Wähle Toleranz  $\epsilon > 0$
  - Prüfe Vorzeichenwechsel:  $f(x_n - \epsilon) \cdot f(x_n + \epsilon) < 0$
  - Falls ja: Nullstelle liegt in  $(x_n - \epsilon, x_n + \epsilon)$
  - Damit gilt:  $|x_n - \xi| < \epsilon$

Vektoren Eigenschaften und Formeln

- Länge/Betrag:  $|\vec{a}| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$
- Einheitsvektor/Normierung:  $\vec{e}_a = \frac{1}{a} \cdot \vec{a} = \frac{\vec{a}}{|\vec{a}|}$
- Orthogonal (Senkrecht):  $\vec{a} \cdot \vec{b} = 0 \rightarrow$  orthogonal (90° Winkel)
- Vektoraddition:  $\begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$  und Skalarmultiplikation:  $\begin{pmatrix} \lambda \cdot a_x \\ \lambda \cdot a_y \end{pmatrix}$
- Skalarprodukt:  $\vec{a} \cdot \vec{b} = a_x \cdot b_x + a_y \cdot b_y = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\varphi)$
- Kreuzprodukt:  $\vec{a} \times \vec{b}$  ist orthogonal zu  $\vec{a}$  und  $\vec{b}$   
 $\vec{a} \times \vec{b} = \begin{pmatrix} ccc a_y \cdot b_z - a_z \cdot b_y \\ a_z \cdot b_x - a_x \cdot b_z \\ a_x \cdot b_y - a_y \cdot b_x \end{pmatrix}$

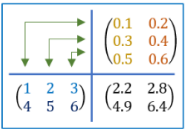
**Matrix** Tabelle mit  $m$  Zeilen und  $n$  Spalten:  $m \times n$ -Matrix  $A$   
 $a_{ij}$ : Element in der  $i$ -ten Zeile und  $j$ -ten Spalte

**Addition und Subtraktion**  $A + B = C \rightarrow c_{ij} = a_{ij} + b_{ij}$

**Skalarmultiplikation**  $k \cdot A = B \rightarrow b_{ij} = k \cdot a_{ij}$

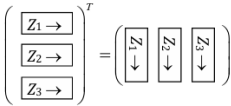
**Matrixmultiplikation**  $A^{m \times n}, B^{n \times k}$   
Bedingung:  $A$   $n$  Spalten,  $B$   $n$  Zeilen.  
Resultat:  $C$  hat  $m$  Zeilen und  $k$  Spalten.

- $A \cdot B = C$
- $c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$
- $A \cdot B \neq B \cdot A$



**Transponierte Matrix**  $A^{m \times n} \rightarrow (A^T)^{n \times m}$

- $A^T$ : Spalten und Zeilen vertauscht
- $(A^T)_{ij} = A_{ji}$  und  $(A \cdot B)^T = B^T \cdot A^T$



Spezielle Matrizen

- **Symmetrische Matrix:**  $A^T = A$
- **Einheitsmatrix:**  $E$  mit  $e_{ij} = 1$  für  $i = j$  und  $e_{ij} = 0$  für  $i \neq j$
- **Diagonalmatrix:**  $a_{ij} = 0$  für  $i \neq j$
- **Dreiecksmatrix:**  $a_{ij} = 0$  für  $i > j$  (obere Dreiecksmatrix) oder  $i < j$  (untere Dreiecksmatrix)

**Permutationsmatrix**  $P$  ist eine Matrix, die aus der Einheitsmatrix durch Zeilenvertauschungen entsteht.

Für die Vertauschung der  $i$ -ten und  $j$ -ten Zeile hat  $P_k$  die **Form**:

