Inhaltsverzeichnis

Zu allen Themengebieten sind die passenden Übungsaufgaben mit angegeben.

Weitere Zusatzthemen, zu denen ich aber nichts aufgeschrieben habe:

- **Kombinatorik (VL I.4, 90 min)**
  Eine gute Anwendung nachdem Mengen und Abbildungen eingeführt wurden. Produktmengen, Potenzmenge, Permutationen (mit Wiederholung), Binomialkoeffizienten (als Anzahl der k-elementigen Teilmengen einer Menge mit n Elementen).

- **Motivation: Differentialgleichungen (VL II.14, 90 min)**
  Ich wollte den Studenten eine weitere Motivation mitgeben, warum Analysis in MfI2 behandelt wird. Ein wichtiger Grund ist die Modellierung von Prozessen durch Differentialgleichungen in Naturwissenschaften, Ingenieurwissenschaften, Technik usw.

  Fadenpendel (klassische Mechanik); Lotka-Volterra-Gleichungen (Räuber-Beute-Modell, Biologie); Wellengleichung; Hitzegleichung.

  Prinzip: reales Problem (z.B. aus Technik oder Wirtschaft) →
  Modellierung mit Differentialgleichungen aus Physik/Chemie/usw. →
  Existenz von Lösungen bekannt (Mathematik), aber nicht explizit →
  numerische Näherungsverfahren →
  Softwareeinsatz

- **Fourier-Transformation (VL II.18,19, 120 min)**

Aufgabe 12.1. Skizzieren Sie die Sägezahnfunktion $f(x) = \frac{\pi - x}{\pi}$ für $x \in [0, 2\pi]$ und $f(x + 2\pi) = f(x)$ für alle $x \in \mathbb{R}$ und bestimmen Sie die Fourier-Reihe von f. Plotten Sie die ersten vier Näherungspolynome.

Aufgabe 12.2. Berechnen Sie die Fourier-Transformierte von $f(x) = \begin{cases} a - |x|, & |x| \leq a \\ 0, & |x| > a \end{cases}$ ($a \in \mathbb{R}_{>0}$) und skizzieren Sie $f(x)$ und $\hat{f}(\xi)$.

1 Abbildungen als Prozeduren (VL I.3, 15 min)

Einige Studenten haben Schwierigkeiten mit der formalen Definition von Abbildungen, vor allem damit, dass Definitions- und Wertebereich Bestandteil der Angabe einer Abbildung $f\colon A \to B$ sind.
Für Studenten mit Programmierhintergrund wird dieses Konzept verständlicher durch Beispiele aus der Praxis, wo Abbildungen durch `procedure` oder `function` spezifiziert werden. Hier für C-artige Syntax:

```
int Square(int n){
  return n*n;
}
```

entspricht der Abbildung
Square: int $\to$ int, $\quad n \mapsto n^2$
mit int $= \{-2^{15}, \ldots, 2^{15} - 1\}$

Die C-Funktion ist aber nicht wohldefiniert für $n \geq 2^8 = 256$ wegen $(2^8)^2 = 2^{16}$. Besser:

```
long Square2(int n){
  return n*n;
}
```

entspricht der Abbildung
Square: int $\to$ long, $\quad n \mapsto n^2$
mit long $= \{-2^{31}, \ldots, 2^{31} - 1\}$

[Die tatsächliche Bitlänge der Datentypen **int** und **long** kann variieren; es sind mindestens vorzeichenbehaftete Ganzzahlen mit 16 bzw. 32 Bits.]
Für weitere Änderungen der Quadratfunktion, z.B. für Fließkommazahlen, werden neue Funktionennamen benötigt — genau so, wie $\mathbb{R} \to \mathbb{R}, x \mapsto x^2$ eine andere Abbildung ist $\mathbb{Z} \to \mathbb{Z}, x \mapsto x^2$.
Hier ein Beispiel mit zwei Argumenten: die Funktionendeklaration **int** ggT(**int** n, **int** m) entspricht der Abbildung ggT: int $\times$ int $\to$ int, $\quad (n, m) \mapsto \mathrm{ggT}(n, m)$.

2 Graphen und Google PageRank (VL I.20, 90 min)

Letzte Vorlesung vor dem Jahreswechsel. Dann schon behandelt: Determinanten, charakteristisches Polynom, Eigenwerte und -vektoren, Diagonalisierbarkeit von Matrizen.

Graphen und Adjazentmatrizen

Didaktisches Konzept: Graphen sind wichtig in der Informatik; hier können Graphen schnell eingefürt werden und ihre Adjazenzmatrizen geben eine Verbindung zur linearen Algebra aus MfI1.

- Definition: Graph $G = (E(G), K(G))$ (Ecken- und Kantenmenge)
- Beispiele: konkrete Graphen, insbesondere Bäume; Verkehrs- und andere Netze
- Varianten: gerichtete Graphen; Graphen mit Schleifen und/oder Mehrfachkanten
- Definition: Adjazenzmatrix $A(G)$ eines Graphen G
- Beispiele und Varianten ($A(G)_{ii} = 2$ für eine ungerichtete Schleife an der Ecke i und $A(G)_{ij} = 1, A(G)_{ji} = 0$ für eine gerichtete Kante $i \to j$).
- Bemerkung: G ungerichteter Graph $\implies$ $A(G)$ symmetrische Matrix

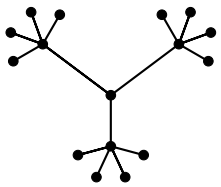
Netzwerke und Perron–Frobenius

Der (endliche, ungerichtete) Graph G modelliere Kontakte von Personen, d.h. die Eckenmenge $E(G)$ ist eine Menge von Menschen und Kanten in G bedeuten, dass sich die entsprechenden Menschen kennen.

- Welche Person ist am besten vernetzt?

Das ist nicht unbedingt die Person mit den meisten Kontakten (diese entspricht der Ecke maximalen Grades in G), denn wirklich wichtige Leute haben Kontakte zu anderen wichtigen Leuten, nicht notwendigerweise zu vielen Leuten.

Beispiel:



Dieser Graph hat drei Ecken mit Grad 5, aber die zentrale Ecke vom Grad 3 scheint wichtiger zu sein.

Idee: sei $I\colon E(G) \to \mathbb{R}_{\geq 0}$ ein Maß, wie wichtig jede Ecke ist. Wir postulieren

$$I(x) = \frac{1}{\theta} \sum_{\substack{y \in E(G), \\ \{x,y\} \in K(G)}} I(y)$$

für ein $\theta > 0$, d.h. die Wichtigkeit einer Ecke ist (bis auf einen Proportionalitsfaktor $1/\theta$) die Summe der Wichtigkeiten aller Nachbarn. Diese Formel übersetzt sich in

$$AI = \theta I,$$

d.h. I ist Eigenvektor von $A(G)$!

Satz von Perron–Frobenius: *Die Matrix $M \in M(n, \mathbb{R}_{\geq 0})$ sei* irreduzibel*, d.h. der gerichtete Graph G_M mit $i \to j \iff M_{ij} > 0$ ist gerichtet zusammenhängend. Dann gibt es genau ein $\theta_0 \in \mathbb{R}_{>0}$ mit folgenden Eigenschaften:*

1. *es gibt $v_0 \in \mathbb{R}^n_{>0}$ mit $Mv_0 = \theta_0 v_0$;*
2. *θ_0 hat algebraische (und damit auch geometrische) Vielfachheit 1;*
3. *für einen beliebigen Eigenwert $\theta \in \mathbb{C}$ von M gilt $|\theta| \leq \theta_0$;*
4. *ist $v \in \mathbb{R}^n_{\geq 0}$ Eigenvektor von M, dann gilt $Mv = \theta_0 v$.*

$\theta_0 = \theta_0(M)$ heißt Perron–Frobenius-Eigenwert *(oder auch* Spektralradius*) von M und v_0 ist (ein)* Frobenius-Eigenvektor*.*

Für einen zusammenhängenden Graphen G ist $A(G)$ irreduzibel, also hat nach Perron–Frobenius $AI = \theta I$ einen eindeutigen positiven Eigenwert θ und einen bis auf skalare Vielfache eindeutigen positiven Eigenvektor $I \in \mathbb{R}^{E(G)}_{>0}$. Der Frobenius-Eigenvektor I von $A(G)$ löst also die Ausgangsfrage und die Ecke(n) mit maximalem $I(x)$ ist am besten vernetzt.

Google PageRank

G sei gerichteter Graph mit n Ecken. Wir denken an den Internet-Graphen ($E(G)$ sind die Webseiten und $K(G)$ sind Links; $n \approx 10^9$). Wir nehmen an, dass jede Seite mindestens einen Link setzt, also Ausgangsgrad > 0 hat — andere Webseiten werden ignoriert.

$$
\begin{aligned}
A &:= A(G) \text{ gerichtete Adjazenzmatrix von } G;\\
D &:= \text{Diagonalmatrix der Ausgangsgrade von } G;\\
J &\in M(n, \{1\}) \text{ die } n \times n\text{-Matrix mit allen Einträgen 1;}\\
\alpha &\in (0,1) \text{ ein Parameter;}\\
M &:= \frac{1-\alpha}{n} J + \alpha D^{-1} A.
\end{aligned}
$$

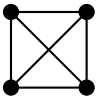
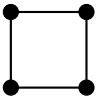
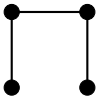
Wegen der Annahme ist D invertierbar. Außerdem haben $D^{-1}A$ und $\frac{1}{n}J$ beide Zeilensummen 1. Also sind alle Einträge von M positiv und M hat auch Zeilensummen 1. Damit ist M irreduzibel mit Perron–Frobenius-Eigenwert 1 und es existiert ein Frobenius-Eigenvektor $v = Mv$. Nach Normierung von v zu einer Wahrscheinlichkeitsverteilung, d.h. $\sum_x v_x = 1$ ist

$$PR(x) := v_X \in [0,1] \quad \text{PageRank der Seite } x \in E(G).$$

- Der Eigenvektor v wird iterativ durch $v^1 := (\frac{1}{n}, \ldots, \frac{1}{n})^t, v^2 := Mv^1, v^3 := Mv^2, \ldots$ approximiert. Die Anzahl der Iterationen zur Berechnung von v mit fixierter Fehlertoleranz wächst (nur!) logarithmisch mit n.
- α ist die Wahrscheinlichkeit, dass ein zufalliger Surfer einem Link auf der aktuellen Seite folgt (mit Wahrscheinlichkeit $1 - \alpha$ springt der Surfer zu einer beliebigen Webseite). In der Praxis $\alpha \approx 0,85$. Kleine Werte für α geben schnellere Konvergenz; größere Werte bilden G besser ab.
- Die Matrizen A und M sind sehr dünn besetzt: im Durchschnitt haben Webseiten 12 Links.
- Für die Praxis macht es keinen großen Unterschied, ob Links als gerichtete oder ungerichtete Kanten betrachtet werden.
- Der Vektor v gibt eine statische Rangliste aller Webseiten. Das ist nur ein Faktor von sehr vielen für SERP (search engine results page).

Andere, teilweise ältere, Anwendungen derselben Idee: bibliografische Referenzen (impact factor, 1977), Neuronen, soziale Netzwerke.

Aufgabe 12.4. Geben Sie Adjazenzmatrizen für die folgenden Graphen an:

G_1 :  G_2 :  G_3 : 

Berechnen Sie die Eigenwerte für ein $A(G_i)$ Ihrer Wahl.

Aufgabe 12.5. Geben Sie formale Definitionen der folgenden Begriffe:

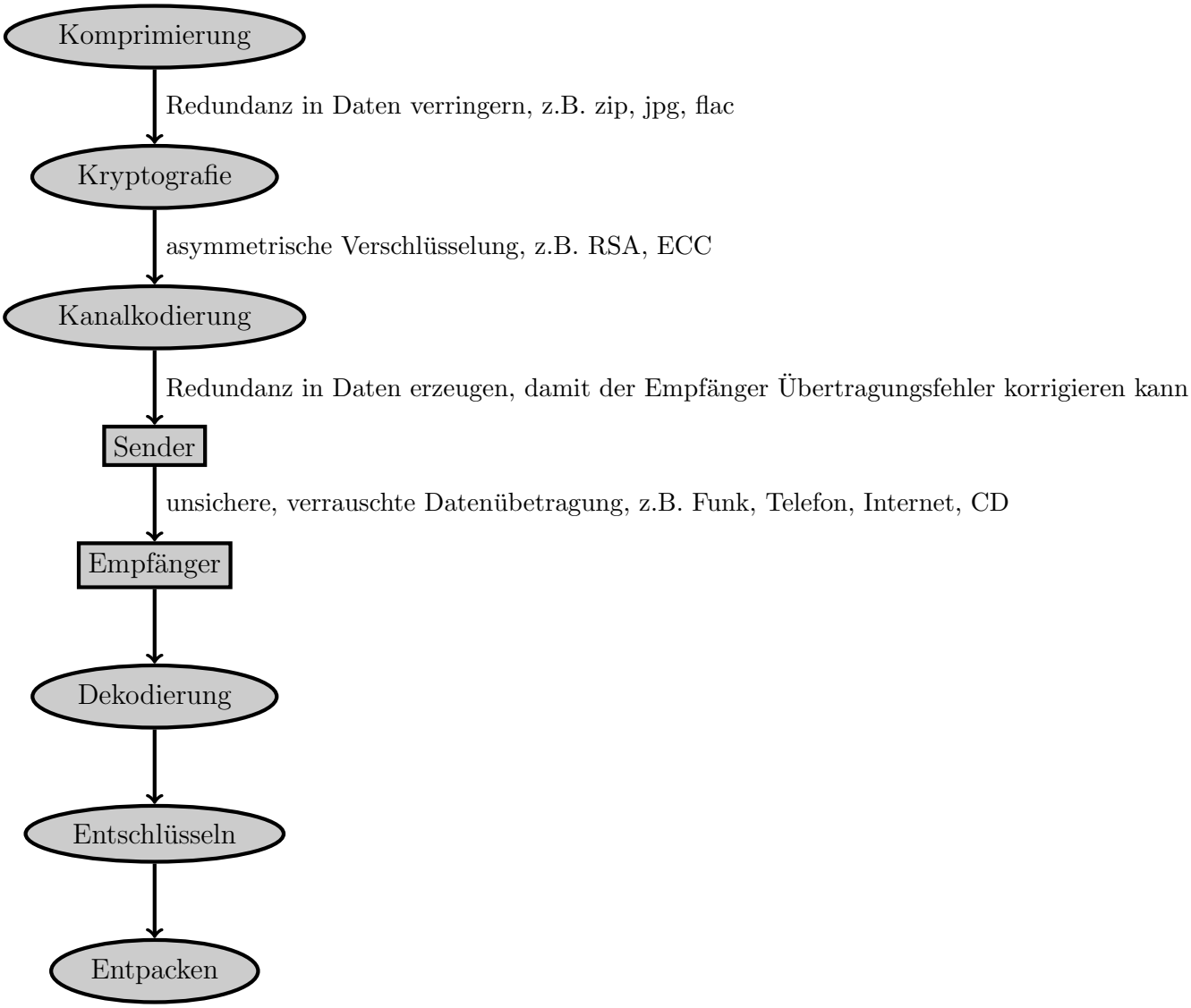
1. Gerichteter Graph.
2. (Ungerichteter) Graph mit Schleifen.
3. (Ungerichteter) Graph mit Schleifen und Mehrfachkanten.

3 Lineare Kodes (VL I.28, 90 min)

Letzte Vorlesung in MfI1.

Didaktisches Konzept: Lineare Kodes sind eine Anwendung von Matrizen und linearer Algebra, die praxisrelevant ist und Informatikstudenten mathematisch motivieren kann.

Als Überblick eine schematische Darstellung der Vorgänge bei Datenübertragungen. Die Studenten werden im Laufe des Studiums Einzelheiten zu allen drei Schritten (Kompression, Kryptographie, Kodierung) kennen lernen.



Hier Kanalkodierung, also Korrektur von Übertragungsfehlern: $I \xrightarrow{c} W \xrightarrow{t} W \xrightarrow{d} I$.

I	Information = Menge der zu übertragenden Daten
W	größere Menge, z.B. durch zusätzliche Bits
c	Kodierung = Einfügen von Redundanz = injektive Abbildung
t	Transfer = Datenübertragung über verrauschten Kanal
d	Dekodierung
C	Code, das Bild der Abbildung $c\colon I \to W$.

Beispiel: Übertragung eines einzelnen Bits: $I = \mathbb{F}_2 = \{0, 1\}$.
Redundanz durch dreifaches Versenden: $C = \{000, 111\} \subset W = \mathbb{F}_2^3$:

- nur die beiden Wörter 000 und 111 werden versendet,
- jedes Wort in $\mathbb{F}_2^3$ kann empfangen werden.

$\mathbb{F}_2^1 \xrightarrow{c} \mathbb{F}_2^3 \xrightarrow{t} \mathbb{F}_2^3 \xrightarrow{d} \mathbb{F}_2^1$, $\quad c\colon 0\mapsto 000,\ 1\mapsto 111$

$d\colon 000\mapsto 0,\ 001\mapsto 0,\ 010\mapsto 0,\ 011\mapsto 1,\ 100\mapsto 0,\ 101\mapsto 1,\ 110\mapsto 1,\ 111\mapsto 1$

Bei der Übertragung t ("binärer symmetrischer Kanal") wird ein Bit mit Wahrscheinlichkeit $p < \frac{1}{2}$ gestört. Die Dekodierung ist fehlerhaft $\iff$ es treten 2 oder 3 Übertragungsfehler auf, und die Wahrscheinlichkeit dafür ist $3p^2(1-p) + p^3 = 3p^2 - 2p^3$.
Für $p = 0,1 = 10\%$ ist $3p^2 - 2p^3 = 0,028 = 2,8\%$.

Ziele für gute Kodierung:

(1) schnelle Kodierung und Dekodierung
(2) erlaubt Korrektur möglichst vieler Übertragungsfehler, d.h. hohe Redundanz
(3) möglichst effizient, d.h. geringe Redundanz

Offensichtlich stehen (2) und (3) im Widerspruch. Die Konstruktion guter Kodes ist also ein nichttriviales Problem. Wegen (1) sucht man Kodes mit zusätzlicher Struktur, z.B. soll $C \subset W$ nicht irgendeine Teilmenge sein, sondern $C \subset \mathbb{F}_2^n$ ein Untervektorraum ("linearer Kode").