- $p_{ii} = p_{jj} = 0$
- $p_{ij} = p_{ji} = 1$
- Sonst gleich wie in  $E_n$

**Wichtige Eigenschaften:**

- $P^{-1} = P^T = P$
- Mehrere Vertauschungen:  $P = P_1 \cdot \dots \cdot P_1$

Quadratische Matrizen

Eigenschaften invertierbarer Matrizen

- $A \cdot A^{-1} = A^{-1} \cdot A = E$  und  $(A^{-1})^{-1} = A$
- $(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$  Die Reihenfolge ist relevant!
- $A$  und  $B$  invertierbar  $\Rightarrow AB$  invertierbar
- $(A^T)^{-1} = (A^{-1})^T$   $A$  invertierbar  $\Rightarrow A^T$  invertierbar

**Inverse einer 2 x 2-Matrix**  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  mit  $\det(A) = ad - bc$

$A^{-1} = \frac{1}{\det(A)} \cdot \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \rightarrow$  NUR Invertierbar falls  $ad - bc \neq 0$

**Inverse berechnen** einer quadratischen Matrix  $A^{n \times n}$

$A \cdot A^{-1} = E \rightarrow (A|E) \rightsquigarrow$  Zeilenoperationen  $\rightsquigarrow (E|A^{-1})$

Lineares Gleichungssystem (LGS)

Matrixform  $A \cdot \vec{x} = \vec{b}$ :  
 $\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \dots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$   $A$ : Koeffizientenmatrix  
 $\vec{x}$ : Vektor der Unbekannten  
 $\vec{b}$ : Vektor der Konstanten

**Rang einer Matrix**  $rg(A) =$  Anzahl Zeilen - Anzahl Nullzeilen  
 $\Rightarrow$  Anzahl linear unabhängiger Zeilen- oder Spaltenvektoren

Zeilenstufenform (Gaussii)

- Alle Nullen stehen unterhalb der Diagonalen, Nullzeilen zuunterst
- Die erste Zahl  $\neq 0$  in jeder Zeile ist eine führende Eins
- Führende Einsen, die weiter unten stehen  $\rightarrow$  stehen weiter rechts

**Zeilenoperationen** erlaubt bei LGS (z.B. Gauss-Verfahren)

- Vertauschen von Zeilen
- Multiplikation einer Zeile mit einem Skalar
- Addition eines Vielfachen einer Zeile zu einer anderen

Lösbarkeit von linearen Gleichungssystemen

- Lösbar:  $rg(A) = rg(A|b)$
- unendlich viele Lösungen:  $rg(A) < n$
- genau eine Lösung:  $rg(A) = n$

**Parameterdarstellung** bei unendlich vielen Lösungen

Führende Unbekannte: Spalte mit führender Eins  
Freie Unbekannte: Spalten ohne führende Eins

$\begin{matrix} x_1 & x_2 & x_3 & x_4 \\ \begin{pmatrix} 1 & -2 & 0 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix} \begin{vmatrix} 5 \\ 3 \end{vmatrix}$

**Homogenes LGS**  $\vec{b} = \vec{0} \rightarrow A \cdot \vec{x} = \vec{0} \rightarrow rg(A) = rg(A | \vec{b})$

- eine Lösung  $x_1 = x_2 = \dots = x_n = 0$ , die sog. *triviale Lösung*.
- unendlich viele Lösungen

Koeffizientenmatrix, Determinante, Lösbarkeit des LGS

Für  $n \times n$ -Matrix  $A$  sind folgende Aussagen äquivalent:

- $\det(A) \neq 0$
- $rg(A) = n$
- $A$  ist invertierbar
- Spalten von  $A$  sind linear unabhängig.
- Zeilen von  $A$  sind linear unabhängig.
- LGS  $A \cdot \vec{x} = \vec{0}$  hat eindeutige Lösung  $x = A^{-1} \cdot 0 = 0$

Fehleranalyse

Wichtige Normen

**1-Norm:**  $\|x\|_1 = \sum_{i=1}^n |x_i|, \|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|$

**2-Norm:**  $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}, \|A\|_2 = \sqrt{\rho(A^T A)}$

**$\infty$ -Norm:**  $\|x\|_\infty = \max_i |x_i|, \|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$

Fehlerabschätzung für LGS

Sei  $\|\cdot\|$  eine Norm,  $A \in \mathbb{R}^{n \times n}$  regulär und  $Ax = b, A\tilde{x} = \tilde{b}$