Satz von Shannon (1948): *Gegeben sei ein symmetrischer binärer Kanal mit Bitfehlerrate $p < \frac{1}{2}$ und Informationsrate $R < 1 + p\log_2(p) + (1-p)\log_2(1-p)$. Dann existiert für jedes $\epsilon > 0$ ein Kode mit Dekodierungsfehlerwahrscheinlichkeit $< \epsilon$.*

Nach diesem Satz gibt es also gute Kodes, die (2) und (3) vereinbaren. Der Beweis ist allerdings nicht konstruktiv und die benötigten Kodes können beliebig groß werden (müssen sogar beliebig groß werden: $\varepsilon \to 0 \implies C = \mathbb{F}_2^k \subset \mathbb{F}_2^M = W$ mit $k, M \to \infty$), wodurch das Finden praktischer Kodes eine interessante und schwere Aufgabe wird.

Typische, vereinfachende Annahmen:

- Alphabet $= \mathbb{F}_q$ endlicher Körper (oft $\mathbb{F}_2$)
- $C = \mathbb{F}_q^k$ Menge aller Wörter der Länge k
- alle übertragenen (Kode-)Wörter gleich lang, d.h. $W = \mathbb{F}_q^M$ (z.B. nicht Morse-Kode)

Definition: $w\colon \mathbb{F}_q^n \to \mathbb{N}, w(v) := |\{i \in \{1, \ldots, n\} : v_i \neq 0\}|$ *Hamming-Gewicht* von v.
$d\colon \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{N}, d(v, v') := w(v - v')$ *Hamming-Abstand* von v und v'.

- Beispiele
- d ist Metrik auf $\mathbb{F}_q^n$, so wie $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}, (v, v') \mapsto \sqrt{(v_1 - v_1')^2 + \cdots + (v_n - v_n')^2}$.
- $C \subseteq \mathbb{F}_q^n$ Kode, dann heißt $h(C) := \min_{\substack{v,v' \in C \\ v \neq v'}} d(v, v')$ *Minimalabstand* von C.

Bei $\geq h(C)$ Übertragungsfehlern kann ein Kodewort in ein anderes umgewandelt werden. Eine einfache Dekodierungsstrategie: $v \in \mathbb{F}_q^n$ empfangen $\implies$ dekodieren zu $c \in C$ mit $d(c, v)$ minimal.

Satz: *(i) $h(C) \geq s + 1 \implies C$ erkennt $\leq s$ Fehler.*
(ii) $h(C) \geq 2t + 1 \implies C$ erkennt $\leq t$ Fehler.

Beweis: $r \in \mathbb{N}$, $v \in \mathbb{F}_q^n$, $B_r(v) := \{w \in \mathbb{F}_q^n \mid d(v, w) \leq r\}$ Ball vom Radius r mit Mittelpunkt v. Falls $h(C) \geq 2t+1$ und $c, c' \in C$, dann ist wegen der Dreiecksungleichung $B_t(c) \cap B_t(c') = \varnothing$. □

- (n, M, d)-*Kode* $C \subseteq \mathbb{F}_q^n$: n Länge der Kodewörter, $M = |C|$ Anzahl der Kodewörter, $d := h(C)$ Minimalabstand von C.
- $C \subseteq \mathbb{F}_q^n$ ist ein *linearer Kode*, wenn C ein $\mathbb{F}_q$-Untervektorraum von $\mathbb{F}_q^n$ ist.
- $[n, k, d]$-*Kode* ist ein linearer Kode (n, q^k, d)-Kode $C \subseteq \mathbb{F}_q^n$, d.h. $k = \dim_{\mathbb{F}_q}(C)$.
- Ein (n, M, d)-Kode hat *Informationsrate* $R := \frac{1}{n} \cdot \log_q(M)$, also $R = \frac{k}{n}$ für C linear.
- $C \subseteq \mathbb{F}_q^n$ linear $\implies d(C) = \min\limits_{0 \neq c \in C} w(c)$.

Beispiele (alle mit $q = 2$):
- $C = \{000, 111\}$ $[3, 1, 3]$-Kode, $R = \frac{1}{3}$
- $C = \{000, 011, 101, 110\}$ $[3, 2, 2]$-Kode, $R = \frac{2}{3}$
- $C = \{00000, 01101, 10110, 11011\}$ $[5, 2, 3]$-Kode, $R = \frac{2}{5}$

$C \subset \mathbb{F}_q^n$ linearer Kode $\implies$ es gibt eine Basis $c_1, \ldots, c_k$ und die *Erzeugermatrix* von C ist $G \in M(k, n, \mathbb{F}_q)$, die zeilenweise die Basiskodewörter $c_1, \ldots, c_k$ enthält. Man kann immer annehmen, dass sie die Form $G = (I_k \mid A)$ mit $A \in M(k, n-k, \mathbb{F}_q)$ hat.

Beispiel: Der *Hamming-Kode* ist der binäre $[7, 4, 3]$-Kode mit $R = \frac{4}{7}$, der durch die Matrix G definiert wird. Er ist 1-fehlerkorrigierend.

$$G = \left(\begin{array}{cccc|ccc} 1 & & & & 1 & 1 & 0 \\ & 1 & & & 0 & 1 & 1 \\ & & 1 & & 1 & 1 & 1 \\ & & & 1 & 1 & 0 & 1 \end{array}\right)$$

Beispiel: Einer der *Hadamard-Kodes* ist der binäre $[8, 3, 4]$-Kode mit $R = \frac{3}{8}$, der durch die Matrix G definiert wird. Er ist 1-fehlerkorrigierend.

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

Beispiel: Für paarweise verschiedene $u_1, \ldots, i_n \in \mathbb{F}_q$ ist der *Reed–Solomon-Kode* (1960)

$$RS(q, k, n) := \{(f(u_1), \ldots, f(u_n)) \in \mathbb{F}_q^n \mid f \in \mathbb{F}_q[x]_{<k}\},$$

wobei $\mathbb{F}_q[x]_{<k}$ die Menge der Polynome mit $\mathbb{F}_q$-Koeffizienten vom Grad kleines als k ist. $RS(q, k, n)$ ist ein $[n, k, n-k+1]_q$-Kode (und optimal mit diesen Werten). Weiter $d(RS(q, k, n)) \geq n - k + 1$, denn ein Polynom $f \in \mathbb{F}_q[x]_{<k}$ hat $\leq k - 1$ Nullstellen. Anwendung der RS-Kodes:

⋆ 1977 Voyager-Sonden (Voyager 1 hat 2012 das Sonnensystem verlassen)

⋆ 1982 CD (Kratzerunterdrückung)

4 Algebra und RSA (VL II.1–4, 360 min)

In MfI1 alle Inhalte geschafft (bis Eigenwerte und Diagonalisierung), darum in MfI2 mit Algebra begonnen.

Didaktisches Konzept: Die Studenten können mit RSA ein einfaches und wichtiges asymmetrisches Kryptographieverfahren mathematisch verstehen und selbst durchführen.

Algebra = Strukturmathematik: es geht um Mengen mit Operationen wie + oder $\cdot$.
Eine der einfachsten Strukturen: Vektorräume (klassifiziert durch Dimension).

- Definition: Gruppe. Abelsche Gruppe.
- Beispiele mit üblicher Addition: $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{Z}/n, M(m,n,K)$ und K-Vektorräume.
- Beispiele mit üblicher Multiplikation: $K^* := K \setminus \{0\}$ für Körper K und die Matrixgruppen $GL_n(K), SL_n(K), O(n)$.

Beispiel: Automorphismengruppen. M sei eine mathematische Struktur, z.B. Menge, Vektorraum, Körper, Graph, Gruppe. Dann ist

$$\mathrm{Aut}(M) := \{f\colon M \to M \text{ Isomorphismus (=strukturerhaltende Bijektion)}$$

eine Gruppe, die *Automorphismengruppe* von M.

- $S_n = \{\text{Bijektionen } \{1,2,\dots,n\} \to \{1,2,\dots,n\}\}$ Permutationen
- $GL_n(K) = \{\text{Isomorphismen } K^n \to K^n \text{ als } K\text{-Vektorraum}\}$
- $O(n) = \{\text{Isomorphismen } \mathbb{R}^n \to \mathbb{R}^n \text{ als } \mathbb{R}\text{-Vektorraum mit Skalarprodukt } \langle -,- \rangle\}$
- $\mathrm{Aut}(G)$ für Graphen G

Permutationen

Es sind synonym:

- Bijektion $f\colon \{1,2,\dots,n\} \to \{1,2,\dots,n\}$
- Permutation (einer n-elementigen Menge)
- Element der symmetrischen Gruppe S_n

Wiederholung: f injektiv $\iff$ f surjektiv $\iff$ f bijektiv.

- Abbildungsschreibweise $\begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ f(1) & f(2) & f(3) & \cdots & f(n) \end{pmatrix}$
- k-Zyklus, Transposition
- Zykelschreibweise
- Fakt: jede Permutation ist Produkt von Transpositionen;
  es reichen sogar die n Transpositionen $(1\ 2), (2\ 3), \dots, (n-1\ n)$.

Beispiele für Umwandlung von Abbildungs- in Zykelschreibweise und umgekehrt; und für Verkettung und Inverse in beiden Schreibweisen.

Anwendungen von Permutationen in der Informatik sind unter anderem: Sortieralgorithmen; Graphentheorie; Kryptographie.

Ringe, Einheitengruppen

- Definition: Ring. Kommutativer Ring.
- Beispiele: $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{Z}/n$, Körper K, $M(n,K)$, Polynomring $K[x]$.
  $C(\mathbb{R}) := \{f\colon \mathbb{R} \to \mathbb{R} \text{ stetig}\}$ Ring der stetigen Funktionen auf $\mathbb{R}$.
- Einheit in einem Ring R. Einheitenmenge $R^* := \{r \in R \text{ Einheit}\}$. R^* ist Gruppe.
- Beispiele: $\mathbb{Z}^* = \{-1, 1\}, M(n,K)^* = GL_n(K), K[x]^* = K^*$.
  $(\mathbb{Z}/n)^* = \{m \in \mathbb{Z}/n\colon (m,n) = 1\}$ wegen Euklidischem Algorithmus.

Restklassenringe

Fixieren $n \in \mathbb{N}$ ("Modul"). Wiederholung:

- $a, b \in \mathbb{Z}$ kongruent modulo n
  $\iff n \mid b - a$
  $\iff a$ und b lassen gleichen Rest bei Teilung durch n
  $\iff a \equiv b \bmod n$
- Das ist eine Äquivalenzrelation (reflexiv, symmetrisch, transitiv) auf $\mathbb{Z}$.
- $\mathbb{Z}/n :=$ Menge der Äquivalenzklassen (=Restklassen modulo n)
- $\mathbb{Z}/n$ hat n Elemente, z.B. $\mathbb{Z}/n = \{0, 1, \ldots, n-1\}$, aber auch $\mathbb{Z}/6 = \{-2, -1, 0, 1, 2, 3\}$.
- $\mathbb{Z}/n$ ist Ring mit Addition und Multiplikation ganzer Zahlen.
- Euklidischer Algorithmus: $a, b \in \mathbb{Z}, d := \mathrm{ggT}(a, b) \implies \exists k, l \in \mathbb{Z} : ka + lb = d$.
- Anwendung: $(\mathbb{Z}/n)^* = \{m \in \mathbb{Z}/n \mid \mathrm{ggT}(m, n) = 1\}$.
- Korollar: $\mathbb{Z}/n$ Körper $\iff n$ Primzahl. (Schreiben $\mathbb{F}_p := \mathbb{Z}/p$.)

Definition: $\varphi\colon \mathbb{N} \to \mathbb{N}$ *Eulersche φ-Funktion*:
$$\varphi(n) := \#(\mathbb{Z}/n)^* = \#\{k \in \{1, \ldots, n-1\} \mid \mathrm{ggT}(k, n) = 1\}.$$

Beispiele: $\varphi(p) = p - 1, \varphi(4) = 2, \varphi(6) = 3$.