**Absoluter Fehler:**

$\|x - \tilde{x}\| \leq \|A^{-1}\| \cdot \|b - \tilde{b}\|$

**Relativer Fehler:**

$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \text{cond}(A) \cdot \frac{\|b - \tilde{b}\|}{\|b\|}$

Mit der Konditionszahl  $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$

Konditionierung

Die Konditionszahl beschreibt die numerische Stabilität eines LGS:

- $\text{cond}(A) \approx 1$ : gut konditioniert
- $\text{cond}(A) \gg 1$ : schlecht konditioniert
- $\text{cond}(A) \rightarrow \infty$ : singular

**Determinante**  $\det(A) \neq 0 \rightarrow A$  ist invertierbar

**Geometrische Interpretation** der Determinante:

Fläche im  $\mathbb{R}^2$  und Volumen im  $\mathbb{R}^3$  durch eine Matrix  $A$  aufgespannt:  $A = |\vec{a} \times \vec{b}| = |\det(A)|$

**Eigenschaften von Determinanten** mit  $A, B \in \mathbb{R}^{n \times n}, \lambda \in \mathbb{R}$

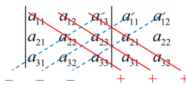
- $\det(A \cdot B) = \det(A) \cdot \det(B)$
- $\det(E) = 1$
- $\det(\lambda \cdot A) = \lambda^n \cdot \det(A)$
- $\det(A^T) = \det(A)$
- $\det(A) = 0 \Leftrightarrow A$  ist singular
- $\det(A^{-1}) = \frac{1}{\det(A)}$

$E$  = Einheitsmatrix

**Determinante 1 x 1-Matrix**  $\det(A) = A_{11}$

**Determinante 2 x 2-Matrix**  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \det(A) = |A| = a \cdot d - b \cdot c$

**Determinante 3 x 3-Matrix**  $A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$   
 $|A| = a \cdot e \cdot i + b \cdot f \cdot g + c \cdot d \cdot h - c \cdot e \cdot g - b \cdot d \cdot i - a \cdot f \cdot h$



Determinante n x n-Matrix

$\det(A) = |A| = \sum_{i=1}^n \text{oder } j=1 (-1)^{i+j} \cdot a_{ij} \cdot |A_{ij}|$

**Tipp:** Entwickeln nach Spalte oder Zeile mit den meisten Nullen!

Tricks und Tipps

- hat  $A$  eine Nullzeile oder -spalte, so ist  $\det(A) = 0$
- hat  $A$  zwei gleiche Zeilen oder Spalten, so ist  $\det(A) = 0$
- $\det(A) \neq 0 \Rightarrow$  Spalten/Zeilen sind linear unabhängig

**Determinante mit Gauss** Spezialfall **Dreiecks- /Diagonalmatrix**

Wende Gauss-Algorithmus an, um  $A$  in Dreiecksform zu bringen.

Es gilt für jede Dreiecksmatrix oder Diagonalmatrix  $D$ :

$\det(D) = (-1)^k \prod_{i=1}^n d_{ii}$

$k$  = Anzahl der Zeilen-Vertauschungen

Bei Skalarmultiplikationen ändert sich  $\det(A)$  um den Skalierungsfaktor

Matrix-Zerlegungen

**LR-Zerlegung**  $(E|A|E) \rightsquigarrow (P|R|L)$

Vorwärtseinsetzen:  $Ly = b$  Rückwärtseinsetzen:  $Rx = y$

**QR-Zerlegung** (noch genauer erklären)

- $A = Q \cdot R$  mit  $Q$  orthogonal und  $R$  obere Dreiecksmatrix
- $Q^T = Q^{-1}$
- $R$  ist eindeutig,  $Q$  nicht