- Betrachten für $m, n \in \mathbb{N}$ die Abbildung $\gamma\colon \mathbb{Z}/mn \to \mathbb{Z}/m \times \mathbb{Z}/n$ mit $\gamma(k \bmod mn) = (k \bmod m, k \bmod n)$.
- Beispiel: $\mathbb{Z}/6 \xrightarrow{\cong} \mathbb{Z}/2 \times \mathbb{Z}/3$ bijektiv.
- Beispiel: $\mathbb{Z}/4 \to \mathbb{Z}/2 \times \mathbb{Z}/2$ weder injektiv noch surjektiv.
- γ ist wohldefiniert und ein Ringhomomorphismus.
- ***Chinesischer Restsatz:*** *$ggT(m, n) = 1 \iff \gamma$ ist Isomorphismus.*
  (mit Beweis und explizitem Beispiel)
- Korollar: $\mathrm{ggT}(m, n) = 1 \implies (\mathbb{Z}/mn)^* = (\mathbb{Z}/m)^* \times (\mathbb{Z}/n)^*$.
- Korollar: $\mathrm{ggT}(m, n) = 1 \implies \varphi(mn) = \varphi(m)\varphi(n)$.
- $\varphi(p^r) = (p^r - 1) - (p^{r-1} - 1) = (p-1)p^{r-1}$, denn $\mathrm{ggT}(a, p^r) = 1 \iff \mathrm{ggT}(a, p) = 1$.

Ordnung eines Gruppenelementes

Definition: $(G, \circ, e)$ Gruppe, $g \in G$ ein Element.
$$o(g) := \min\{l \geq 1 \mid g^l = e\} \in \mathbb{N} \cup \{\infty\} \text{ } \textit{Ordnung} \text{ von } g.$$

Beispiel: $G = (S_n, \circ, \mathrm{id})$ symmetrische Gruppe, $\pi \in S_n$ ein k-Zykel $\implies o(\pi) = k$.
Z.B. $\pi = (2\,3\,4\,5) \in S_5 \implies \pi^2 = (2\,4)(3\,5), \pi^3 = (2\,5\,4\,3) = \pi^{-1}, \pi^4 = \mathrm{id} \implies o(\pi) = 4$.

Beispiel: $G = GL_2(\mathbb{R})$ ($\circ$ = Hintereinanderausführung), g Spiegelung an einer Geraden durch den Ursprung $\implies o(g) = 2$.
Z.B. $g = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$

Beispiel: $G = (\mathbb{Z}/n, +, 0)$ additive Gruppe des Restklassenringes
$o(\overline{k}) = \min\{l \geq 1 \mid l\overline{k} = \overline{0} \in \mathbb{Z}/n\} = n/\mathrm{ggT}(n, k)$.
Z.B. $n = 12$: $o(\overline{3}) = 4, o(\overline{5}) = 12$.

Beispiel: $G = ((\mathbb{Z}/n)^*, \cdot, 1)$ multiplikative Einheitengruppe des Restklassenringes $\mathbb{Z}/n$.
Z.B. $(\mathbb{Z}/12)^* = \{\overline{1}, \overline{5}, \overline{7}, \overline{11}\}$ und $o(\overline{5}) = 2$, denn $\overline{5}^2 = \overline{25} = \overline{1}$.
Es gibt keine einfache Formel für $o(\overline{k})$, $\overline{k} \in (\mathbb{Z}/n)^*$.

Satz von Lagrange: *G endliche Gruppe, $g \in G \implies o(g) \mid \#G$.*

Beweis: $U := \{e, g, g^2, \ldots\}$ Untergruppe von G mit $\#U = o(g)$.
Definieren eine Relation auf G: $x \sim y \iff x = yg^k$ für ein $k \in \mathbb{Z} \iff xy^{-1} \in U$.
$\sim$ ist eine Äquivalenzrelation und die Nebenklassen sind xU. Wie für jede Äquivalenzrelation gilt: Klassen xU und yU sind entweder gleich oder disjunkt, und $G = \coprod_{x \in G} xU$.
Außerdem sind hier alle Klassen gleich groß, denn für $x, y \in G$ ist $yx^{-1} \cdot : xU \to yU, xg^k \mapsto (yx^{-1})xg^k = yg^k$ eine Bijektion. Also $\#xU = \#eU = \#U = o(g)$ und $\#G = \#(\coprod_{x \in G} xU) = a \cdot \#U = ao(g)$, wobei a die Anzahl der Nebenklassen ist. Somit $o(g) \mid \#G$. □

Beispiele:

- $G = S_4, \#S_4 = 4! = 24 \implies$ Ordnung einer Permutation in S_4 ist $1, 2, 3, 4, 6, 8, 12, 24$
- $\#G = p$ Primzahl $\implies$ alle Elemente haben Ordnung 1 ($e \in G$) oder Ordnung p.

Korollar (Euler): *$G = (\mathbb{Z}/n)^*$, $k \in \mathbb{Z} \implies o(\overline{k}) \mid \varphi(n) = \#G$, d.h. $\overline{k}^{\varphi(n)} \equiv 1 \bmod n$.*

Beispiele:

- $n = 15 = 3 \cdot 5$, $\varphi(15) = (3-1)(5-1) = 8$
  $\implies k^8 \equiv 1 \bmod 15 \ \forall k \in \mathbb{Z}$ mit $\mathrm{ggT}(15, k) = 1$.
  Z.B. $k = 2$: $2^8 = 256 = 17 \cdot 15 + 1 \equiv 1 \bmod 15$.
- Das erlaubt eine effiziente Berechnung von Potenzen modulo n: $2^{1005} \equiv 2^5 = 32 \equiv 2 \bmod 15$ wegen $1005 \equiv 5 \bmod 8 = \varphi(15)$.
- Ein Spezialfall ist der "kleine Satz von Fermat": p Primzahl, $k \in \mathbb{Z}$ mit $p \nmid k$, d.h. $\mathrm{ggT}(p, k) = 1$. Dann gilt $k^{p-1} \equiv 1 \bmod p$, d.h. $p \mid k^{p-1} - 1$.
  Z.B.: $p = 7, k = 3$: $7 \mid 3^6 - 1 = 728$.

Das RSA-Verfahren

Ziel: Übertragung vertraulicher Daten über einen unsicheren Kanal (der abgehört werden kann). Beispiele sind Passwörter für Webseiten oder Online-Einkäufe.

Lösung: öffentliche Schlüssel (public key cryptography)

- Schlüssel zum Kodieren sind öffentlich
- Schlüssel zum Dekodieren sind privat
- jeder Teilnehmer hat eigenes Schlüsselpaar
- es ist nur mit sehr hohem Aufwand möglich, den privaten Schlüssel aus dem öffentlichen Schlüssel zu bestimmen

Umsetzung:

1. Schlüsselerzeugung
   - wähle zwei große Primzahlen p, q
   - $n := pq$, so dass $\varphi(n) = (p-1)(q-1)$
   - wähle $e \in \mathbb{Z}$ mit $1 < e < \varphi(n)$ und $\mathrm{ggT}(e, \varphi(n)) = 1$
   - berechne $d \in \mathbb{Z}$ mit $de \equiv 1 \bmod \varphi(n)$ (Euklidischer Algorithmus)
   - *öffentlicher Schlüssel:* (n, e), d.h. n =Modul, e =Verschlüsselungsexponent
   - *privater Schlüssel:* d Entschlüsselungsexponent
     (Auch $p, q, \varphi(n)$ müssen privat bleiben, werden im Weiteren aber nicht gebraucht.)
2. Verschlüsselung
   - A möchte Nachricht an B senden
   - B stellt öffentlichen Schlüssel (n, e) bereit und hat privaten Schlüssel d
   - Nachricht sei eine Zahl $N \in \{0, 1, \ldots, n-1\}$
   - A sendet $c := N^e \bmod n$ über den unsicheren Kanal
3. Entschlüsselung
   - B empfängt $c \in \{0, 1, \ldots, n-1\}$
   - B berechnet $c^d \bmod n$
   - Es gilt: $c = (N^e)^d \equiv N \bmod n = pq$.

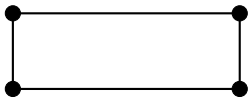
Beweis der letzten Aussage: $(N^e)^d = N^{ed} \equiv N \bmod n = pq \iff N^{ed} \equiv N \bmod p$ und $N^{ed} \equiv N \bmod q$ (chinesischer Restsatz). Fallunterscheidung:
I: $p \nmid N$, d.h. $\mathrm{ggT}(p, N) = 1$, dann $de \equiv 1 \mod \varphi(n) = (p-1)(q-1) \iff (p-1)(q-1) \mid de-1 \implies p-1 \mid de-1$ und $ed-1 = h(p-1)$, somit $N^{ed} = N \cdot N^{ed-1} = N \cdot (N^{(p-1)})^h \equiv N \cdot 1h = N \bmod p$ nach dem kleinen Satz von Fermat (das benutzt $\mathrm{ggT}(p, N) = 1$).
II: $p \mid N \iff N \equiv 0 \bmod p \implies N^{ed} = 0 \bmod p$. □

Bemerkungen:

- Worauf die Sicherheit von RSA beruht: Aus n und e kann man d nur bestimmen, indem man $n = pq$ faktorisiert und das ist sehr aufwändig für $p, q \approx 10^{100}$ (aktuelle Schlüsselgrößen).
- Die Asymmetrie des Verfahrens wird auch in anderen Algorithmen benutzt (diskrete Logarithmen; EEC = elliptic curve cryptography).
- Geschichte: 1978, Ron Rivest, Adi Shamir, Leonard Adleman (auch 1973 Clifford Cocks)

Aufgabe 1.1.

(a) Bestimmen Sie alle Automorphismen dieses Graphens:



(b) Geben Sie einen Graphen G mit mindestens zwei Ecken an, der nur den trivialen Automorphismus besitzt, also $\mathrm{Aut}(G) = \{\mathrm{id}_G\}$.

Aufgabe 1.2. Gegeben seien die Permutationen π_1, π_2, $\pi_3 \in S_5$ mit

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 2 & 1 & 5 & 4 \end{pmatrix}, \ \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 5 & 4 & 2 & 3 \end{pmatrix} \text{ und } \pi_3 = (2\,4\,3\,5).$$

(a) Stellen Sie π_1 und π_2 in Zyklenschreibweise und π_3 in Abbildungsschreibweise dar.

(b) Bilden Sie die Kompositionen $\pi_1 \circ \pi_2$, $\pi_2 \circ \pi_1$, $\pi_1 \circ \pi_3$, $(\pi_1 \circ \pi_3)^{-1}$, jeweils in Zyklen- und in Abbildungsschreibweise (also acht Rechnungen).

Aufgabe 1.3.

(a) Geben Sie $(\mathbb{Z}/10)^*$ an, also alle modulo 10 invertierbaren ganzen Zahlen zwischen 1 und 9. Stellen Sie die Gruppentafel für $(\mathbb{Z}/10)^*$ auf und vergleichen Sie die Ordnung der Gruppe mit $\varphi(10)$.

(b) Berechnen Sie $\varphi(720)$.

(c) Finden Sie alle $n \in \mathbb{N}$ mit $\varphi(n) = 16$.

Aufgabe 1.4. Finden Sie ein $x \in \mathbb{Z}$ mit $x \equiv 25 \bmod 48$ und $x \equiv 6 \bmod 13$.

Aufgabe 1.5. Eine Gruppe G heißt *zyklisch*, wenn sie von einem Element $g \in G$ erzeugt wird, d.h. $G = \{g^n \mid n \in \mathbb{Z}\}$. Welche der nachstehenden Einheitengruppen sind zyklisch?

$$(\mathbb{Z}/8)^*, \quad (\mathbb{Z}/10)^*, \quad (\mathbb{Z}/11)^*, \quad (\mathbb{Z}/14)^*.$$

Aufgabe 2.1. Berechnen Sie die Ordnungen folgender Gruppenelemente:

(a) $\left(\begin{smallmatrix}1&2&3&4&5&6&7&8&9\\4&2&6&8&9&1&3&7&5\end{smallmatrix}\right) \in S_9$

(b) $\overline{7} \in \mathbb{Z}/120$

(c) $\overline{7} \in (\mathbb{Z}/120)^*$

(d) die Drehmatrix $D_\alpha = \left(\begin{smallmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{R})$ für $\alpha = 30°$ und $\alpha = 17°$

$7, 49, 343 \equiv -17, -119 = 1$.

Aufgabe 2.2. Geben Sie alle Einheiten im Matrizenring $M(2, \mathbb{F}_2)$ an.

Aufgabe 2.3. Berechnen Sie die modulare Potenz $13^{77777} \in \mathbb{Z}/543$ ohne Hilfsmittel mit dem Satz von Euler.

Aufgabe 2.4. Wir wollen Ver- und Entschlüsselung im RSA-Algorithmus an einem kleinen Beispiel testen, mit $p = 61$, $q = 79$, $e = 127$.
Sie können hier und für Aufgabe 2.5 Hilfsmittel für die modulare Arithmetik benutzen, wie zum Beispiel `http://ptrow.com/perl/calculator.pl`.

(a) Berechnen Sie $n = pq$ und $\varphi(n)$ und den Entschlüsselungsexponenten d ($de \equiv 1 \bmod \varphi(n)$),

(b) Verschlüsseln Sie die Nachrichten $N = 2$ und $N = 222$.

(c) Entschlüsseln Sie die in (b) erhaltenen Codes.

Aufgabe 2.5. Führen Sie das RSA-Verfahren zusammen mit einem Kommilitonen durch.

(a) Wählen Sie Primzahlen $p, q > 1000$ und berechnen Sie n und $\varphi(n)$

(b) Wählen Sie einen Verschlüsselungsexponent $e > 100$ mit $\mathrm{ggT}(e, \varphi(n)) = 1$.

(b) Berechnen Sie den Entschlüsselungsexponenten d.

(d) Geben Sie den öffentlichen Schlüssel (n, e) an Ihren Kommilitonen.

(e) Lassen Sie sich eine verschlüsselte Nachricht schicken.

(f) Entschlüsseln Sie diese Nachricht und vergleichen Sie!

(x) Nehmen Sie den öffentlichen Schlüssel (n', e') Ihres Kommilitonen und eine kodierte Nachricht $c \in \mathbb{Z}/n$. Was müssten Sie tun, um c zu entschlüsseln?

5 Digitale Bildbearbeitung (VL II.26, 45 min)

Letzte Vorlesung in MfI2.

Drei Typen digitaler Bildbearbeitung:

- low level: Bild $\mapsto$ Bild (z.B. Farbfilter, Kompression wie jpg, Artefakte entfernen)
- mid level: Bild $\mapsto$ Bild oder Attribute (z.B. Morphologie, Segmentierung)
- high level: Regionen $\mapsto$ Attribute (Erkennung, Beschreibung, "computer vision")

Modellierung

- analoges, einfarbiges Bild: $f\colon [a,b]\times[c,d]\to\mathbb{R}$ (Grauwert)
- analoges, buntes Bild: $f\colon [a,b]\times[c,d]\to\mathbb{R}^3$ (Farben z.B. in RGB)

Gehen hier von einfarbigen Bildern aus. Die *Digitalisierung* (Übergang: stetig $\mapsto$ diskret) eines Bildes $f\colon [a,b]\times[c,d]\to\mathbb{R}$ erfolgt in Definitions- und Wertebereich:

- sampling: Digitalisierung der Koordinaten
- quantisation: Digitalisierung der Amplituden (Farbwerte)

und produziert als digitales Bild eine Matrix $f\in M(m,n,Z/2^l)$ mit $m\times n$ Pixeln in 2^l Graustufen. Typische Formate sind 640×320, 1024×768.

Spatial linear filters

Solche Filter operieren auf den Bildkoordinaten (also nicht im Frequenzbereich) mit den folgenden Eigenschaften:

- Es wird immer eine rechteckige Umgebung ("Fenster") modifiziert, z.B. 3×3 oder 5×5-Quadrate.
- Die Operation ist linear.

Typischerweise wird die Operation dann auf das ganze Bild angewendet, jeder Pixel kommt einmal als Mittelpunkt des Fensters vor.
Es sei W die 3×3-Matrix, die für die Operation benutzt wird ("Maske") und $W_{x,y}$ die $m\times n$-Matrix, die aus Nullen besteht, mit W zentriert an (x,y). Dann ist für $f\in M(m,n,\mathbb{Z}/2^l)$ die lokale Operation $f\mapsto W_{x,y}\cdot f$ (Matrixmultiplikation) und die globale Version ist $f\mapsto g:=\sum_{x,y} W_{x,y}\cdot f$.

Glättung (smoothing, blurring)

$$W=\frac{1}{9}\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}\qquad\text{oder}\qquad W=\frac{1}{16}\begin{pmatrix}1&2&1\\2&4&2\\1&2&1\end{pmatrix}$$

beschreiben lokale Mittelungen der Farbwerte; die Gesamtintensität bleibt erhalten.
Dieser Vorgang entspricht einer Integration (Summation). Wird er wiederholt angewendet, wird das Bild einfarbig mit der Durchschnittsintensität.

Schärfen (sharpening, deblurring)

Das ist eine Umkehroperation zum Glätten, bei der Kanten betont werden. Mathematisch findet eine (diskrete) Ableitung statt und eine Variante benutzt den Laplace-Operator.

- Für eine Funktion $g(x)$ in einer Variablen haben wir die Näherungen $g'(x)\approx\frac{1}{h}(f(x+h)-f(x)$ oder auch $g'(x)\approx\frac{1}{h}(f(x+h/2)-f(x-h/2)$ und für die zweite Ableitung $g''(x)\approx\frac{1}{h^2}(g(x+h)-2g(x)+g(x-h))$.
- Für $f\in C^2(\mathbb{R}^2)$ ist $\Delta f(x,y):=f_{xx}+f_{yy}=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}$ der Laplace-Operator.
- Wenn wir f_{xx} und f_{yy} einzeln wie eben approximieren, dann ist $f_{xx}\approx\frac{1}{h^2}(f(x+h,y)-2f(x,y)+f(x-h,y))$ und $f_{yy}\approx\frac{1}{h^2}(f(x,y+h)-2f(x,y)+f(x,y-h))$.

- Für Bildbearbeitung ist die kleinste Schrittweite $h = 1$ Pixel: $f \in M(m, n, \mathbb{Z}/2^l)$
  $\Delta f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$

Das entspricht der Maske W (die Maske W' wird auch benutzt und ist eine andere Näherung des reellen Laplace-Operators):

$$W = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \text{oder} \qquad W' = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Dieser Filter wird benutzt als $f \mapsto f - \Delta f$.

Beispiel: Eine sigmoide Funktion $f : \mathbb{R} \to \mathbb{R}$ gibt einen Farbübergang in einer Dimension an, etwa von weiß zu schwarz. Wenn dieser Übergang steil genug ist, wird der Effekt bei $f - f''$ noch verstärkt, ohne die rein weißen/schwarzen Bereiche zu verändern: