$\mathcal{M}\alpha\mathrm{T}\mathbb{Z}\varepsilon$ **frühstückt**

Revision 31 (14.01.2013)

Steffen Ohrendorf et al.

http://earvillage.square7.ch

Als Informatiker sollte man immer Linux zu Hause haben. Falls mal Besuch kommt.

`http://german-bash.org/83851`

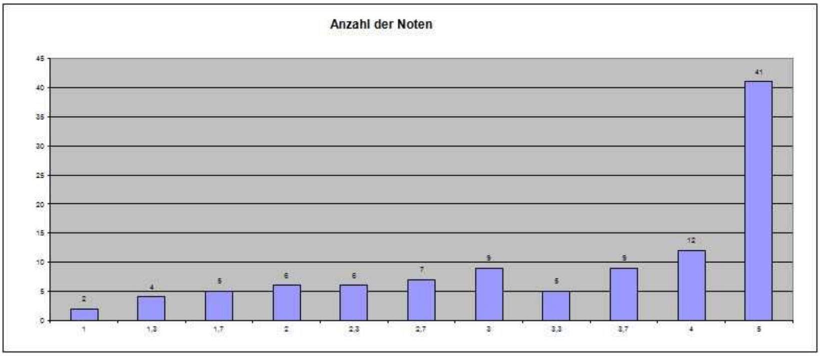
Inhaltsverzeichnis

Vorwort

Wichtiger Hinweis

Dieses Dokument ist *nicht* als Skript zu verstehen und sollte nicht als alleinige Basis zum Lernen dienen. Es ist als begleitendes Material gedacht, um eventuell das vorhandene Wissen aufzufrischen.
Wer beim Durchlesen oder Nachschlagen ein Thema nicht voll versteht, sollte *auf jeden Fall* noch einmal einen Blick in das Skript der entsprechenden Vorlesung werfen.

Es ist durchaus ratsam, das Studium ernst zu nehmen und in den Vorlesungen nicht zu schlafen, denn sonst passiert sowas hier:



(Klausur Analysis I, WS 2011/12)

Danksagung

Ich möchte folgenden Personen danken:

- Jens Hollmann, Dr. Alexander Voß, Hans Joachim Pflug und Benno Willemsen für die Vorlesungen, Präsentationen, Notizen und Inspirationen.
- Tim Böhmer für den Teil „Kombinatorik“ auf Seite 10.
- Frederic Klein für einen wichtigen Teil von „Lineare Gleichungssysteme“ auf Seite 26.
- Mike Boddin für das Korrekturlesen von Revision 21.
- Einem Spender, der mich dazu angespornt hat, den Teil „Lineare Algebra“ noch einmal zu überarbeiten und zu vervollständigen.

Eine Bitte

Bitte nehmen Sie sich die Zeit, mir über `http://earvillage.square7.ch` eine kurze Nachricht zukommen zu lassen. Ich weiß selbst, wie gern man etwas benutzt, ohne dem Schöpfer/Ersteller/Autor/etc. eine Nachricht zukommen zu lassen, wie seine Arbeit denn nun ankommt.

Ich habe bereits einiges an Arbeit in dieses PDF investiert, und aufgrund der vermutlich geringen Nutzerbasis ist es schwer, überhaupt an Rückmeldungen zu gelangen. Mir geht es nicht darum, dass Sie mich loben – lediglich eine Nachricht, dass Sie meine Arbeit nutzen, lässt mich weitermachen. Denn wozu sollte ich mir die Mühe machen, wenn sie ins Leere verpufft?

Außerdem bitte ich Sie, nicht zu zögern, mir Fehler und Unvollständigkeiten mitzuteilen. Auf diese Weise weiß ich, dass mein PDF genutzt wird, und Sie helfen auch den anderen Lesern Fehler zu vermeiden.

Teil I.

Mathematische Grundlagen

1. Mengenlehre

1.1. Definitionen

> Eine Menge ist eine Zusammenfassung von bestimmten wohl unterschiedenen Objekten unserer Anschauung oder unseres Denkens – welche die Elemente der Menge genannt werden – zu einem Ganzen.
>
> (Cantor)

Explizite Aufzählung $M = \{a, b, c, d, \ldots\}$

Aufzählung nach Eigenschaft $P = \{x \mid x \text{ ist Primzahl}\}$

Leere Menge $\varnothing$ oder $\{\}$. Achtung: $\{\varnothing\} = \{\{\}\}$ ist eine Menge, die eine leere Menge enthält. Es gilt $\forall M : \varnothing \in M$.

(Echte) Teilmenge A ist (echte) Teilmenge von B: $A \subset B := A \neq B \wedge \forall a \in A : a \in B$.

Unechte Teilmenge A ist (unechte) Teilmenge von B: $A \subseteq B := \forall a \in A : a \in B$.

Durchschnitt (Schnittmenge) $A \cap B := \{x \mid x \in A \wedge x \in B\}$

Vereinigung $A \cup B := \{x \mid x \in A \vee x \in B\}$

Differenzmenge $A \setminus B := \{x \mid x \in A \wedge x \notin B\}$

1.2. Rechenregeln

Prioritäten, absteigend sortiert: Klammern, Komplement, Durchschnitt, Vereinigung.

Kommutativgesetz $A \cap B = B \cap A$, $A \cup B = B \cup A$

Assoziativgesetz $(A \cap B) \cap C = A \cap (B \cap C)$, $(A \cup B) \cup C = A \cup (B \cup C)$

Distributivgesetz $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Transitivgesetz $A \subset B \wedge B \subset C \Rightarrow A \subset C$

Komplement $\overline{\mathbb{R}} = \varnothing$, $\overline{\varnothing} = \mathbb{R}$, $\overline{\overline{A}} = A$

Potenzmenge Menge aller Teilmengen einer Menge: $P(A)$. Es ist $|P(A)| = 2^{|A|}$.

Kreuzprodukt $A \times B := \{(a, b) \mid a \in A \wedge b \in B\}$

1.2.1. De Morgansche Regeln

$$
\begin{aligned}
A \setminus (B \cap C) &= (A \setminus B) \cup (A \setminus C) \\
A \setminus (B \cup C) &= (A \setminus B) \cap (A \setminus C) \\
\overline{(A \cup B)} &= \overline{A} \cap \overline{B} \\
\overline{(A \cap B)} &= \overline{A} \cup \overline{B}
\end{aligned}
$$

1.3. Relationen

Eine Relation R auf einer Menge A ist eine (unechte) Teilmenge von $A \times A$.

Reflexivität $\forall a \in A : (a, a) \in R$

Symmetrie $\forall a, b \in A : (a, b) \in R \Leftrightarrow (b, a) \in R$

Transitivität $\forall a, b, c \in A : (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$

Äquivalenzrelation Eine Äquivalenzrelation $\sim$ auf einer Menge M muss reflexiv, symmetrisch und transitiv (rst) sein. Es gilt dann: $(a, b) \in R \Leftrightarrow a \sim b$.

1.4. Gruppe

Eine nicht-leere Menge M, auf der eine Verknüpfung $\circ$ definiert ist, heißt Gruppe, wenn für $\forall x, y, z \in M$ gilt:

Abgeschlossenheit $x \circ y \in M$

Assoziativität $(x \circ y) \circ z = x \circ (y \circ z)$

Neutralelement Es existiert *genau ein* Element $n \in M$, für das gilt: $x \circ n = n \circ x = x$

Inverses Element Zu jedem $x \in M$ existiert ein $x^{-1} \in M$, für das gilt[1]:
$x \circ x^{-1} = x^{-1} \circ x = n$

Mit folgender zusätzlicher Bedingung heißt die Gruppe kommutativ bzw. Abelsch:

Kommutativ $x \circ y = y \circ x$

1.5. Ring

Eine nicht-leere Menge M heißt Ring, wenn auf ihr zwei Verknüpfungen $\oplus$ und $\odot$ mit folgenden Eigenschaften definiert sind:

1. M ist eine kommutative Gruppe bezüglich $\oplus$.

2. Ohne das Neutralelement bzgl. $\oplus$ erfüllt der Rest von M alle Axiome einer Gruppe bezüglich $\odot$, jedoch *nicht* das Axiom des inversen Elements.

3. Es gelten die Distributivgesetze ($\forall x, y, z \in M$):

$$\begin{aligned} (x \oplus y) \odot z &= (x \odot z) \oplus (y \odot z) \\ x \odot (y \oplus z) &= (x \odot y) \oplus (x \odot z) \end{aligned}$$

Anmerkung: Das neutrale Element bezüglich der „Addition“ $\oplus$ wird oft mit 0 bezeichnet, das neutrale Element bezüglich der „Multiplikation“ $\odot$ oft als 1.

1.6. Körper

Eine nicht-leere Menge M heißt Körper, wenn sie die Eigenschaften des Ringes erfüllt und zusätzlich ein inverses Element bezüglich $\odot$ besitzt.

1.7. Lineare Abbildungen

Funktion/Abbildung $f : A \to B$ oder $f : A \ni a \to f(a) \in B$

Definitionsbereich Das A bei $f : A \to B$.

Zielmenge Das B bei $f : A \to B$.

Bildmenge $f(A) \subseteq B$, auch: Wertemenge/Bild von A bzgl. f.

Urbild $f^{-1}(b) := \{a \in A \mid f(a) = b\} \subseteq A$, Urbild von $b \in B$.

---

[1] Das inverse Element x^{-1} ist nicht mit dem Kehrwert zu verwechseln.

Injektivität $f(u) = f(v) \Rightarrow u = v$.

Surjektivität $f(A) = B$.

Bijektivität Injektiv und surjektiv.

1.7.1. Homomorphismus

A und B seien Vektorräume über einem Körper K. Eine Abbildung $f : A \to B$ heißt *Homomorphismus* bzw. *lineare Abbildung*, falls gilt:

Additivität $\forall x, y \in A : f(x + y) = f(x) + f(y)$

Homogenität $\forall x \in A \forall \lambda \in K : f(\lambda x) = \lambda f(x)$

Die Menge der linearen Abbildungen von A nach B wird auch mit $\mathrm{Hom}(A, B)$ bezeichnet.

Kern von f bzw. *Nullraum* von f: $\mathrm{Ker}(f) = \{x \in A \mid f(x) = 0\}$. Enthält entweder genau ein Element (die 0) oder unendlich viele.

Regulär $\mathrm{Ker}(f) = \{0\}$

Singulär $|\mathrm{Ker}(f)| = \infty$

Isomorphismus ist ein bijektiver Homomorphismus.

1.8. Modulo-Arithmetik

Zwei Zahlen $a, b \in \mathbb{Z}$ heißen zu einer festen Zahl $m \in \mathbb{N} \setminus \{0\}$ kongruent modulo m, wenn gilt:

$$a \equiv b \pmod{m} \quad \Leftrightarrow \quad |a - b| = n \cdot m,\ n \in \mathbb{N}$$

Das heißt, dass die Differenz beider Zahlen a und b durch m teilbar ist. Weiter gelten:

$$\begin{aligned}
(x + y) \bmod p &= ((x \bmod p) + (y \bmod p)) \bmod p \\
(x \cdot y) \bmod p &= ((x \bmod p) \cdot (y \bmod p)) \bmod p \\
x^n \bmod p &= (x \bmod p)^n \bmod p
\end{aligned}$$

2. Kombinatorik

Die Kombinatorik beschreibt eine Auswahl von k Objekten aus n Elementen, welche in einer Ergebnismenge zusammengefasst werden. Gesucht ist dann die Mächtigkeit a der Ergebnismenge.

Permutation Berücksichtigung der Reihenfolge

Kombination Ignorieren der Reihenfolge

	mit Wdh.	ohne Wdh.
Permutation	$\left\|\mathrm{PER}^{\mathrm{MW}}(n,k)\right\| = n^k$	$\left\|\mathrm{PER}^{\mathrm{OW}}(n,k)\right\| = \binom{n}{k} k!$
Kombination	$\left\|\mathrm{KOM}^{\mathrm{MW}}(n,k)\right\| = \binom{n+k-1}{k}$	$\left\|\mathrm{KOM}^{\mathrm{OW}}(n,k)\right\| = \binom{n}{k}$

Betragsstriche beachten! Die Funktionen geben die Kombinationen bzw. Permutationen zurück.

Teilchen-/Fächermodell Grundmenge: Fächer, Auswahl: Besetzung der Fächer mit Teilchen.

Mächtigkeit der Grundmenge ist Anzahl der Fächer, Experiment erfolgt durch Verteilen der Teilchen in die Fächer.

Urnenmodell Grundmenge: Kugeln in Urne, Auswahl: Ziehung der Kugeln.

Mächtigkeit der Grundmenge ist Anzahl der Kugeln, Experiment besteht aus Ziehung der Kugeln.

3. Komplexe Zahlen

Definition $z = x + iy = re^{i\varphi} = r\left(\cos(\varphi) + i\sin(\varphi)\right)$

Konjugiert komplex $\bar{z} = x - iy$

Betrag $|z| = \sqrt{x^2 + y^2} = \sqrt{z \cdot \bar{z}}$

Potenzen $z^n = r^n \exp(n \cdot i\varphi)$

Wurzeln $\sqrt[n]{z} = \sqrt[n]{r} \cdot \exp\left(\frac{i\varphi + 2k\pi i}{n}\right) \mid k = 0, 1, \ldots, (n-1)$

4. Sonstiges

4.1. Polarkoordinaten

$$
\begin{aligned}
\varphi &= \begin{cases} +\arccos\left(\frac{x}{r}\right) & \text{für } y \geq 0 \\ -\arccos\left(\frac{x}{r}\right) & \text{für } y < 0 \end{cases} \\
&= (\operatorname{sign}(y) + 1 - |\operatorname{sign}(y)|) \cdot \arccos\left(\frac{x}{r}\right)
\end{aligned}
$$

5. Beweis durch vollständige Induktion

☑ Beispiel vorhanden auf Seite 117.

Induktionsanfang (IA) Beweise, dass eine Aussage $f(n) = g(n)$ für ein gewähltes n_0 gilt (i.d.R. für $n_0 = 1$).

Induktionsvoraussetzung (IV) Gleichsetzen der rekursiven Formel und der expliziten Formel, also $f(n) = g(n)$ explizit aufschreiben.

Induktionsbehauptung (IB) Satz: „Wenn $f(n) = g(n)$ für ein beliebiges aber festes n gilt, so gilt die Aussage auch für $n + 1$.“

Induktionsschluss (IS) Überprüfen, ob $f(n+1) = g(n+1)$ unter Verwendung der IV stets eine wahre Aussage ergibt, d.h., eine Seite der Aussage muss mit Hilfe der IV in die andere Seite umgeformt werden. Beim Einsetzen der IV sollte über dem Gleichheitszeichen „IV“ stehen. Schlusssatz: „Damit ist die IV bewiesen.“

Teil II.

Lineare Algebra

6. Vektoren, Vektorräume

6.1. Definitionen

Vektor Gerichtete Strecke ohne Position.

Gebundener Vektor Vektor mit Position, zwischen zwei Punkten: $\overrightarrow{PQ}$.

Ortsvektor Gebundener Vektor mit Position im Ursprung.

Parallelität $\exists \alpha : \vec{a} = \alpha \vec{b} \Leftrightarrow \vec{a} \| \vec{b}$ (gleiche Richtung: $\alpha > 0$, gegensätzliche Richtung: $\alpha < 0$).

Orthogonalität $\left\langle \vec{a}, \vec{b} \right\rangle = 0 \quad \Leftrightarrow \quad \vec{a} \perp \vec{b} \quad \Leftrightarrow \quad \|\vec{a} + \vec{b}\|^2 = \|\vec{a}\|^2 + \|\vec{b}\|^2$

6.1.1. Vektorraum

Sei K ein beliebiger Körper und V eine nicht-leere Menge. V heißt Vektorraum, wenn für $\forall x, y \in V$ und $\forall \lambda, \mu \in K$ gilt:

- Folgende Abbildungen sind definiert:
  - $x \oplus y : \quad V \times V \mapsto V, \qquad (x, y) \mapsto x \oplus y$
  - $x \odot y : \quad K \times V \mapsto V, \qquad (\lambda, x) \mapsto \lambda \odot x$
- V ist eine kommutative Gruppe bezüglich $\oplus$.
- $\lambda \odot (\mu \odot x) = (\lambda \cdot \mu) \odot x$.
- $1 \odot x = x$ (hier ist 1 das Neutralelement bzgl. der Multiplikation aus K).
- $\lambda \odot (x \oplus y) = (\lambda \odot x) \oplus (\lambda \odot y)$.
- $(\lambda + \mu) \odot x = (\lambda \odot x) \oplus (\mu \odot x)$.

Dabei nennt man die x, y *Vektoren* und die λ, μ *Skalare*.

Linearkombination $\sum_{i=1}^{n} \lambda_i \vec{v}_i$ heißt Linearkombination von $\vec{v}_1, \ldots, \vec{v}_n$.

Lineare Hülle Die Menge aller Linearkombinationen $L = \left\{ \sum_{i=1}^{n} \lambda_i \vec{v}_i \mid \lambda_i \in K \right\} \subseteq V$ heißt lineare Hülle von $\vec{v}_1, \ldots, \vec{v}_n$.

Erzeugendensystem Gilt $L(\vec{v}_1, \ldots, \vec{v}_n) = V$, so bilden die Vektoren ein Erzeugendensystem.

Basis Auch: *minimales Erzeugendensystem.* Ist ein Erzeugendensystem, bei dem die Vektoren linear unabhängig sind. Daher hat eine Basis für einen Vektorraum V mit $\dim(V) = n$ genau n Basisvektoren.

Dimension Hat ein Vektorraum V eine Basis $\{\vec{v}_1, \ldots, \vec{v}_n\}$, so ist $\dim(V) = n$.

6.1.1.1. Untervektorraum

Eine Teilmenge U eines Vektorraums V über einem Körper K ist ein Untervektorraum, wenn gilt:

- $U \neq \varnothing$
- $\forall x, y \in U, \forall \lambda \in K \ : \ x + y \in U \wedge \lambda x \in U$

6.1.2. Skalarprodukt

V sei bel. Vektorraum über einem Körper $K \in \{\mathbb{R}, \mathbb{C}\}$ und $a,\ b,\ c \in V$ sowie $\lambda \in K$, dann heißt $\langle *, * \rangle : V \times V \mapsto K$ Skalarprodukt falls gilt:

SP0 $\langle a, b \rangle \in K$

SP1 $\langle a, b \rangle = \overline{\langle b, a \rangle}$ (Gilt wegen fehlendem Imaginärteil auch für $K \neq \mathbb{C}$.)

SP2 $\langle a, b + c \rangle = \langle a, b \rangle + \langle a, c \rangle$ und $\langle a + b, c \rangle = \langle a, c \rangle + \langle b, c \rangle$

SP3 $\langle \lambda a, b \rangle = \lambda \langle a, b \rangle = \left\langle a, \overline{\lambda} b \right\rangle$ (Gilt wegen fehlendem Imaginärteil auch für $K \neq \mathbb{C}$.)

SP4 $\langle a, a \rangle = 0 \Leftrightarrow a = 0$ und $\langle a, a \rangle > 0 \Leftrightarrow a \neq 0$

Standardskalarprodukt $\left\langle \vec{a}, \vec{b} \right\rangle := \sum_i a_i b_i$

6.1.3. Norm

Sei V ein Vektorraum über $\mathbb{R}$ sowie $a, b \in \mathbb{R}^n$ und $\lambda \in \mathbb{R}$, dann heißt $\| * \| : V \mapsto \mathbb{R}$ Norm falls $\forall a, b \in \mathbb{R}^n$ und $\forall \lambda \in \mathbb{R}$ gilt:

N0 $\|a\| \in \mathbb{R}$

N1 $\|a\| \geq 0$

N2 $\|a\| = 0 \Leftrightarrow a = 0$

N3 $\|\lambda a\| = |\lambda| \, \|a\|$

N4 $\|a + b\| \leq \|a\| + \|b\|$ (Dreiecksungleichung)

Standardnorm $\|a\| = \sqrt{\langle a, a \rangle}$ (auch *euklidische Norm*, *Zweiernorm* oder *Betrag*)

Betragssummennorm $\|a\|_1 = \sum_i |a_i|$ (auch *Einernorm*)

Maximumnorm $\|a\|_\infty = \max\left(\{|a_i|\}\right)$

Einheitsvektor $\|a\| = 1$

Kanonischer Einheitsvektor $\|a\| = 1 \wedge \exists_1 a_i : a_i = 1$

6.1.4. Errechnung von Normalen

Sei $\vec{v}$ ein Vektor mit n Komponenten, so lassen sich dazu $n-1$ linear unabhängige Normalen nach folgendem Prinzip errechnen:

1. Wähle eine Komponente $n_i \neq 0$.
2. Vertausche n_i mit einer anderen Komponente n_j $(i \neq j)$ und setze die restlichen Komponenten auf 0.
3. Negiere eine der gewählten gebliebenen Komponenten.

6.1.5. Orthonormalisierungsverfahren nach Gram-Schmidt

Wird benutzt, um aus m linear unabhängigen Vektoren $\vec{v}_1, \ldots, \vec{v}_m$ paarweise orthogonale normalisierte Vektoren $\vec{w}_1, \ldots, \vec{w}_m$ zu erzeugen.

1. Ersten normalisierten Vektor erzeugen:
$$\vec{w}_1 = \frac{\vec{v}_1}{\|\vec{v}_1\|}$$
2. Errechnen des nächsten orthogonalen Vektors:
$$\vec{r}_{k+1} = \vec{v}_{k+1} - \sum_{i=1}^{k} \langle \vec{v}_{k+1}, \vec{w}_i \rangle \, \vec{w}_i$$
3. Orthogonalen Vektor normalisieren:
$$\vec{w}_{k+1} = \frac{\vec{r}_{k+1}}{\|\vec{r}_{k+1}\|}$$
4. Für jeden weiteren Vektor bei 2. fortfahren.

6.2. Operationen mit zwei Vektoren

Projektion von $\vec{a}$ in Richtung $\vec{b}$ $\frac{\langle \vec{a},\vec{b}\rangle}{\langle \vec{b},\vec{b}\rangle}\vec{b}$ (Fußnote[1])

Winkel zwischen $\vec{a}$ und $\vec{b}$ $\left\langle \vec{a},\vec{b}\right\rangle = \|\vec{a}\| \cdot \|\vec{b}\| \cdot \cos(\varphi)$

Schwarzsche Ungleichung $\left|\left\langle \vec{a},\vec{b}\right\rangle\right| \leq \|\vec{a}\| \cdot \|\vec{b}\|$

Dreiecksungleichung $\|\vec{a}+\vec{b}\| \leq \|\vec{a}\| + \|\vec{b}\|$

6.2.1. Kreuzprodukt

Das Kreuzprodukt[2] ist nur definiert[3] im $\mathbb{R}^3$: $\|\vec{a} \times \vec{b}\| = \|\vec{a}\| \cdot \|\vec{b}\| \cdot \sin(\varphi)$. Es gelten folgende Eigenschaften:

Antikommutativität $\vec{a} \times \vec{b} = \text{-}\left(\vec{b} \times \vec{a}\right)$

Distributivität $\vec{a} \times \left(\vec{b}+\vec{c}\right) = \left(\vec{a} \times \vec{b}\right) + (\vec{a} \times \vec{c})$ ($\left(\vec{a}+\vec{b}\right) \times \vec{c}$ analog.)

Skalierbarkeit $\forall \alpha \in \mathbb{R}\ :\ (\alpha\vec{a}) \times \vec{b} = \alpha\left(\vec{a} \times \vec{b}\right) = \vec{a} \times \left(\alpha\vec{b}\right)$

Keine Assoziativität $\left(\vec{a} \times \vec{b}\right) \times \vec{c} \neq \vec{a} \times \left(\vec{b} \times \vec{c}\right)$

Außerdem gilt: $\|\vec{a} \times \vec{b}\|^2 = \langle \vec{a},\vec{a}\rangle \left\langle \vec{b},\vec{b}\right\rangle - \left\langle \vec{a},\vec{b}\right\rangle^2$

6.3. Vektorraum aus Polynomen

Rekursive Faktorzerlegung $p_n(x) = (x-x_0)p_{n-1}(x) + r$

Zerlegung mit n Nullstellen $p_n(x) = a_n \sum_{i=1}^{n} (x - x_i)$ wobei x_i die Nullstellen sind[4].

Die Funktionen $\{1, x, x^2, \ldots, x^n\}$ bilden eine Basis von P_n (Polynome vom Grad $\leq n$): $\dim(P_n) = n+1$.

P_n ist genau dann linear unabhängig, wenn $\sum_{i=0}^{n} \lambda_i x^i = 0(x) \Rightarrow \lambda_i = 0\, \forall_{0 \leq i \leq n}$ gilt.

[1] Auch: Komponente von $\vec{a}$ entlang $\vec{b}$.

[2] Auch: Vektorprodukt

[3] Das Kreuzprodukt lässt sich auch für beliebige $\mathbb{R}^n$ definieren, indem man für $n > 3$ Komponenten auf 0 setzt bzw. Komponenten für $n < 3$ mit 0 ergänzt.

[4] Siehe Hornerschema

6.4. Ebenengleichungen

Sei V ein n-dimensionaler Vektorraum über einem Körper K, $\vec{x}, \vec{p}, \vec{n}, \vec{r}_{1,\dots,n-1} \in V$ und $\lambda_{1,\dots,n-1} \in K$. $\vec{p}$ ist der Ortsvektor der (Hyper-)Ebene, $\vec{r}_{1,\dots,n-1}$ sind die Richtungsvektoren, $\vec{n}$ ist die Normale. Ein Körper der Dimension $n-1$ in einem Vektorraum der Dimension n wird als Hyperebene bezeichnet, für $n = 2$ als Gerade und für $n = 3$ einfach als Ebene.

Parameterform $E : \vec{x} = \vec{p} + \sum_{i=1}^{n-1} \lambda_i \vec{r}_i$ (Fußnote[5])

Normalform $E : \langle \vec{x} - \vec{p}, \vec{n} \rangle = 0$ bzw. $E : \langle \vec{x}, \vec{n} \rangle = \langle \vec{p}, \vec{n} \rangle$ (Fußnote[6])

Hessesche Normalform Wie gewöhnliche Normalform, nur mit $\|\vec{n}\| = 1$.

Koordinatenform Die Gleichung $\langle \vec{x}, \vec{n} \rangle = \langle \vec{p}, \vec{n} \rangle$ ausgeschrieben: $E : \sum_{i=1}^{n-1} x_i n_i = \langle \vec{p}, \vec{n} \rangle$.

6.4.1. Umformungen

		von		
		PF	*NF*	*KF*
nach	*PF*	–	via KF	s.u.
	NF	s.u.	–	s.u.
	KF	via NF	s.u.	–

Tabelle 6.1.: Umformungswege Ebenengleichungen

6.4.1.1. Parameterform zur Hesseschen Normalform

Im $\mathbb{R}^3$:

1. Normale $\vec{n}$ bestimmen: Vektorprodukt der Richtungsvektoren bilden und normalisieren.
2. Gleichung aufstellen: $E : \langle \vec{x}, \vec{n} \rangle = \langle \vec{p}, \vec{n} \rangle$, dabei die rechte Seite durch das Ergebnis des Skalarprodukts ersetzen.

Normalform zur Koordinatenform

Die Gleichung $\langle \vec{x}, \vec{n} \rangle = \langle \vec{p}, \vec{n} \rangle$ ausgeschrieben: $E : \sum_{i=1}^{n-1} x_i n_i = \langle \vec{p}, \vec{n} \rangle$.

[5] Auch: Punkt-Richtungs-Form.
[6] Auch: parameterlose Form.

Koordinatenform zur Hesseschen Normalform

1. Aus den Faktoren der x_i den Normalenvektor $\vec{n}$ bestimmen (der i-te Faktor ist die i-te Komponente der Normale) und ggf. normalisieren.
2. Einen beliebigen Punkt $\vec{p}$ der Ebene ermitteln.
3. Mithilfe der Parameterform die Hessesche Normalform ermitteln.

Koordinatenform zur Parameterform

1. n paarweise verschiedene Punkte der Ebene ermitteln.
2. Einen Punkt als Ortsvektor $\vec{p}$ wählen.
3. Von den restlichen Punkten $\vec{p}$ abziehen, das sind dann die Richtungsvektoren $\vec{r}_{1,\ldots,n-1}$.

Alternativen

Die Umrechnung der Koordinaten- und Normalform zur Parameterform lässt sich auch mit Hilfe der Errechnung von linear unabhängigen Normalen zur Ebenennormale durchführen. Siehe dazu Seite 18.

6.5. Abstandbestimmung

Allgemein:

1. Richtung $\vec{r}$ des Abstandsvektors bestimmen.
2. Vektor $\vec{a}$ als Vektor zweier beliebiger Punkte auf den beiden Objekten bestimmen (Verbindungsvektor).
3. Kürzester Abstandsvektor ist dann: $\vec{d} = \frac{\langle \vec{a}, \vec{r} \rangle}{\langle \vec{r}, \vec{r} \rangle} \vec{r}$

Hyperebenen $\vec{r}$ entspricht der Normale der Ebene.

Geraden im $\mathbb{R}^3$ Funktioniert nur bei nicht-parallelen Geraden, $\vec{r}$ entspricht dann dem Kreuzprodukt der Richtungsvektoren.

6.6. Schnittmengenbestimmung

☑ Beispiel vorhanden auf Seite 118.

1. Umformung beider Gleichungen in jeweils die parameterlose Form und die Parameterform.

2. Parameterlose Form in die Parameterform einsetzen und Wert eines Parameters berechnen.
3. Erhaltenen Parameter in die Parametergleichung einsetzen.

7. Matrizen

7.1. Definition

Eine $m \times n$-Matrix besteht aus m Zeilen und n Spalten. Die Indizes der einzelnen Werte bestehen aus der Zeile gefolgt von der Spalte.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Die Elemente einer Matrix vom Typ[1] (m, n) lassen sich mit $a_{ij} \mid 1 \leq i \leq m \wedge 1 \leq j \leq n$ ansprechen. Man kann Matrizen auch alternativ $A = (a_{ij}),\ i = 1, \ldots, m,\ j = 1, \ldots, n$ schreiben.

Gleichheit $A = B$ bzw. $(a_{ij}) = (b_{ij})$ gilt, wenn die Typen gleich sind und $\forall i, j : a_{ij} = b_{ij}$ gilt.

Addition, Subtraktion $A + B := (a_{ij} + b_{ij})$

Nullmatrix $a_{ij} = 0,\ i = 1, \ldots, m,\ j = 1, \ldots, n$

Einheitsmatrix $E = (\delta_{ij})$ mit $\delta_{ij} = \begin{cases} 0 & \text{falls } i \neq j \\ 1 & \text{falls } i = j \end{cases}$

Transponierte Matrix Die transponierte Matrix A^T einer Matrix A erhält man durch Vertauschung von Zeilen und Spalten.

Inverse Matrix Für die inverse Matrix A^{-1} gilt $AA^{-1} = A^{-1}A = E$. Es gibt nicht immer ein A^{-1}.

Quadratisch $m = n$

Zeilenmatrix $m = 1$

Spaltenmatrix $n = 1$

Rang Der Rang einer Matrix ist die Dimension der größten Untermatrix U, für die $\det(U) \neq 0$ ist.

[1] Der Typ kann auch als $m \times n$ geschrieben werden.

7.2. Rechenregeln

Multiplikation mit Skalar $\lambda A := (\lambda a_{ij})$

Multiplikation (allgemein) Voraussetzung: A ist eine $m \times n$-Matrix, B eine $n \times r$-Matrix. Dann gilt: $C = A \cdot B := (c_{ik})$ mit $c_{ik} = \sum_{j=1}^{n} a_{ij}\, b_{jk}$. Es entsteht eine $m \times r$-Matrix. *Achtung:* Im allgemeinen ist $AB \neq BA$.

Sofern die Produkte existieren, gelten:

Distributivgesetz $A(B + C) = AB + AC$

Assoziativgesetz $(AB)C = A(BC) = ABC$

- $(AB)^T = B^T A^T$
- $EA = AE = A$

7.2.1. Falksches Schema

Das Falksche Schema kann als Hilfsmittel zur Matrizenmultiplikation benutzt werden.

Hier ein Beispiel für $\begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$.

Schritt 1: Die rechte Matrix etwas höher aufschreiben als die linke.

$$\begin{array}{cc|cc} & & 1 & 2 \\ & & 0 & 1 \\ \hline 1 & 4 & & \\ 2 & 5 & & \\ 3 & 6 & & \end{array}$$

Schritt 2: Quasi die Skalarprodukte aus den Zeilen von A und den Spalten von B bilden.

$$\begin{array}{cc|cc} & & 1 & 2 \\ & & 0 & 1 \\ \hline 1 & 4 & 1 & \\ 2 & 5 & & \\ 3 & 6 & & \end{array}$$

... und so weiter ...

$$\begin{array}{cc|cc} & & 1 & 2 \\ & & 0 & 1 \\ \hline 1 & 4 & 1 & \\ 2 & 5 & 2 & \\ 3 & 6 & 3 & \end{array}$$

$$\begin{array}{cc|cc} & & 1 & 2 \\ & & 0 & 1 \\ \hline 1 & 4 & 1 & 6 \\ 2 & 5 & 2 & 9 \\ 3 & 6 & 3 & 12 \end{array}$$

Schritt 3: Produkt abschreiben.

$$C = \begin{pmatrix} 1 & 6 \\ 2 & 9 \\ 3 & 12 \end{pmatrix}$$

7.3. Quadratische Matrizen

Hauptdiagonale existiert nur bei quadratischen Matrizen ($m = n$), es sind die Elemente $(a_{kk}) \mid 1 \leq k \leq n$ gemeint.

Nebendiagonalen Siehe unten. Nebendiagonalen (markiert mit n) heißen sowohl die Diagonalen direkt über und direkt unter der Hauptdiagonalen (H) als auch die zur Hauptdiagonalen senkrechte Diagonale.

$$\begin{pmatrix} H & n & \circ & \circ & n \\ n & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & n \\ n & \circ & \circ & n & H \end{pmatrix}$$

Dreiecksmatrix Wenn alle Elemente über der Hauptdiagonalen 0 sind heißt die Matrix „untere Dreiecksmatrix“. Sind die Elemente *unter* der Hauptdiagonalen 0, so heißt die Matrix „obere Dreiecksmatrix“.

Diagonalmatrix $i \neq j \wedge a_{ij} = 0$

Symmetrische Matrix $a_{ij} = a_{ji} \Leftrightarrow A = A^T$

7.4. Lineare Gleichungssysteme

Ein lineares Gleichungssystem (LGS) besteht aus m linearen Gleichungen und n Unbekannten:

$$\sum_{j=1}^{m} a_{ij} x_j = b_i,\ 1 \leq i \leq m \quad \leftrightarrow \quad \begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & = & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & = & b_m \end{array}$$

Homogen $\forall i : b_i = 0$

Inhomogen $\exists i : b_i \neq 0$

Koeffizientenmatrix (a_{ij})

Erweiterte Matrix Koeffizientenmatrix erweitert um die b_i

Äquivalenz Zwei erweiterte Matrizen A, B sind äquivalent (Schreibweise: $A \sim B$), wenn sie dieselben Lösungen haben.

Stufenform Die Koeffizientenmatrix der erweiterten Matrix bildet eine obere Dreiecksmatrix (von oben nach unten verschiebt sich die Pivot-Spalte nach rechts).

Reduzierte Stufenform Die Koeffizientenmatrix hat pro Zeile und Pivot-Spalte genau eine 1 und sonst nur 0. Bei quadratischen Matrizen bildet die Koeffizientenmatrix eine Einheitsmatrix.

Pivot-Element Das erste Element einer Zeile[2] $\neq 0$.

Pivot-Spalte Spalte mit Pivot-Element.

Unterbestimmt Weniger linear unabhängige Gleichungen als Unbekannte.

Überbestimmt Mehr linear unabhängige Gleichungen als Unbekannte[3].

Nicht lösbar Lösungsspalte der reduzierten Stufenform ist eine Pivot-Spalte.

Eindeutig lösbar Die Koeffizientenmatrix der reduzierten Stufenform ist eine Einheitsmatrix. Bei quadratischen Koeffizientenmatrizen ist die Determinante $\neq 0$ (Gleichungen sind linear unabhängig).

Mehrdeutig lösbar Das LGS ist unterbestimmt. Bei quadratischen Koeffizientenmatrizen ist die Determinante $= 0$ (Gleichungen sind linear abhängig).

7.4.1. Lösungsmethoden

Einfache Verfahren sind Gleichsetzungs-, Einsetzungs- und Additionsverfahren (sinnvoll bis max. 3 Unbekannte).

Gauß-Algorithmus Standardverfahren, auch bekannt als GAUSSsches Eliminationsverfahren.

Cramersche Regel Lösungsverfahren mit Hilfe von Determinanten.

Methode der kleinsten Quadrate Wird benutzt, um bei überbestimmten Gleichungssystemen diejenige Lösung zu ermitteln, die die geringste Abweichung zum erwarteten Ergebnis liefert.

Verallgemeinerte Inverse Wird bei unterbestimmten Gleichungssystemen benutzt, um die Lösung mit der geringsten Norm zu ermitteln.

7.4.1.1. Gauß-Algorithmus

Ziel Bestimmung der zu einem LGS äquivalenten reduzierten Stufenform unter Verwendung von Äquivalenzumformungen.

[2] Auch die Lösungsspalte kann ein Pivot-Element enthalten.
[3] Über- und Unterbestimmtheit lassen sich erst nach Reduktion auf die Stufenform angeben.

Verwendete Rechenschritte

- Addition eines Vielfachen einer Zeile zu einer anderen
- Vertauschen zweier Zeilen
- Skalarmultiplikation mit einem Skalar $\lambda \in \mathbb{R} \setminus \{0\}$

Algorithmus

1. Spaltenweises Vorgehen: LGS in eine Stufenform überführen (nicht eindeutig).
2. Zeilenweises Vorgehen: Von unten nach oben das Pivotelement auf 1 bringen, Elemente darüber auf 0 bringen.

7.4.1.2. Cramersche Regel

Sei $A\vec{x} = \vec{b}$ ein lineares Gleichungssystem mit $A \in (n \times n, K)$ und $\vec{x}, \vec{b} \in K^n$ sowie $\det(A) \neq 0$, dann wird eine Matrix A_i definiert als Matrix A, in der die i-te Spalte durch $\vec{b}$ ersetzt wurde. Die Lösung des Gleichungssystems ist dann gegeben als $x_i = \frac{\det(A_i)}{\det(A)}$.

7.4.1.3. Methode der kleinsten Quadrate

Sei $A \in M(m \times n, \mathbb{R})$, $m \geq n$, $A\vec{x} = \vec{b}$, $\vec{b} \in \mathbb{R}^m$, dann gilt für die beste Näherungslösung:

$$\vec{x} = (A^T A)^{-1} A^T \vec{b}$$

7.4.1.4. Verallgemeinerte Inverse

Sei $A \in M(m \times n, \mathbb{R})$, $m < n$, $A\vec{x} = \vec{b}$, $\vec{b} \in \mathbb{R}^m$, dann gilt für die beste Näherungslösung mit der kleinsten Norm von $\vec{x}$:

$$\vec{x} = A^T (A A^T)^{-1} \vec{b}$$

7.4.1.5. Darstellung unendlich vieler Lösungen

1. Reduzierte Stufenform errechnen.
2. Nicht-Pivot-Spalten mit dem entsprechenden freien Koeffizienten durch Subtraktion auf die rechte Seite bringen.
3. Erweiterung um die fehlenden Zeilen: Konstante Vektoren um 0 ergänzen, abhängige Vektoren der entsprechenden freien Koeffizienten um 1 ergänzen.

Beispiel:

$$\left(\begin{array}{ccc|c} 2 & -4 & 1 & 8 \\ 1 & -1 & -7 & 6 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & 0 & -15 & 8 \\ 0 & 1 & -8 & 2 \end{array}\right)$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \end{pmatrix} + z \cdot \begin{pmatrix} 15 \\ 8 \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \\ 0 \end{pmatrix} + z \cdot \begin{pmatrix} 15 \\ 8 \\ 1 \end{pmatrix}$$

7.5. Determinanten

Determinanten können nur für quadratische Matrizen berechnet werden. Ihr Wert ist ein Skalar. Bis zur Größe 3×3 gibt es spezielle Gleichungen. Die Menge aller $m \times n$-Matrizen über einem Körper K wird als $M(m \times n, K)$ geschrieben.

$n = 1$ $\det(a_1) = a_1$

$n = 2$ $\det \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} = \det(\vec{a}, \vec{b}) = a_1 b_2 - b_1 a_2$

$n = 3$ $\det \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} = \det(\vec{a}, \vec{b}, \vec{c}) = a_1 b_2 c_3 + b_1 c_2 a_3 + c_1 a_2 b_3 - a_1 c_2 b_3 - b_1 a_2 c_3 - c_1 b_2 a_3$

Letzteres lässt sich mit der Regel von Sarrus veranschaulichen:

$$\begin{array}{ccccccccc} & & a_1 & b_1 & c_1 & a_1 & b_1 & & \\ & & a_2 & b_2 & c_2 & a_2 & b_2 & & \\ & & a_3 & b_3 & c_3 & a_3 & b_3 & & \\ - & - & - & & & + & + & + & \end{array}$$

7.5.1. Rechenregeln

Die folgenden Regeln lassen sich analog auch für den $\mathbb{R}^n$ anwenden. Die Regeln für die Spalten gelten Aufgrund von D6 auch für Zeilen.

D1 $\det(\vec{a}, \vec{b}, \vec{c}) = \det(\vec{c}, \vec{a}, \vec{b}) = \det(\vec{b}, \vec{c}, \vec{a})$ (Rotation von Spalten)

D2 $\det(\vec{a}, \vec{b}, \vec{c}) = -\det(\vec{b}, \vec{a}, \vec{c})$ (Vertauschung von Spalten)

D3 $\det(\vec{a}, \vec{a}, \vec{b}) = 0$ (Duplikat einer Spalte)

D4 $\forall \alpha \in \mathbb{R} : \det(\alpha \cdot \vec{a}, \vec{b}, \vec{c}) = \alpha \cdot \det(\vec{a}, \vec{b}, \vec{c})$ (Multiplikation einer Spalte mit einem Skalar)

D5 $\det(\vec{a}, \vec{b}, \vec{c} + \vec{d}) = \det(\vec{a}, \vec{b}, \vec{c}) + \det(\vec{a}, \vec{b}, \vec{d})$ (Addition zweier Spalten)

D6 $\det(A) = \det(A^T)$ (Invarianz bei Transponierung)

Außerdem gilt im $\mathbb{R}^3$:

$$\left\langle \vec{a} \times \vec{b}, \vec{c} \right\rangle = \det(\vec{a}, \vec{b}, \vec{c})$$

(Die Determinante im $\mathbb{R}^3$ gibt also das Volumen des durch die Vektoren $\vec{a}$, $\vec{b}$ und $\vec{c}$ aufgespannten Spats an.)

Weiter gilt: $\det(AB) = \det(A) \cdot \det(B)$.

7.5.2. Entwicklungssatz nach Laplace

Sei A_{ij} definiert als die $(n-1) \times (n-1)$-Untermatrix der Matrix A durch Streichen von Zeile i und Spalte j. Dann gilt für die Determinante:

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

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

Dabei ergibt $(\text{-}1)^{i+j}$ ein Schachbrettmuster:

$$\begin{pmatrix} + & - & + & \cdots \\ - & + & - & \cdots \\ + & - & + & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

7.5.3. Berechnung nach Gauß

Formt man die Matrix in eine Dreiecksmatrix um, so gilt für die Determinante:

$$\det(A) = \prod_{i=1}^{n} a_{ii}$$

7.5.4. Komplementärmatrix

Die Komplementärmatrix (auch *Adjunkte* genannt) $\tilde{A} = (\tilde{a}_{ij})$ zu einer Matrix A ist mit Hilfe der Unterdeterminanten (*Minore*, Einzahl *Minor*) definiert als

$$\begin{aligned} \tilde{a}_{ij} &= (\text{-}1)^{i+j} \cdot \det(A_{ji}) \\ \tilde{A} &= \operatorname{adj}(A) \end{aligned}$$

Es bietet sich an:

1. Aufschreiben der Minore und nach dem Schachbrettmuster negieren (also $\tilde{A}^T$ aufschreiben).

2. Transponieren der neuen Matrix.

7.5.5. Umkehrmatrix

7.5.5.1. Berechnung mit dem Gauß-Jordan-Algorithmus

Funktioniert wie die Lösung eines linearen Gleichungssystems. Man Schreibt eine Blockmatrix der Form $A|E$ auf und versucht, diese mit den Rechenregeln für Determinanten in die Form $E|B$ zu bringen. Gelingt dies, so ist $B = A^{-1}$.

7.5.5.2. Berechnung mit der Komplementärmatrix

Es gilt $A^{-1} = \frac{1}{\det(A)}\tilde{A}$.

7.5.6. Definitheit

Eine symmetrische Matrix A ist:

positiv definit	$\forall\vec{x} \neq \vec{0} : \vec{x}^T \cdot A \cdot \vec{x} > 0$
positiv semi-definit	$\forall\vec{x} \neq \vec{0} : \vec{x}^T \cdot A \cdot \vec{x} \geq 0$
negativ definit	$\forall\vec{x} \neq \vec{0} : \vec{x}^T \cdot A \cdot \vec{x} < 0$
negativ semi-definit	$\forall\vec{x} \neq \vec{0} : \vec{x}^T \cdot A \cdot \vec{x} \leq 0$
indefinit	wenn keine Aussage zutrifft

7.5.6.1. Hurwitz-Kriterium

Sei D_k die Determinante der $k \times k$-Teilmatrix von A von links oben bzw. rechts unten. Dann gilt für A:

positiv definit	$D_k > 0$
positiv semi-definit	$D_k > 0$
negativ definit	$(-1)^k D_k > 0$
negativ semi-definit	$(-1)^k D_k \geq 0$

7.6. Transformationsmatrizen

Die Koordinaten eines Vektors $\vec{x}$ zu einer Basis A werden als $K_A(\vec{x})$ geschrieben und geben die Linearfaktoren zur Basis A an, wodurch die Linearkombination der Vektoren von A zu $\vec{x}$ eindeutig bestimmt ist.

Zur Transformation der Koordinaten von einer Basis A zu einer Basis B und umgekehrt gilt folgendes:

Von A **nach** B $K_B(\vec{x}) = \underbrace{B^{-1}A}_{T_B^A} \cdot K_A(\vec{x})$

Von B **nach** A $K_A(\vec{x}) = \underbrace{A^{-1}B}_{T_A^B} \cdot K_B(\vec{x})$

Dabei heißt T_B^A *Transformationsmatrix* von A nach B. In T_B^A stehen die Koordinaten der Einheitsvektoren von A zur Basis B.

7.6.1. Basistransformation

Seien A und A' Basen des $\mathbb{R}^m$ sowie B und B' Basen des $\mathbb{R}^n$ und $\Phi : \mathbb{R}^m \to \mathbb{R}^n$ eine lineare Abbildung, dann lässt sich eine *Abbildungsmatrix* $M_B^A(\Phi)$ von Φ bezüglich A und B angeben:

$$
\begin{aligned}
K_B(\Phi(\vec{x})) &= M_B^A(\Phi) \cdot K_A(\vec{x}) \\
M_{B'}^{A'}(\Phi) &= T_{B'}^B \cdot M_B^A(\Phi) \cdot T_A^{A'}
\end{aligned}
$$

$$
\begin{tikzcd}
K_A(\vec{x}) \arrow[r, "M_B^A"] & K_B(\Phi(\vec{x})) \arrow[d, "T_{B'}^B"] \\
K_{A'}(\vec{x}) \arrow[u, "T_A^{A'}"'] \arrow[r, "M_{B'}^{A'}"'] & K_{B'}(\Phi(\vec{x}))
\end{tikzcd}
$$

7.7. Orthogonale Matrizen

Eine Matrix $A \in M(n \times n, \mathbb{R})$ heißt orthogonal, wenn $A^T A = E$ bzw. $A = A^{-1}$ gilt. Man schreibt $A \in O(n)$. Die Spalten- bzw. Zeilenvektoren von A bilden jeweils ein Orthonormalsystem, daher gilt $|\det(A)| = 1$.

Drehmatrix im $\mathbb{R}^2$

$$A = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

7.8. Eigenwerte und -vektoren

1. Lösung des *charakteristischen Polynoms* $p(\lambda) = \det(\lambda E - A) \stackrel{!}{=} 0$, die gefundenen λ sind die *Eigenwerte*.
2. Einsetzen der λ_i in $(\lambda_i E - A)\vec{x} = 0$. Die *speziellen* Lösungen $\vec{x}_i$ sind die *Eigenvektoren* zu den entsprechenden Eigenwerten.

8. Lineare Abbildungen

8.1. Definitionen

Seien V, W Vektorräume über einem Körper K ($f : V \to W$), dann heißt f *lineare Abbildung* bzw. *Homomorphismus*, wenn $\forall x, y \in V$ und $\forall \lambda \in K$ gilt:

- $f(x + y) = f(x) + f(y)$
- $f(\lambda x) = \lambda f(x)$

Die Menge aller Homomorphismen über zwei Vektorräume V, W wird als $\hom(V, W)$ bezeichnet.

Bild $\text{bild}(f) = f(V) := \{f(v) \mid v \in V\} \subseteq W$ (V ist Unterraum von W, wenn f linear ist.) Die lineare Hülle der Spaltenvektoren der $m \times n$-Abbildungsmatrix einer Abbildung $f : \mathbb{R}^n \to \mathbb{R}^m$ ist gleich $\text{bild}(f)$.

Kern $\ker(f) := \{v \in V \mid f(v) = 0\}$ (auch genannt: *Nullraum* von f). Es ist immer $0 \in \ker(f)$. Außerdem: $|\ker(f)| \in \{1, \infty\}$

Rang $\text{rg}(f) = \dim(\text{bild}(f))$

Dimensionssatz $\dim(V) = \dim(\ker(f)) + \text{rg}(f)$. Wenn $\dim(V) = \dim(W) < \infty$, dann gilt: f injektiv $\Leftrightarrow$ f surjektiv.

regulär $\ker(f) = \{0\}$

singulär $|\ker(f)| > 1$

Injektivität f regulär $\Leftrightarrow$ f injektiv

Matrixdarstellung Jede $m \times n$-Matrix liefert eine lineare Abbildung.

Isomorphismus f bijektiv, A quadratisch, $\det(A) \neq 0$. Das Bild der Basis von V ist eine Basis von W.

Urbild $f^{-1} := \{v \mid f(v) = w \; \forall w \in W\}$. Immer vorhanden.

Umkehrabbildung Urbild einer linearen Abbildung: $f(f^{-1}(x)) = f^{-1}(f(x)) = x$

8.1.1. Überprüfung auf Linearität

Es gibt zwei Möglichkeiten:

1. Obige Definition des Homomorphismus überprüfen.
2. $f(x)$ als Matrixdarstellung Ax, dann prüfen, ob $f(x) = Ax$ (gilt nur für $\mathbb{R}^n \to \mathbb{R}^m$).

8.1.2. Berechnung der Abbildung

Seien V, W Vektorräume über einem Körper K, $\{v_1, \ldots, v_n\}$ Basis von V, $\{w_1, \ldots, w_n\}$ beliebige Vektoren von W, dann gilt:

$$\exists_1 f : V \to W \forall 1 \leq i \leq n : f(v_i) = w_i$$

Damit ist f durch Abbildung der v_i eindeutig bestimmt.

8.1.3. Vektorraum aus Matrizen

Vektorraum der $m \times n$-Matrizen über einem Körper K bzgl. der Matrixaddition und –Multiplikation mit einem Skalar: $M(m \times n, K)$.

Invertierbarkeit Wenn $\exists A \in \mathbb{R}^{n \times n} : AA^{-1} = A^{-1}A = E$ gilt, dann heißt A invertierbar und A^{-1} Inverse von A. A ist genau dann invertierbar, wenn $\det(A) \neq 0$.
Es gilt: $(A^{-1})^T = (A^T)^{-1}$ und $(\lambda A)^{-1} = \frac{1}{\lambda} A^{-1}$.

Teil III.

Analysis

9. Folgen

☑ Beispiel vorhanden auf Seite 115.

9.1. Definitionen

Nach oben beschränkt $\exists S: \ a_n \leq S \,\forall n$

Nach unten beschränkt $\exists S: \ a_n \geq S \,\forall n$

Beschränkt $\exists S: \ |a_n| \leq S \,\forall n$

Alternierend $a_n \cdot a_{n+1} < 0 \,\forall n$

Häufungswert Eine Zahl h heißt Häufungswert, wenn es unendlich viele n gibt, für die $|h - a_n| < \varepsilon$ gilt.

Grenzwert Eine Zahl g heißt Grenzwert, wenn $\exists n_0 : |g - a_n| < \varepsilon \,\forall n \geq n_0$ gilt. Dann ist $g = \lim_{n\to\infty} a_n$.

Konvergenz/Divergenz Eine Folge heißt konvergent, wenn sie einen Grenzwert besitzt, sonst divergent. *Jede konvergente Folge ist beschränkt.*

Nullfolge Eine Folge mit Grenzwert 0 heißt Nullfolge.

Satz von Bolzano-Weierstraß Jede beschränkte Folge hat einen Häufungswert.

Aus obigen Definitionen lässt sich folgern:

- Jeder Grenzwert ist ein Häufungswert.
- Ist h der einzige Häufungswert einer Folge, so ist er auch der Grenzwert der Folge.
- Eine Folge mit mehreren Häufungswerten ist divergent.
- Ein h ist ein Häufungswert einer Folge (a_n), wenn es eine Unterfolge (a'_n) gibt, deren Grenzwert h ist.

9.2. Wichtige Grenzwerte

$$\sqrt[n]{a} \to 1$$

$$\sqrt[n]{n} \to 1$$

$$\sqrt[n]{n!} \to \infty$$

$$\left(1 + \frac{1}{n}\right)^n \to e$$

$$\left(1 + \frac{a}{n}\right)^n \to e^a$$

$$\left(1 - \frac{a}{n}\right)^n \to e^{-a}$$

$$\frac{1}{n}\sqrt[n]{n!} \to \frac{1}{e}$$

$$\frac{a^n}{n!} \to 0$$

$$\frac{a^n}{n!} \to 0$$

$$\frac{n^n}{n!} \to \infty$$

$$\binom{a}{n} \to 0 \mid a > -1$$

$$\frac{a^n}{n^k} \to \infty \mid a > 1 \wedge k = \text{const.}$$

$$\frac{a^n}{n^k} \to 0 \mid |a| < 1 \wedge k = \text{const.}$$

9.3. Rechenregeln

$$\begin{matrix} a_n \to a \\ b_n \to b \end{matrix} \Rightarrow \begin{cases} a_n \pm b_n & \to a \pm b \\ a_n \cdot b_n & \to a \cdot b \\ \frac{a_n}{b_n} & \to \frac{a}{b} \quad \text{für } b_n \neq 0,\ b \neq 0 \\ a_n^{b_n} & \to a^b \quad \text{für } a_n > 0,\ a > 0 \\ a_n^c & \to a^c \quad \text{für } a_n > 0,\ a > 0,\ c \in \mathbb{R} \end{cases}$$

Arithmetisches Mittel $a_n \to a \Rightarrow \frac{a_1 + \cdots + a_n}{n} \to a$

Geometrisches Mittel $a_n \to a,\ a_n > 0 \Rightarrow \sqrt[n]{a_1 \cdots a_n} \to a$

Quetschlemma $a_n, b_n \to a \wedge a_n \leq x_n \leq b_n\ \forall n \geq n_0 \ \Rightarrow\ x_n \to a$

Geometrische Folge $a_n = q^n$ ist für $|q| < 1$ eine Nullfolge, für $|q| > 1$ divergent.

Polynom-Brüche Für $a_n = \frac{P(n)}{Q(n)}$ sieht man das Verhalten, wenn man P und Q durch die höchste Potenz von n im Nenner kürzt.

9.4. Rekursive Folgen

Monotoniekriterium Jede monotone und beschränkte Folge ist konvergent.

Konvergenzkriterium von Cauchy Eine Folge (a_n) konvergiert genau dann, wenn $\exists n_0 \forall n, m > n_0 : |a_m - a_n| < \varepsilon$ gilt[1].

[1] Auch bekannt als „Kriterium ohne Grenzwert".

10. Reihen

☑ Beispiel vorhanden auf Seite 116.

10.1. Definitionen

Partialsumme $s_n := \sum_{k=0}^{n} a_k$ heißt Partialsumme der unendlichen Reihe $\sum_{k=0}^{\infty} a_k$.

Unendliche Reihe Die unendliche Reihe ist definiert als Folge der Partialsummen s_n.

Konvergenz/Divergenz Ist die Partialsummenfolge konvergent, so konvergiert auch die Reihe. Konvergiert die Partialsummenfolge *nicht*, so ist die Reihe divergent.

Grenzwert Im Konvergenzfall von (s_n) ist der Grenzwert der Partialsummenfolge $g = \lim\limits_{n\to\infty} s_n$.

Endliche geometrische Reihe $\sum_{k=0}^{n} q^k = \frac{1-q^{n+1}}{1-q} \mid q \neq 1$

Geometrische Reihe $\sum_{k=0}^{\infty} q^k$ konvergiert wenn $|q| < 1$. Dann ist der Grenzwert $\frac{1}{1-q}$.

Harmonische Reihe $\sum_{k=0}^{\infty} \frac{1}{k^a}$ konvergiert wenn $a > 1$.

Notwendiges Konvergenzkriterium $\sum_{k=0}^{\infty} a_k$ konvergiert nur, wenn (a_k) eine Nullfolge ist.

Leibnizkriterium $\sum_{k=0}^{\infty}(-1)^k a_k,\ a_k > 0$ konvergiert nur, wenn (a_k) eine monotone Nullfolge ist.

Absolute Konvergenz $\sum_{k=0}^{\infty} a_k$ heißt absolut konvergent, wenn $\sum_{k=0}^{\infty} |a_k|$ konvergiert.

Alternierend $a_k \cdot a_{k+1} < 0\ \forall k \in \mathbb{N}$

10.2. Rechenregeln

Sind $\sum_{k=1}^{\infty} a_k = a$, $\sum_{k=1}^{\infty} b_k = b$ konvergente Reihen und ist $r \in \mathbb{R}$, so gilt:

Addition $\sum_{k=1}^{\infty} (a_k + b_k) = \sum_{k=1}^{\infty} a_k + \sum_{k=1}^{\infty} b_k = a + b$

Multiplikation $\sum_{k=1}^{\infty} r \cdot a_k = r \cdot \sum_{k=1}^{\infty} a_k = r \cdot a$

Cauchy-Produkt $\left(\sum_{k=0}^{\infty} a_k\right) \cdot \left(\sum_{k=0}^{\infty} b_k\right) = \sum_{k=0}^{\infty} \left(\sum_{i=0}^{k} a_i \cdot b_{k-i}\right)$

Cauchy-Produkt für absolut konvergente Reihen $\left(\sum_{k=0}^{\infty} a_k\right) \cdot \left(\sum_{k=0}^{\infty} b_k\right) = a \cdot b$

Cauchy-Schwarz-Ungleichung $\sum_{k=1}^{n} a_k \cdot b_k \leq \sqrt{\sum_{k=1}^{n} a_k^2} \cdot \sqrt{\sum_{k=1}^{n} b_k^2}$

Minkowsky-Ungleichung $\sum_{k=1}^{n} (a_k + b_k)^2 \leq \sqrt{\sum_{k=1}^{n} a_k^2} \cdot \sqrt{\sum_{k=1}^{n} b_k^2}$

10.3. Kriterien

Majorantenkriterium $\sum_{k=1}^{\infty} x_k$ ist absolut konvergent, wenn es eine konvergente Reihe $\sum_{k=1}^{\infty} a_k \mid a_k > 0$ gibt, für die $\forall k \geq k_0 : |x_k| < a_k$ gilt.

Minorantenkriterium $\sum_{k=1}^{\infty} x_k$ ist divergent, wenn es eine divergente Reihe $\sum_{k=1}^{\infty} a_k \mid a_k > 0$ gibt, für die $\forall k \geq k_0 : a_k < x_k$ gilt.

Verdichtungskriterium Ist (x_k) eine monoton fallende Nullfolge, so sind $\sum_{k=1}^{\infty} x_k$ und $\sum_{k=1}^{\infty} 2^k x_{(2^k)}$ entweder beide divergent oder beide konvergent.

Quotientenkriterium Sei $g = \lim\limits_{k\to\infty} \left|\frac{x_{k+1}}{x_k}\right|$ und $x = \sum_{k=1}^{\infty} x_k$, dann ist x absolut konvergent für $g < 1$, divergent für $g > 1$. Keine Aussage falls $g = 1$.

Wurzelkriterium Sei $g = \lim\limits_{k\to\infty} \sqrt[k]{|x_k|}$ und $x = \sum_{k=1}^{\infty} x_k$, dann ist x absolut konvergent für $g < 1$, divergent für $g > 1$. Keine Aussage falls $g = 1$.

Grenzwertkriterium Sind (a_k) und (b_k) Folgen mit $a_k, b_k > 0$ und $0 < \lim\limits_{k\to\infty} \frac{a_k}{b_k} < \infty$, so sind die Reihen entweder beide konvergent oder beide divergent.

Integralkriterium Ist $f(x) > 0$ monoton fallend auf $[m, \infty[$, so zeigen $\sum_{k=m}^{\infty} f(k)$ und $\int_m^{\infty} f(x)\,\mathrm{d}x$ dasselbe Konvergenzverhalten.

11. Potenzreihen

11.1. Definitionen

Reihen der Form

$$\sum_{n=0}^{\infty} a_n (x - x_0)^n$$

heißen Potenzreihen.

Koeffizienten sind die a_n.

Entwicklungspunkt ist das x_0.

Konvergenzbereich Menge aller einsetzbaren x, so dass die entstehende numerische Reihe konvergiert.

Konvergenzradius Der Konvergenzradius r wird bestimmt durch $\frac{1}{r} = \lim\limits_{n\to\infty} \sqrt[n]{|a_n|}$ oder $\frac{1}{r} = \lim\limits_{n\to\infty} \left|\frac{a_{n+1}}{a_n}\right|$.

Ableitungsregel $f^{(n)}(x_0) = n! \cdot a_n$. Daraus folgen $f'(x) = \sum_{n=1}^{\infty} n a_n (x - x_0)^{n-1}$ und $f''(x) = \sum_{n=2}^{\infty} n(n-1) a_n (x - x_0)^{n-2}$, etc..

11.2. Wichtige Grenzwerte

$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x} \mid \lvert x\rvert < 1$	$\sum_{n=0}^{\infty} \binom{r}{n} x^n = (1+x)^r \mid \lvert x\rvert < 1$

11.3. Rechenregeln

Mit $f(x) = \sum_{n=0}^{\infty} a_n x^n$ und $g(x) = \sum_{n=0}^{\infty} b_n x^n$:

$$f(x) \pm g(x) = \sum_{n=0}^{\infty} (a_n \pm b_n)\, x^n$$

$$f(x) \cdot g(x) = \sum_{n=0}^{\infty} c_n x^n \text{ mit } c_n = \sum_{k=0}^{n} a_k b_{n-k}$$

Siehe das Cauchy-Produkt auf Seite 42.

$$f(g(x)) = \sum_{n=0}^{\infty} a_n \left(\sum_{k=0}^{\infty} b_k x^k \right)^n$$

Potenzreihen dürfen summandenweise differenziert und integriert werden. Dabei bleibt der Konvergenzradius unverändert.

11.4. Wichtige Potenzeihen

11.4.1. Taylorreihe

Das Taylorpolynom n-ten Grades der Funktion f bei x_0 ist definiert als

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

Der Taylorsatz sagt: Ist f in einer Umgebung x_0 hinreichend oft differenzierbar, so gilt $f(x) = T_n(x) + R_n(x)$, wobei das Restglied definiert ist als:

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}$$

wobei ξ zwischen x_0 und x liegt.

Achtung! Da ξ i.d.R. nicht bekannt ist, kann man das Restglied nur abschätzen!

11.5. Taylorreihenentwicklungen

Bei rationalen Funktionen kann man eine Partialbruchzerlegung versuchen, um auf folgende Taylorreihen zu kommen (gilt nur für $|z| < 1$):

$\frac{1}{1-z} = \sum_{n=0}^{\infty} z^n$	$\frac{1}{1+z} = \sum_{n=0}^{\infty} (-1)^n z^n$	$\frac{1}{(1-z)^2} = \sum_{n=0}^{\infty} (n+1) z^n$

11.6. Newton-Verfahren

zur näherungsweisen Bestimmung:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

12. Funktionen mehrerer Veränderlicher

12.1. Allgemeines

- Schreibweise: $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$.
- Folgen im $\mathbb{R}^n$: $\langle x_n \rangle$
  - Analog: Konvergenz, Häufungspunkte, Grenzwert.
  - Konvergenz: $\forall \varepsilon > 0 : \exists n_0(\varepsilon) : \forall n \geq n_0(e) : \|\vec{x}_n - \vec{x}\| < \varepsilon$.
- $\vec{f}$ lässt sich als Vektor von Komponentenfunktionen schreiben.
- Für viele (nicht alle) Eigenschaften gilt: $\vec{f}$ hat die entsprechende Eigenschaft, wenn alle Komponentenfunktionen diese Eigenschaft haben.
- Im $\mathbb{R}^n$ sind alle Normen äquivalent.

12.1.1. Stetigkeit

Sei $U \subseteq \mathbb{R}^n$ eine offene Menge, $f : U \to \mathbb{R}^n$, $\vec{x}_0 \in U$. Dann heißt f stetig in $\vec{x}_0$, wenn für alle Folgen $\vec{x}_n$ mit Grenzwert $\vec{x}_0$ gilt:

$$\lim_{n\to\infty} f(\vec{x}_n) = f\left(\lim_{n\to\infty} \vec{x}_n\right) = f(\vec{x}_0)$$

Formal: $\lim\limits_{\vec{x}\to\vec{x}_0} f(\vec{x}) := \lim\limits_{n\to\infty} f(\vec{x}_n)$ für alle Folgen $\vec{x}_n$.

12.1.2. Epsilon-Umgebung

Sei $\| * \|$ eine Norm im $\mathbb{R}^n$, dann heißt $U_\varepsilon(\vec{x}_0) = \{\vec{x} \in \mathbb{R}^n \mid \|\vec{x} - \vec{x}_0\| < \varepsilon\}$ eine Epsilon-Umgebung von $\vec{x}_0$ bzgl. $\| * \|$.

12.1.3. Weiteres

Sei $D \subseteq \mathbb{R}^n$.

Innerer Punkt $\vec{x}_0$ heißt innerer Punkt, wenn gilt: $\exists \varepsilon > 0 : U_\varepsilon(\vec{x}_0) \subseteq D$. (D.h., $\vec{x}_0$ ist kein Randpunkt von D.)

Offene Menge D heißt offene Menge, wenn alle Punkte von D innere Punkte sind.

Häufungspunkt $\vec{x}$ heißt Häufungspunkt, wenn gilt: $\forall\varepsilon > 0\ \exists\vec{x} \neq \vec{x}_0 \in D : \vec{x} \in U_\varepsilon(\vec{x}_0)$

Abgeschlossen D heißt abgeschlossen, wenn D alle Häufungspunkte enthält.

Kompakt D heißt kompakt, wenn D abgeschlossen und beschränkt ist.

Konvex D heißt konvex, wenn $\forall a, b \in D : \forall\lambda \in [0; 1] : \lambda a + (1 - \lambda)b \in D$ gilt. In Worten: Die Verbindungslinie zwischen zwei Punkten a und b liegt komplett in D.

12.2. Partielle Ableitungen

12.2.1. Allgemeines

Schreibweisen für die Ableitung einer Funktion $f(\vec{x})$ nach einer Komponente x_i: $\frac{\partial}{\partial x_i}f$, $D_i f$, f_{x_i}.

Partiell diff'bar Alle partiellen Ableitungen existieren.

Stetig partiell diff'bar Alle partiellen Ableitungen sind stetig. Ist dies in einem Punkt $\vec{x}_0$ der Fall, so ist f in $\vec{x}_0$ stetig.

Nabla-Operator $\nabla := \begin{pmatrix} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{pmatrix}^T$

Gradient Der Gradient einer Funktion f ist eine Zeilenmatrix mit den komponentenweisen partiellen Ableitungen: $\mathrm{grad}(f) := \begin{pmatrix} f_{x_1} & \cdots & f_{x_n} \end{pmatrix}^T = \nabla \cdot f$ (f ist hier als Skalar anzusehen, sodass die Funktion mit jeder Komponente von ∇ multipliziert wird).

Divergenz $\mathrm{div}(\vec{f}) := \left\langle \nabla, \vec{f} \right\rangle$ – Punkte mit $\mathrm{div}(\vec{f}) > 0$ heißen Quellen, Punkte mit $\mathrm{div}(\vec{f}) < 0$ heißen Senken.

Rotation $\mathrm{rot}(\vec{f}) := \nabla \times \vec{f}$

Jacobi-Matrix $J := A\vec{f} = \begin{pmatrix} \nabla \cdot f_1 \\ \vdots \\ \nabla \cdot f_n \end{pmatrix}$

Totale Diff'barkeit $\vec{f} : \mathbb{R}^n \to \mathbb{R}^m$ heißt total/vollständig diff'bar in $\vec{x}_0$, wenn eine Jacobi-Matrix existiert mit $\lim\limits_{\vec{x}\to\vec{x}_0} \frac{f(\vec{x})-f(\vec{x}_0)-J(\vec{x}_0)(\vec{x}-\vec{x}_0)}{\|\vec{x}-\vec{x}_0\|} = \vec{0}$.

Richtungsableitung im Punkt $\vec{x}_0$ in Richtung $\vec{a}$: $\nabla f(\vec{x}_0) \cdot \frac{\vec{a}}{\|\vec{a}\|}$

Tangentialebene im Punkt $\vec{x}_0$: $f(\vec{x}_0) + \nabla f(\vec{x}_0)(\vec{x} - \vec{x}_0)$

12.2.2. Extrema

Voraussetzung Die Funktion muss zweimal stetig diff'bar sein und auf einer offenen Teilmenge des $\mathbb{R}^n$ definiert sein.

Notwendige Bedingung $\operatorname{grad}(f)(\vec{x}_0) = \vec{0}$

Kritischer Punkt Ein Punkt $\vec{x}$ heißt kritischer Punkt, wenn er die notwendige Bedingung erfüllt.

Hesse-Matrix Für eine Funktion $f : \mathbb{R} \to \mathbb{R}^n$ ist die HESSE-Matrix definiert als $Hf := (f_{x_i x_j}) \mid i, j = 1, \ldots, n$.

Für eine Funktion $f : \mathbb{R}^3 \to \mathbb{R}$ wäre die Hesse-Matrix also:

$$Hf = \begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{yx} & f_{yy} & f_{yz} \\ f_{zx} & f_{zy} & f_{zz} \end{pmatrix}$$

Für zweimal stetig diff'bare Funktionen ist die Hesse-Matrix symmetrisch (die Reihenfolge der Ableitungen kann vertauscht werden).

Untersuchung der Hesse-Matrix

☑ Beispiel vorhanden auf Seite 116.

1. Berechne die kritischen Punkte $\vec{a}$.
2. Ermittle für jeden kritischen Punkt die Definitheit[1] von $(Hf)(\vec{a})$.

$(Hf)(\vec{a})$

- pos. definit
  - Min.
- neg. definit
  - Max.
- indefinit
  - SP.
- pos. semi-definit
  - Min. $\vee$ SP.
- neg. semi-definit
  - Max. $\vee$ SP.

Höhenlinienmethode Sei $\vec{x}_0$ der zu kritische Punkt, dann gilt für das Vorzeichen von $\lim\limits_{\vec{x} \to \vec{x}_0} f(\vec{x}) - f(\vec{x}_0)$:

- Stets positiv: $\vec{x}_0$ ist Minimum.
- Stets negativ: $\vec{x}_0$ ist Maximum.
- Alternierend: $\vec{x}_0$ ist Sattelpunkt.

[1] Siehe dazu das HURWITZ-Kriterium auf Seite 31.

12.2.3. Regressionsanalyse

Ziel Möglichst gute Näherung von Messwerten durch eine Kurve.

Wichtig Die Art der Kurve muss man vorher festlegen.

Für ein Polynom der Art $p(x) = \sum_{i=1}^{n} a_i x^{i-1}$ lassen sich die a_i aus m Messpunkten bestimmen durch das LGS:

$$\left(\begin{array}{cccc|c} m & \sum x_j & \cdots & \sum x_j^n & \sum y_j \\ \sum x_j & \sum x_j^2 & \cdots & \sum x_j^{n+1} & \sum x_j y_j \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum x_j^n & \sum x_j^{n+1} & \cdots & \sum x_j^{2n} & \sum x_j^n y_j \end{array}\right) \qquad | \quad j = 1, \dots, m$$

Die Lösung ergibt die a_i.
Für eine Punktemenge im $\mathbb{R}^2$ ergibt sich mit der gesuchten Lösung $ax + b$:

$$\begin{pmatrix} \sum_{k=1}^{n} x_k^2 & \sum_{k=1}^{n} x_k \\ \sum_{k=1}^{n} x_k & n \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} x_k y_k \\ \sum_{k=1}^{n} y_k \end{pmatrix}$$

12.2.4. Extrema mit Nebenbedingungen

Zuerst die kritischen Punkte $\vec{x}_0$ bestimmen, für die $\operatorname{grad}(f)(\vec{x}_0) = \vec{0}$ gilt, bestimmen. Danach eine der folgenden Methoden anwenden.

12.2.4.1. Untersuchung der Hesse-Matrix

Lagrange-Funktion $\operatorname{grad}(f) + \lambda \cdot \operatorname{grad}(g) = \vec{0}$

Sei $f : \mathbb{R}^n \to \mathbb{R}$ mit k Nebenbedingungen (NB) $g_1(\vec{x}), \dots, g_k(\vec{x})$. Ein Punkt $\vec{a}$, der die Nebenbedingungen $g_1(\vec{x}) = 0, \dots, g_k(\vec{x}) = 0$ erfüllt, erfüllt die *Rangbedingung*, wenn die Gradienten $(\operatorname{grad}(g_1))(\vec{a})$ bis $(\operatorname{grad}(g_k))(\vec{a})$ linear unabhängig sind:

$$\operatorname{rg} \begin{pmatrix} \leftarrow\,\text{--} & (\operatorname{grad}(g_1))(\vec{a}) & \text{--}\rightarrow \\ \leftarrow\,\text{--} & \vdots & \text{--}\rightarrow \\ \leftarrow\,\text{--} & (\operatorname{grad}(g_k))(\vec{a}) & \text{--}\rightarrow \end{pmatrix} = k$$

12.2.4.2. Bestimmung der Kandidaten

1. Punkte, die die Rangbedingung *nicht* erfüllen, aber alle NB, sind Kandidaten.

Teil IV.

Stochastik

13. Kombinatorik

Für vertiefende Erklärungen und Übungsaufgaben mit Lösungen empfiehlt sich [Pap11].

13.1. Grundlagen

Variation oder *Permutation* beschreibt die Möglichkeiten, n Elemente unter Beachtung der Reihenfolge anzuordnen. Man spricht von geordneten Stichproben.

Kombination beschreibt die Möglichkeiten, n Elemente *ohne* Beachtung der Reihenfolge anzuordnen. Man spricht von ungeordneten Stichproben.

Wiederholung wird auch als *Ziehen mit Zurücklegen* beschrieben.

Anzahl der Ziehungen heißt Ordnung.

	mit Wdh.	ohne Wdh.
Kombination	$\mathrm{C}_W(n;k) = \binom{n+k-1}{k}$	$\mathrm{C}(n;k) = \binom{n}{k}$
Variation	$\mathrm{V}_W(n;k) = n^k$	$\mathrm{V}(n;k) = \frac{n!}{(n-k)!}$

Tabelle 13.1.: Kombinatorikfunktionen

13.1.1. Zufallsexperiment

Bedingungen:

- Das Experiment lässt sich unter gleichen Bedingungen beliebig oft wiederholen.
- Es gibt mehrere sich gegenseitig ausschließende Ereignisse.
  - Diese Ereignisse heißen *Elementarereignisse*. Sie werden mit dem kleinen Omega bezeichnet: ω_i.
  - Die Menge aller Elementarereignisse heißt *Ergebnismenge*: Ω.
  - Eine Teilmenge A von Ω heißt *Ereignis*.

  - Die Potenzmenge von Ω heißt *Ereignisraum.*

- Die Ereignisse lassen sich nicht mit Sicherheit voraussagen (*zufallsbedingt*).

13.1.1.1. Laplace-Experiment

- Jedes Elementarereignis hat die Wahrscheinlichkeit $p(\omega_i) = \frac{1}{|\Omega|}$.
- Die Wahrscheinlichkeit eines Ereignisses A ist $P(A) = \frac{|A|}{|\Omega|}$

13.1.1.2. Bernoulli-Experiment

Nur zwei sich gegenseitig ausschließende Ereignisse mit konstanten Wahrscheinlichkeiten.

13.1.2. Definitionen

Gemeinsame Wahrscheinlichkeit $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ (Wahrscheinlichkeit, dass A *oder* B stattfinden.)

Bedingte Wahrscheinlichkeit $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(B)}$ (Wahrscheinlichkeit von A unter der Bedingung, dass B bereits stattgefunden hat.)

Stochastisch unabhängig $P(A \cap B) = P(A) \cdot P(B)$

Totale Wahrscheinlichkeit (Bayessche Formel): $P(B) = \sum_{i=1}^{n} P(A_i) \cdot P(B|A_i)$

Zufallsvariable $X : \omega \mapsto \mathbb{R}$, ist eine Funktion, welche jedem ω eine reelle Zahl zuordnet, welche irgend einen Sinn ergibt (bspw. Gewinn pro Augenzahl beim Würfeln).

Dichtefunktion Anzahl der günstigen Ereignisse über die Zufallsvariable aufgetragen.

Beispiel: X ordnet beim Werfen zweier Würfel den Augenzahlentupeln die Summe zu. Dann wird auf der x-Achse die Augensumme aufgetragen und auf der y-Achse die Anzahl der entsprechenden Ereignisse.

Wahrscheinlichkeitsfunktion $f(x)$ mit $\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 1$, ist die normierte Dichtefunktion.

Verteilungsfunktion $F(x) = P(X \leq x)$, die Aufsummierung der Wahrscheinlichkeiten von $-\infty$ bis zu einem x, bei stetigen X ein Integral.

Erwartungswert Im Durchschnitt erwarteter Wert, $\mu = \mathrm{E}(X) = \sum x_i \cdot f(x_i)$ bzw. $\mu = \mathrm{E}(X) = \int_{-\infty}^{\infty} x \cdot f(x)\,\mathrm{d}x$.

Mittelwert Arithmetisches Mittel, $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Median Mittlerer Wert, d.h., wenn die Messgrößen sortiert[1] sind ($x_1 \leq x_2 \leq \ldots \leq x_n$), ist der Median

$$\widetilde{x} = \begin{cases} x_{\left(\frac{n+1}{2}\right)} & \text{für } n \text{ ungerade} \\ \frac{1}{2}\left(x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)}\right) & \text{für } n \text{ gerade} \end{cases}$$

Modalwert Wert mit der größten Wahrscheinlichkeit bzw. Maximum der Dichtefunktion.

Varianz $\sigma^2 = \mathrm{Var}(X) = \sum(x_i - \mu)^2 f(x_i)$ bzw. $\sigma^2 = \mathrm{Var}(X) = \int_{-\infty}^{\infty}(x - \mu)^2 f(x)\,\mathrm{d}x$

Standardabweichung $\sigma = \sqrt{\mathrm{Var}(X)}$

Lineare Transformation Bekannt sind a, b und X, dann gilt für $Z = aX + b$: $\mu_Z = a\mu_X + b$ und $\sigma_Z^2 = a^2\sigma_X^2$.

Kovarianz $\mathrm{cov}(X;Y) := E(X \cdot Y) - E(X) \cdot E(Y)$

Korrelationskoeffizient $\varrho_{XY} := \frac{\mathrm{cov}(X;Y)}{\sigma_X \cdot \sigma_Y}$

13.1.3. Zentraler Grenzwertsatz

Sind $X_1, \ldots, X_n$ stochastisch unabhängige Zufallsvariablen, welche die *gleiche* Verteilungsfunktion mit *gleichem* μ und σ haben, so gilt für $n \to \infty$:

$$\begin{aligned} P\left(\sum_{i=1}^{n} X_i \leq x\right) &= \Phi\left(\frac{x - n\mu}{\sqrt{n}\sigma}\right) \\ \mathrm{Var}\left(\sum_{i=1}^{n} X_i\right) &= n\sigma^2 \\ \mathrm{E}\left(\sum_{i=1}^{n} X_i\right) &= n\mu \end{aligned}$$

[1] Notfalls müssen die Messgrößen noch per Hand sortiert werden.

14. Verteilungen

14.1. Binomialverteilung

(Bin, diskret, mit Wiederholung)

Bei Bernoulli-Experimenten. Mit n als Anzahl der Versuche und $p = P(A)$ ist die Wahrscheinlichkeit, dass A in den n Versuchen k-mal auftritt:

$$\begin{aligned} f(k) &= \binom{n}{k} p^k (1-p)^{n-k} \\ \mu &= np \\ \sigma^2 &= np(1-p) \end{aligned}$$

14.1.1. Grenzwertsatz von Moivre-Laplace

Sei X eine binomialverteilte Zufallsvariable, dann gilt:

$$\begin{aligned} Z = \frac{X-\mu}{\sigma} &= \frac{X-np}{\sqrt{np\,(1-p)}} \\ \lim_{n\to\infty} Z &= \Phi(u) \end{aligned}$$

14.2. Geometrische Verteilung

(diskret, mit Wiederholung)

Wahrscheinlichkeit, dass nach k Versuchen ein Ereignis mit Wahrscheinlichkeit p eintritt:

$$\begin{aligned} f(k) &= (1-p)^k \cdot p \\ \mu &= \frac{1-p}{p} \\ \sigma^2 &= \frac{1-p}{p^2} \end{aligned}$$

14.3. Hypergeometrische Verteilung

(Hyp, diskret, ohne Wiederholung)

In einer Urne sind N Kugeln, davon M weiße und $N - M$ schwarze. Mit n als Anzahl der Versuche ist die Wahrscheinlichkeit, dass k weiße Kugeln gezogen werden:

$$\begin{aligned} f(k) &= \frac{\binom{M}{k} \cdot \binom{N-M}{n-k}}{\binom{N}{n}} \\ \mu &= n\frac{M}{N} \\ \sigma^2 &= \frac{nM(N-M)(N-n)}{N^2(N-1)} \end{aligned}$$

14.4. Poisson-Verteilung

(Ps, diskret, mit Wiederholung, bei seltenen Ereignissen)

$$\begin{aligned} f(k) &= \frac{\mu^k}{k!} \cdot e^{-\mu} \\ \sigma^2 &= \mu \end{aligned}$$

14.5. Normalverteilung

(N, stetig, mit Wiederholung)

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \\ &= \varphi\left(\frac{x-\mu}{\sigma}\right) \\ F(x) &= \Phi\left(\frac{x-\mu}{\sigma}\right) \end{aligned}$$

Falls $\mu = 0$ und $\sigma = 1$, dann handelt es sich um eine Standardnormalverteilung mit:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}x^2\right)$$

Jede normalverteilte Zufallsvariable X lässt sich in die Standardnormalverteilung U umformen mit:

$$u = \frac{x-\mu}{\sigma}$$

Ist eine Zufallsvariable normalverteilt, so schreibt man $X \sim \mathrm{N}(\mu, \sigma)$, bei Standardnormalverteilung $X \sim \mathrm{N}(0, 1)$.

14.6. Exponentialverteilung

(Exp, stetig, ohne Wiederholung)

$$
\begin{aligned}
f(x) &= \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \\
F(x) &= \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \\
\mu &= \frac{1}{\lambda} \\
\sigma^2 &= \frac{1}{\lambda^2}
\end{aligned}
$$

14.7. χ^2-Verteilung

Die Chi-Quadrat-Verteilung gibt die Verteilung der Varianz einer Stichprobe an.

Seien $X_1, \ldots, X_n$ stochastisch unabhängige Zufallsvariablen mit $X_i \sim \mathrm{N}(0; 1)$, dann gilt für $\chi^2 = X_1^2 + \ldots + X_n^2$ mit n als Freiheitsgrad:

$$
\begin{aligned}
f(z) &= \begin{cases} A_n \cdot z^{\left(\frac{n-2}{2}\right)} \cdot e^{-\frac{z}{2}} & z \geq 0 \\ 0 & z < 0 \end{cases} \\
F(z) &= A_n \cdot \int_0^z u^{\left(\frac{n-2}{2}\right)} \cdot e^{-\frac{u}{2}} \, \mathrm{d}u \\
\mu &= n \\
\sigma^2 &= 2n
\end{aligned}
$$

Die Normierungskonstante A_n berechnet sich als:

$$
A_n = \frac{1}{2^{\frac{n}{2}} \cdot \Gamma\left(\frac{n}{2}\right)}
$$

14.7.1. Gamma-Funktion

$\alpha > 0,\ n \in \mathbb{N} \backslash \{0\}$

$$\begin{aligned}
\Gamma(\alpha) &= \int_0^\infty t^{\alpha-1} \cdot e^{-t}\, \mathrm{d}t \\
\Gamma\left(\frac{1}{2}\right) &= \sqrt{\pi} \\
\Gamma(1) &= 1 \\
\Gamma(\alpha+1) &= \alpha \cdot \Gamma(\alpha) \\
\Gamma(n+1) &= n! \\
\Gamma\left(n+\frac{1}{2}\right) &= \frac{1 \cdot 3 \cdot 5 \cdots (2n-1)}{2^n}\sqrt{\pi}
\end{aligned}$$

14.8. t-Verteilung nach Student

Schätzfunktion für die Verteilung der Abweichung vom Stichprobenmittelwert zum realen Mittelwert bei normalverteilten Daten.

14.9. Mehrere Veränderliche

- $P(a_1 \leq X \leq a_2; b_1 \leq Y \leq b_2) = F(a_1; b_1) + F(a_2; b_2) - F(a_1; b_2) - F(a_2; b_1)$

Randverteilung

$$\begin{aligned}
f_1(x) &= \int_{y=-\infty}^{\infty} f(x; y)\, \mathrm{d}y \\
f_2(y) &= \int_{x=-\infty}^{\infty} f(x; y)\, \mathrm{d}x
\end{aligned}$$

Stochastisch unabhängig $F(x; y) = F_1(x) \cdot F_2(y)$

Additionssatz

$$\begin{aligned}
\mathrm{E}\left(\sum X_n\right) &= \sum \mathrm{E}(X_n) \\
\mathrm{Var}\left(\sum X_n\right) &= \sum \mathrm{Var}(X_n)
\end{aligned}$$

Multiplikationssatz

$$\mathrm{E}\left(\prod X_n\right) = \prod \mathrm{E}(X_n)$$

15. Statistik/Schätzungen

Empirische Varianz $s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2$

Empirische Standardabweichung $s = \sqrt{s^2}$

Empirischer Variantionskoeffizient $V := \frac{s}{\overline{x}}$

Empirische Kovarianz $\operatorname{cov}(X, Y) = s_{XY} := \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})$

Empirischer Korrelationskoeffizient $r_{XY} := \frac{\operatorname{cov}(X,Y)}{s_X \cdot s_Y}$

Regressionsgerade $y = \hat{a} + \hat{b}x$ mit $\hat{b} = \frac{s_{XY}}{s_X^2}$ und $\hat{a} = \overline{y} - \hat{b}\overline{x}$

Maximum-Likelihood-Methode

Sei f eine Verteilungsfunktion sowie ϑ ein zu bestimmender Parameter, dann gilt zur Bestimmung von ϑ:

$$\begin{aligned} L = L(\vartheta) &:= \prod_{i=1}^{n} f(x_i; \vartheta) \\ \frac{\partial L}{\partial \vartheta} &\overset{!}{=} 0 \\ \frac{\partial^2 L}{\partial^2 \vartheta} &< 0 \end{aligned}$$

Evtl. bietet sich die Logarithmierung von L an: $L^* := \log(L)$.

15.1. Konfidenzintervalle

Konfidenz-/Vertrauensniveau $\gamma = 1 - \alpha$

Normalverteilung: μ unbekannt, σ^2 bekannt

(u: Quantil der Normalverteilung)

$$\overline{x} - u_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \overline{x} - u_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}$$

Normalverteilung: μ unbekannt, σ^2 unbekannt

(u: Quantil der t-Verteilung nach Student)

$$\overline{x} - u_{1-\frac{\alpha}{2};n-1} \frac{s}{\sqrt{n}} \leq \mu \leq \overline{x} - u_{1-\frac{\alpha}{2};n-1} \frac{s}{\sqrt{n}}$$

Normalverteilung: σ^2 unbekannt

(u: Quantil der χ^2-Verteilung)

$$(n-1)\frac{s^2}{u_{1-\frac{\alpha}{2};2}} \leq \sigma^2 \leq (n-1)\frac{s^2}{u_{1-\frac{\alpha}{2};1}}$$

Binomialverteilung: p unbekannt

(u: Quantil der Normalverteilung)

$$\hat{p} - \frac{c}{n}\sqrt{n\hat{p}(1-p)} \leq p \leq \hat{p} + \frac{c}{n}\sqrt{n\hat{p}(1-p)}$$

16. Übersicht

	mit Wdh.	ohne Wdh.
stetig	N	Exp
diskret	Bin oder Ps	Hyp

Tabelle 16.1.: Entscheidungshilfe für Wahrscheinlichkeitsfunktionen

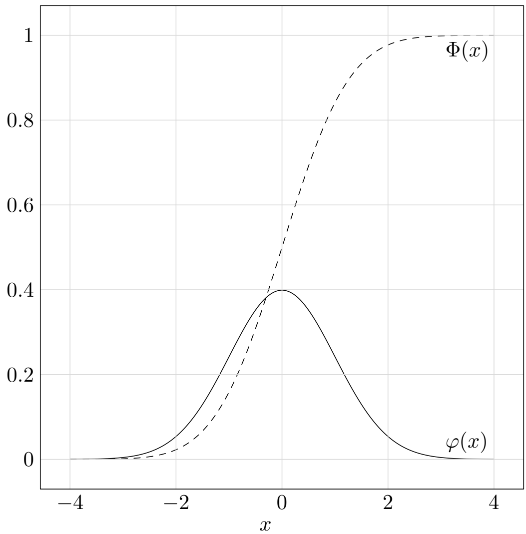


Abbildung 16.0.1.: Standardnormalverteilung

		Näherung	
		Bin	Ps
Verteilung	Bin		$np \leq 10n$ $n \geq 1500p$
			$\mathrm{Ps}(\mu = np)$
	Hyp	$\frac{M}{N} \in [0.1, 0.9]$ $n < 0.05N$ $n > 10$	$\frac{M}{N} \notin]0.1, 0.9[$
		$\mathrm{Bin}\left(p = \frac{M}{N}\right)$	$\mathrm{Ps}\left(\mu = n\frac{M}{N}\right)$

		N
Verteilung	Bin	$np(1-p) > 9$
		$\mathrm{N}\left(\mu = np, \sigma = \sqrt{np(1-p)}\right)$
	Hyp	$\frac{M}{N} \in]0.1, 0.9[$ $n < 0.05N$ $n > 30$
		$\mathrm{N}\left(\mu = n\frac{M}{N}, \sigma = \sqrt{n\frac{M}{N}\left(1 - \frac{M}{N}\right)\frac{N-n}{N-1}}\right)$
	Ps	$\mu > 10$
		$\mathrm{N}\left(\mu, \sigma = \sqrt{\mu}\right)$

Tabelle 16.2.: Näherungen diskreter Wahrscheinlichkeitsfunktionen

Teil V.

IT Grundlagen

17. Zahlendarstellungen

Byte 8 Bit

Vorzeichenlos Wertebereich bei n Stellen: 0 bis $2^n - 1$.

Vorzeichenbit Das höchste Bit wird als Vorzeichen gewertet[1].

- Wertebereich bei n Stellen: -$\left(2^{n-1} - 1\right)$ bis $2^{n-1} - 1$.

Die Subtraktion kann nicht auf die Addition zurückgeführt werden, daher ist ein zusätzliches Steuerwerk zur Subtraktion nötig.

Einerkomplement *Logische Komplementbildung*

Erweiterung der einfachen Vorzeichenbitdarstellung um Subtraktionen auf Additionen zurückführen zu können.

- Wertebereich bei n Stellen: -$\left(2^{n-1} - 1\right)$ bis $2^{n-1} - 1$.
- Das höchstwertige Bit ist gesetzt, falls die Zahl negativ ist.
- Negation durch binäre Komplementbildung (NOT).

Addition und Subtraktion benötigen keine besondere Vorzeichenbehandlung, allerdings ist die Berechnung komplexer als beim Zweierkomplement, da der Übertrag bei der Addition zum Ergebnis hinzu addiert werden muss.

Siehe [Wik12c].

Zweierkomplement *Arithmetische Komplementbildung*

- Erweiterung des Einerkomplements.
- Wertebereich bei n Stellen: -2^{n-1} bis $2^{n-1} - 1$.
- Negation durch binäre Komplementbildung (NOT) und Addition von 1.
- Addition und Subtraktion benötigen keine besondere Vorzeichenbehandlung.

[1] Daraus ergibt sich der Umstand, dass zwei Darstellungen für die 0 möglich sind: $+0$ und -0.

Dadurch werden das Problem der doppelten Null und das Problem der Addition des Übertrags behoben[2].

Siehe [Wik12g].

Festkommazahlen Bestehen aus n Vorkommabits und m Nachkommabits.

Siehe [Wik12d].

Gleitkommazahlen (IEEE 754) Der *single*-Datentyp ist 32 Bit groß. Mathematische Darstellung: $(-1)^s \cdot m \cdot 2^e$. Der *Exponent* e hat den Wertebereich -126 bis 127 (wobei $e = E - B$ gilt, und E im Datenwort gespeichert wird – B ist der sog. *Bias* und beträgt 127). Die *Mantisse* m belegt 23 Bit. Die Darstellung ist *normalisiert*, d. h. der Exponent wird so berechnet, dass genau eine 1 vor dem Komma steht – somit kann dieses eine Bit gespart werden (*Hidden Bit*) und steht dem *Vorzeichenbit* s zur Verfügung.

Berechnungen sind höllisch kompliziert. Die Zahlen sind logarithmisch verteilt (zwischen 1 und 10 liegen genauso viele Werte wie zwischen 10 und 100). Die ersten Prozessoren besaßen noch keine eingebaute FPU, diese konnte jedoch nachträglich als Zusatzprozessor eingebaut werden.

Siehe [Wik12f, Wik12e].

Exponent e	Mantisse m	Bedeutung
-127	$= 0$	± 0
-127	$\neq 0$	$0.m$
$-126 \leq e \leq 127$	$\neq 0$	$1.m$
-128	$= 0$	$\pm\infty$
-128	$\neq 0$	NaN

Tabelle 17.1.: Sonderwerte IEEE 754

17.1. Stellenwertsystem

Aus einem Eingabealphabet Σ_E mit der Basis $b = |\Sigma_E|$ und den Stellen σ_i lässt sich ein Stellenwertsystem nach dem Schema $\sum_{i=0}^{n} b^i \sigma_i$ aufbauen. Das Eingabealphabet Σ_E wird bijektiv auf die Werte $0 \ldots |\Sigma_E| - 1$ abgebildet.

Als Symbol für das Eingabealphabet wird i.d.R. das große Sigma Σ verwendet. Schreibt man Zahlen auf, die eine andere Basis als 10 haben, so setzt man die Zahl in Klammern und schreibt die Basis als Index: $(1111)_2 = 15$.

[2] Daraus ergibt sich allerdings der Umstand, dass die kleinste darstellbare negative Zahl -2^{n-1} bei Negation wieder sich selbst ergibt.

Größe	Wertebereich	C++	Java
8 Bit	-128 ... +127	`char`	`byte`
16 Bit	-32 768 ... +32 767	`short`	`short`
32 Bit	-2 147 483 648 ... +2 147 483 647	`int`	`int`
64 Bit	ca. $\pm 9 \cdot 10^{18}$	`long`	`long`
32 Bit	$\pm 1.4 \cdot 10^{-45} \ldots \pm 3.4 \cdot 10^{38}$	`float`	`float`
64 Bit	$\pm 4.9 \cdot 10^{-324} \ldots \pm 1.7 \cdot 10^{308}$	`double`	`double`

Tabelle 17.2.: Wertebereiche Datentypen

Beispiel Dezimalsystem mit $\Sigma_E = \{0, \ldots, 9\}$ und $b = 10$: $5678 = \sum_{i=0}^{3} \sigma_i 10^i = 10^0 \cdot 8 + 10^1 \cdot 7 + \cdot 10^2 \cdot 6 + 10^3 \cdot 5$.

Beispiel Hexadezimalsystem mit $\Sigma_E = \{0, \ldots, 9, A, \ldots, F\}$ und $b = 16$, wobei $A \mapsto 10, \ldots, F \mapsto 15$: $(123)_{16} = 3 \cdot 16^0 + 2 \cdot 16^1 + 1 \cdot 16^2 = 3 + 32 + 256 = 291$.

18. Codierungen

18.1. ASCII

ASCII steht für *American Standard Code for Information Interchange*.
Wichtige Zeichen:

- 0x00 = NUL
- $0x0A$ = LF (Line Feed)
- $0x0D$ = CR (Carriage Return)
- 0x20 = SP (Space/Leerzeichen)
- 0x30 – 0x39 (Ziffern 0 bis 9)
- 0x40 (@)
- $0x41 - 0x5A$ (Buchstaben A bis Z)
- $0x61 - 0x7A$ (Buchstaben a bis z)

18.2. Unicode

Zielsetzung Ein eindeutige Code für jedes Zeichen weltweit.

Aufbau 16 Bit Wortlänge, die ersten 128 Zeichen entsprechen dem ASCII-Code.

18.3. Unicode Transformation Format (UTF)

Zielsetzung Verlustfreie Kompression von Unicode.

UTF-8 Je Zeichen 1 – 4 Bytes, variable Länge.

UTF-16 Je Zeichen 2 – 4 Bytes, variable Länge.

UTF-32 Je Zeichen 4 Bytes, feste Länge.

Codierung von UTF-8

Das erste Byte beginnt mit n Einsen, gefolgt von einer 0. Die Anzahl n der Einsen gibt an, aus wie vielen Bytes das Zeichen besteht. Ist $n > 0$, so beginnen die folgenden $n-1$ Bytes jeweils mit den Bits $(10)_2$. Die restlichen Bits stehen jeweils für die Codierung des Zeichens zur Verfügung.

Beispiel $1110\,xxxx \mid 10xx\,xxxx \mid 10xx\,xxxx$: Es ist $n = 3$, und es stehen 16 Bits zur Zeichencodierung zur Verfügung.

18.4. Hamming-Code

Motivation Erkennung und Korrektur von Übertragungsfehlern.

Prinzip Ein Datenwort aus N Bits wird in A Datenbits und P Korrektur-/Paritätsbits mit $N = A + P$ unterteilt.

Beispiel für $A = 4$ und $P = 3$: Errechnung der Korrekturbits mit

$$\begin{aligned} e_1 &= a_1 + a_2 + a_3 \bmod 2 \\ e_2 &= a_2 + a_3 + a_4 \bmod 2 \\ e_3 &= a_1 + a_2 + a_4 \bmod 2 \end{aligned}$$

Wenn $\forall i : p_i = e_i$ erfüllt ist, so sind die Daten korrekt übermittelt worden.

Siehe [Wik04, VKKS11]

18.5. EAN (European Article Number)

Aufbau Besteht aus 13 Ziffern

2 Ziffern Ländercode, 5 Ziffern Hersteller, 5 Ziffern Produktnummer, 1 Prüfziffer

Prüfsummenberechnung Jede Ziffer mit geradem Index wird zunächst mit 3 multipliziert, dann werden alle Zahlen aufaddiert. Die Prüfziffer der Summe $s = \sum_{i=0}^{12} a_i \cdot (1 + 2 \cdot (i \bmod 2))$ ist dann $p = (-s) \bmod 10$ (somit ist $(s + p) \bmod 10 = 0$).

18.6. Hash-Summen

Motivation Erkennung von Übertragungsfehlern.

Prinzip Mit komplexen mathematischen Verfahren wird dafür gesorgt, dass die Veränderung eines einzelnen Bits im Eingabedatenstrom mit großer Wahrscheinlichkeit eine große Veränderung des Ausgabeworts verursacht.

Bekannte Verfahren MD5 (Message Digest Algorithm 5), SHA-1 (Secure Hash Algorithm), CRC-32 (Cyclic Redundancy Check).

18.7. Huffman-Codierung

Zielsetzung Verlustfreie Kompression durch Codierung von häufig vorkommenden Zeichen mit kurzen Codes und selten vorkommenden Zeichen mit langen Codes.

Prinzip

Schritt 1 Zählen der Häufigkeiten aller Zeichen im Eingabestrom. Kombination aus Häufigkeit und Zeichen als Knoten notieren.

Schritt 2 Bilde einen neuen Knoten, der die zwei geringsten Häufigkeiten als Unterknoten hat. Der Wert des neuen Knotens ist die Summe der beiden Unterknoten. Wiederhole diesen Schritt mit allen „Top-Level-Knoten“, bis nur noch ein einziger Wurzelknoten übrig ist.

Schritt 3 Kodierung des Pfades von der Wurzel zu den Zeichen, wobei Links und Rechts eindeutig als 0 und 1 (oder andersherum) kodiert werden.

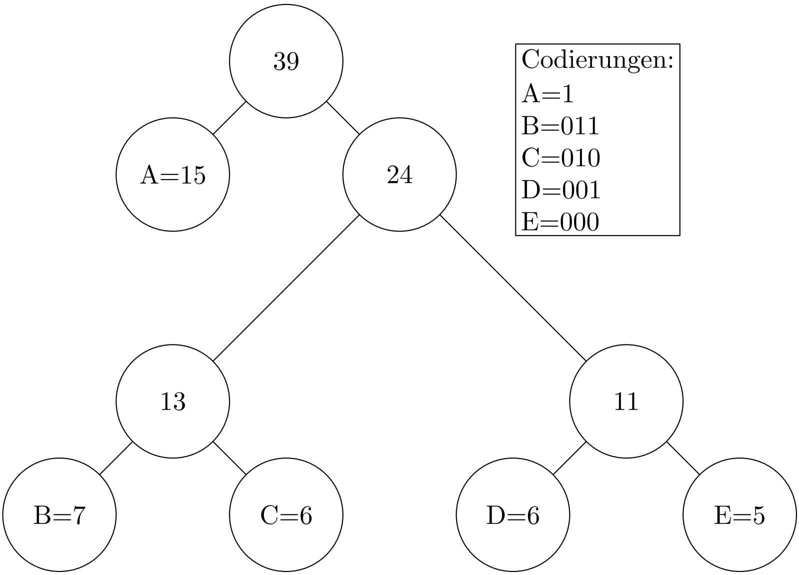


Abbildung 18.7.1.: Huffman-Codierung

19. Von-Neumann Rechnerarchitektur

19.1. Komponenten

ALU Arithmetic Logic Unit – führt Rechenoperationen und logische Verknüpfungen durch.

Control Unit Steuer-/Leitwerk – interpretiert Programmanweisungen und koordiniert Datentransfers und ALU.

CPU Central Processing Unit – besteht aus ALU und Control Unit.

RAM Random Access Memory – Lese- und Schreibzugriffe möglich, für Programme und Daten genutzt.

ROM Read Only Memory – Nur Lesezugriff möglich.

Memory Speicherwerk – Speichert Programme und Daten, besteht aus RAM und ROM.

Bus-System Besteht aus Datenbus zur Übertragung von Daten zwischen CPU und Speicher und Adressbus zur Bestimmung der benutzten Daten im Speicher.

I/O-Unit Steuert das Bus-System.

Kernpunkte:

- Rechnerstruktur ist unabhängig vom bearbeiteten Problem.
- Programme und Daten sind im selben Speicher und können modifiziert werden.
- Der Hauptspeicher besteht aus adressierbaren Zellen gleicher Größe.
- Alle Komponenten sind der Control Unit unterworfen.

19.1.1. CPU

Ausführung von arithmetischen und logischen Operationen. Zwischenspeicherung mittels sog. „Registern“:

- Akkumulator-Register *A* (Intel-Architektur: `al` (8 Bit), `ah` (8 Bit), `ax` (16 Bit), `eax` (32 Bit), `rax` (64 Bit))
- Operanden-Register *MBR* (Memory Buffer Register)
- Übertrags-Register *L* (Latch)[1]

19.1.2. Control Unit

Überwachung der Programmdurchführung:

1. Adresse des ersten Befehls in den *Befehls-Zähler* (PC: Program Counter) laden.

   Intel-Architektur: *Instruction Pointer* `ip` *(16 Bit),* `eip` *(32 Bit),* `rip` *(64 Bit).*

2. Control Unit holt den Befehl in das *Befehls-Register.*

   Intel-Architektur: intern im Prozessor.

3. Befehlszähler um 1 erhöhen.

   Intel-Architektur: Da die Befehlslängen variabel sind, wird der Instruction Pointer entsprechend verändert.

4. Ausführung des Befehls.

5. Weiter zu 2.

19.1.3. RAM

SRAM Static RAM: Schneller Befehlsspeicher (Cache).

Größenordnung des SRAM bei ca. 4 bis 254 KB, Zugriffszeit entspricht i.d.R. der Taktfrequenz des Prozessors. Sehr teuer. Als *1*st *Level Cache* direkt im Prozessor integriert.

DRAM Dynamic RAM: Eigentlicher Arbeitsspeicher.

Größere Kapazität, aber auch höhere Zugriffszeiten, siehe 19.2 auf der nächsten Seite. Wesentlich kostengünstiger als SRAM.

[1] Auch als *Carry Bit* bekannt.

19.1.4. Bus

Datenbus Bidirektional.

Überträgt die Daten zwischen den Komponenten im Computer. Die Computerarchitektur (16, 32 oder 64 Bit) gibt i.d.R. Auskunft darüber, wie viele Bits der Datenbus parallel übertragen kann.

Adressbus Unidirektional.

Gibt die Speicheradressen an, auf die über den Datenbus zugegriffen wird. Die Computerarchitektur gibt die Breite des Busses an, bei n Leitungen kann man auf 2^n Speicherstellen zugreifen (daher ist die 32-Bit Architektur auf 2^{32} Bytes bzw. 4 GB Arbeitsspeicher begrenzt).

Steuer-/Kontrollbus Koordiniert Daten- und Adressbus.

19.2. Flaschenhals

Während die Ausführungszeit für einen Befehl relativ kurz ist, dauert das Lesen und Schreiben von Speicher i.d.R. relativ lange. Die CPU muss also auf den Speicher warten.

Die Kommunikation zwischen Speicher und CPU heißt Von-Neumann-Flaschenhals.

19.2.1. Erweiterungen zur Umgehung des Problems

- Hohe Parallelität zwischen Verarbeitungs- und Ein-/Ausgabevorgängen.

  Stichwort: DMA (Direct Memory Access). Dies ist i.d.R. ein eigener Baustein im Rechner, der ohne Belastung der CPU parallel Daten kopieren kann.

- Große Registeranzahl: Ermöglicht, dass Zwischenwerte in der CPU gehalten werden können und somit nicht in den Speicher geschrieben werden müssen.

- Caches: Kleiner, aber schneller CPU-interner Zwischenspeicher. Basiert auf der Idee, dass Speicherzugriffe i.d.R. „räumlich nahe“ stattfinden.

  Ist eine Kopie eines Arbeitsspeicherbereichs, der nur bei Bedarf zurückgeschrieben wird.

19.2.2. Harvard-Architektur

Idee Aufteilung von Befehls- und Datenspeicher, sowie Trennung der Busse für Befehle und Daten.

Ziele Parallele Bereitstellung von Befehlen und Daten sowie erhöhte Sicherheit, da Befehle sich nicht mehr selbst modifizieren können (Überlaufschutz).

19.3. RISC/CISC

RISC Reduced Instruction Set Computer

Verzichtet auf komplexe Befehle, einzelne Befehle sind fest verdrahtet (in Hardware gegossen). I.d.R. sind viele Register vorhanden.

- Die Abarbeitungszeit pro Befehl ist nahezu gleich.

Pro Bessere Optimierung durch Compiler.

Kontra Größere Programmdateien wegen gleich großer Befehle.

CISC Complex Instruction Set Computer

- Komplexe Befehle, Befehlssatz meist in Form von Microcode (quasi Software für einen Befehlsprozessor im Prozessor). Enthält weniger Register.
- Die Abarbeitungszeit pro Befehl ist unterschiedlich.

Pro Weniger Hauptspeicherbedarf durch die komplexen Befehle.

Kontra Die komplexen Befehle werden häufig nicht voll ausgenutzt.

20. Betriebssysteme

20.1. Definition

DIN 44330 Ein Betriebssystem umfasst die Programme eines digitalen Rechnersystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechnersystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen.

Das Betriebssystem liegt als Softwareschicht zwischen Hardware und Anwendungen. Die Komplexität der Hardware wird somit vor den Anwendungen verborgen (HAL, Hardware Abstraction Layer).

20.2. Anforderungen

Hohe Zuverlässigkeit Korrektheit, Sicherheit, Verfügbarkeit, Robustheit, Schutz von Benutzerdaten

Hohe Leistung Gute Auslastung der Systemressourcen, kleiner Verwaltungsaufwand, hoher Durchsatz, kurze Reaktionszeit

Hohe Benutzerfreundlichkeit Angepasste Funktionalität, einfache Benutzerschnittstelle, Hilfestellung

Einfache Wartbarkeit Einfache Upgrades, einfache Erweiterbarkeit, Portierbarkeit

Geringe Kosten Sowohl für Anschaffung als auch Betrieb.

20.3. Aufgaben

- Prozessverwaltung (Task Management, Task Scheduling)
  - Zuteilung von Ressourcen
  - Prozess Besteht aus Programmcode und Umgebung (Kontext). Der Kontext beinhaltet:
    * den privaten Adressraum
    * globale Variablen

    * geöffnete Streams (Dateien, Sockets, ...)
    * abhängige Prozesse
  - Multitasking
    * präemptives Multitasking: das Betriebssystem entscheidet wann welcher Prozess für wie lange ausgeführt wird
    * kooperatives Multitasking: der Prozess muss die Kontrolle selbst abgeben

- Speicherverwaltung
- Betriebsmittelverwaltung (Konfliktverwaltung)
  - Aktive, zeitlich aufteilbare (CPU)
  - Passive, exklusiv nutzbare (I/O-Geräte)
  - Passive, räumlich aufteilbare (RAM)
- Dateiverwaltung
  - Überprüfung auf Existenz, Zugriffs-/Rechteverwaltung, Lese-/Schreibkonflikt
- Rechteverwaltung
- ...

20.4. Typen

Stapelverarbeitungssysteme Auch Batch-Systeme genannt.

Das System arbeitet nicht interaktiv, sondern nimmt nur Aufträge an und legt diese in einer Job-Queue ab. Die Queue wird vollkommen autonom abgearbeitet, die Ausgabe erfolgt über Dateien oder sonstige Geräte (Monitor nicht sinnvoll).

Interaktive Systeme/Dialogsysteme Benutzer steuert die Programme bspw. über Maus/Tastatur. Ausgabe der Programmergebnisse kann auf beliebigen Geräten erfolgen.

Echtzeitsysteme Meist mit Sensoren eingesetzt. Hat harte Anforderungen an die Reaktionsfähigkeit. Beispiel: Airbag-System.

20.5. Task Scheduler

Ein Prozess kann sich in folgenden Zuständen befinden:

rechnend Der Prozessor ist dem Prozess zugeteilt.

blockiert Der Prozess wartet auf ein externes Ereignis.

rechenbereit Der Prozess ist ausführbar, aber der Prozessor führt einen anderen Prozess aus.

Ein guter Scheduling Algorithmus muss folgende Anforderungen erfüllen:

Fairness Jeder Prozess erhält einen gerechten Anteil der Ressourcen (auch Berücksichtigung der *Prozessprioritäten*).

Effizienz Die Ressourcen sind möglichst gleichmäßig ausgelastet.

20.5.1. Batch-Systeme

- Es sollte nicht vorkommen, dass die CPU Leertakte hat.
- Die Anzahl der bearbeiteten Aufgaben pro Zeit-Einheit sollte maximal sein.
- Die Zeit von der Einreihung eines Jobs in die Job-Queue bis zur Fertigstellung sollte minimal sein.

20.5.2. Dialogsysteme

- Der Benutzer sollte möglichst zeitnah Rückmeldung auf Aktionen bekommen.
- Prozesse, die Interaktion erfordern, sollten vor anderen Prozessen bevorzugt werden.
- Die Antwortzeit der Prozesse sollte mit der Benutzererwartung korrelieren.
- Aus Benutzersicht einfache Aufgaben sollten schneller erledigt werden als andere.

20.5.3. Echtzeitsysteme

- Die Zeitfenster der Prozesse müssen eingehalten werden.
- Das System muss vorhersagbar/deterministisch reagieren.

20.5.4. Strategien

FIFO First Come, First Serve (FCFS) bzw. First In, First Out (FIFO)

Prozesse werden der Reihenfolge ihres Starts nach bearbeitet. Ein Prozesswechsel findet nur statt, wenn der aktuelle Prozess sich beendet oder zu warten beginnt.

SJF Shortest Job First

Wird auf Batch-Systemen angewendet. Jobs werden nach geschätzter Ausführungszeit aufsteigend sortiert verarbeitet. Nachteil: Große Jobs bekommen möglicherweise nie Rechenzeit wenn sich immer wieder kleinere Jobs davor drängeln

Round-Robin Prozessen wird der Reihe nach eine (gleichgroße) Zeitscheibe auf der CPU zugeteilt.

20.6. Speicherverwaltung

... ermöglicht den Prozessen Zugriff auf den Arbeitsspeicher. Es wird zwischen physischer und virtueller Speicherverwaltung unterschieden. Die Speicherverwaltung muss des Weiteren die Speicherbereiche verschiedener Prozesse voneinander abkapseln (Stichwort Viren).

20.6.1. Physische Speicherverwaltung

- Die Summe des von den gleichzeitig geladenen Prozessen verwendeten Arbeitsspeichers kann nicht größer sein als der physikalisch vorhandene Speicher.
- Es kann *Swapping* benutzt werden damit mehr Prozesse quasi-parallel betrieben werden können.
- Probleme:
  - Fragmentierung des Speichers
  - Suche nach freien Blöcken mittels Belegungstabelle sehr aufwändig

Gebräuchliche Varianten um einen freien Block zu finden sind:

First Fit Verwendet den erstbesten freien Speicherblock. Ist schnell, verkleinert aber evtl. große freie Blöcke, so dass anschließend evtl. ein großer Prozess nicht mehr geladen werden kann.

Next Fit Wie First Fit, startet die Suche aber nicht am Anfang des Speichers, sondern dort, wo der letzte Prozess eingefügt wurde.

Best Fit Sucht den kleinsten passenden freien Block. Verteilt den Speicher gut, ist aber sehr langsam.

20.6.2. Virtuelle Speicherverwaltung

... behebt die Probleme physischer Speicherverwaltung.

Jedem Prozess wird ein logisch zusammenhängender Speicherbereich zur Verfügung gestellt, tatsächlich kann der physische Speicher aber fragmentiert sein (ähnlich RAID, siehe 22.5 auf Seite 84). Die Gesamtheit aller virtuellen Adressen heißt *virtueller Adressraum*, der physische Speicher ist in sog. *Pages* unterteilt. Die virtuellen Pages werden auf die physischen Pages abgebildet.

Greift ein Prozess auf eine virtuelle Page zu, so versucht der Prozessor mit Hilfe der *Pagetable* auf die abgebildete physische Page zuzugreifen. Schlägt der Zugriff fehl, löst der Prozessor einen *Page Fault* aus. Dieser wird i.d.R. dazu genutzt, ausgelagerte physische Pages von der Festplatte wieder in den Arbeitsspeicher zu laden.

20.6.2.1. Auslagerungsstrategien

FIFO First In, First Out

Die am längsten im Speicher liegende Page wird ausgelagert.

LRU/LFU Least Recently/Frequently Used

Am am längsten nicht mehr benutzte/am wenigsten benutzte Page wird ausgelagert.

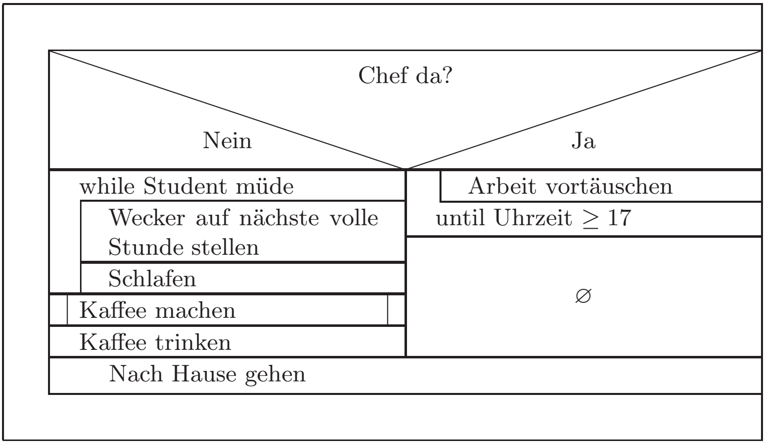
Unversehrtheit Es werden die Pages ausgelagert, welche sich im Arbeitsspeicher nicht verändert haben und somit keine teure Schreiboperation auf der Festplatte erfordern.

NRU Not Recently Used

Eine Page wird ausgelagert, wenn sie innerhalb eines Zeitfensters weder gelesen noch beschrieben wurde.

21. Grafische Algorithmendarstellungen

21.1. Struktogramme



LaTeX Code:

```
\begin{struktogramm}(100,66)
\forever
    \ifthenelse{3}{3}{Chef da?}{Nein}{Ja}
        \while{while Student m"ude}
            \assign{Wecker auf n"achste volle Stunde stellen}
            \assign{Schlafen}
        \whileend
        \sub{Kaffee machen}
        \assign{Kaffee trinken}
    \change
        \until{until Uhrzeit $\geq 17$}
            \assign{Arbeit vort"auschen}
        \untilend
    \ifend
    \assign{Nach Hause gehen}
\foreverend
\end{struktogramm}
```

21.2. Flussdiagramme

Elemente eines Flussdiagramms:

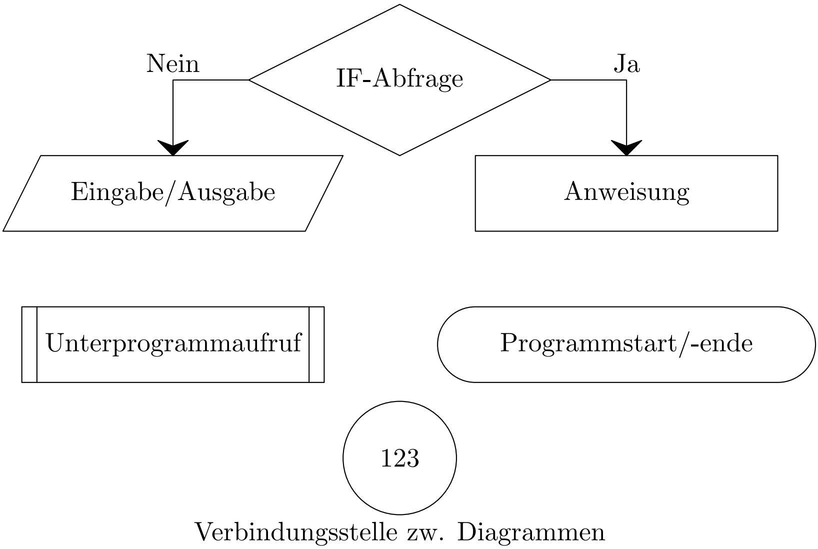


Abbildung 21.2.1.: Elemente eines Flussdiagramms

22. Speichermedien

Magnetisch Magnetkarten/-bänder, Disketten, Magnetplatte

Optisch Mikrofilme, optische Speicherkarten/-platten/-bänder

Elektronisch Chipkarten, Halbleiterspeicher

22.1. Diskette

Zugriffszeit 90-100 ms

Datenübertragungsrate 60 KB/s bei 300 U/min.

Typische Größe 1.44 MB

22.2. Magnetplatte

Zugriffszeit 3.5 ms

Datenübertragungsrate 150 MB/s

Typische Größe 250 bis 1 000 GB

22.3. Solid State Disc (SSD)

Zugriffszeit 0.3 ms

Datenübertragungsrate 480 MB/s

Typische Größe 960 GB

Erhöhte Stoßfestigkeit, erhöhte Schreib-/Lesegeschwindigkeit, geringerer Energieverbrauch.

Besteht aus NAND-Flash. Bauweisen sind SLC (Single Level Cells, ein Bit pro Zelle) und MLC (Multi Level Cells, zwei Bit pro Zelle).

Nachteil: Zellen sind nur begrenzt beschreibbar, üblicherweise ca. 3 000 bis 100 000 mal.

22.4. Compact Disc

Zugriffszeit 0.3 ms

Datenübertragungsrate 1 228 MBit/s

Typische Größe 540 bis 900 MB

- Oberfläche besteht aus „Pits“ (Hügel) und „Lands“ (Täler).
- Ein Wechsel zwischen Pit und Land entspricht einer binären 1. Findet kein Wechsel statt, so ist das eine binäre 0.
- Technische Randbedingung: Mindestens 2 und höchstens 11 Nullen zwischen zwei Einsen.
- Daher sind 8-Bit-Codes nicht zu benutzen, es werden stattdessen 14-Bit-Codes benutzt. Siehe „Eight-to-Fourteen-Modulation“ (EFM).

22.5. RAID

RAID steht für *Redundant Array of Independent Disks*.

Idee Paralleler Einsatz mehrerer Festplatten mit gleichem Inhalt, um I/O-Anfragen parallel verarbeiten zu können.

Prinzip Mehrere Festplatten (u. U. auch Partitionen) werden entweder von einer Hardwarekomponente oder vom Betriebssystem so gekapselt, dass sie wie eine einzige logische Platte erscheinen. Die Daten werden verteilt abgespeichert.

22.5.1. RAID Typen

RAID-0 Mehrere physikalische Platten werden zu einer logischen zusammengefasst.

Die logische Platte wird in Streifen aufgeteilt, die „round robin“ auf die physikalischen Platten verteilt werden.

Keine Redundanz/Ausfallsicherheit der Daten.

RAID-1 Physikalische Platten haben denselben Inhalt, wodurch Lesezugriffe parallelisiert werden können.

Bei 2 Platten können defekte Daten wahrscheinlich nicht wiederhergestellt werden, bei mehreren Platten steigt die Wahrscheinlichkeit, allerdings ohne jemals sicher über die Korrektheit der wiederhergestellten Daten sein zu können.

RAID-0+1 Dies ist ein via RAID-1 gespiegeltes RAID-0 System.

RAID-3 Prüfsummen werden zusätzlich auf einer weiteren physikalischen Platte gespeichert.

RAID-5 Prüfsummen werden verteilt auf den physikalischen Platten gespeichert. Somit enthält jede Platte gleichzeitig Daten und Prüfsummen.

22.6. Dateisysteme

Dateisysteme sind Systeme zur Organisation von Dateien.

- Dateien müssen geöffnet und wieder geschlossen werden können.
- Dateinamen müssen physischen Daten zugeordnet werden können.
- Spezielle Eigenschaften eines Datenträgers (Festplatte, USB-Stick, CD-ROM, ...) müssen im System berücksichtigt werden.

Abhängig vom Dateisystem habe Dateien unterschiedliche Attribute:

- Dateiname
- Ablageort (Ordner)
- Größe
- Zugriffsrechte
- Meta-Daten

Die Anforderungen an ein modernes Dateisystem umfassen:

Paralleler Zugriff Bereitstellung von Locks (*Multiple Readers, Single Writer*), welche für die gesamte Datei oder (bspw. bei Datenbanksystemen) nur für Bereiche einer Datei bestimmt werden können.

Ausfallsicherheit Bspw. bei Stromausfall muss Datenkonsistenz gewährleistet werden. Lösung: *Journaling*.

22.6.1. Journaling

... beschreibt eine Methode zur Datenkonsistenzsicherung.

Prinzip Alle Schreibaktionen werden – ohne zunächst die Originaldaten zu verändern – in ein Protokoll (Journal) geschrieben. Erst wenn im Journal der Abschluss einer Schreiboperation vermerkt wurde, werden (durch Umschreiben des Dateimappings) die neuen Daten referenziert. Stürzt das System also ab, so bleibt das Dateisystem konsistent, da die Veränderungen bislang nur im Journal durchgeführt wurden.

Metadaten-Journaling beschränkt sich auf die Konsistenzsicherung des Dateisystems an sich, d. h. nur die Konsistenz der Attribute von Dateien ist garantiert, aber nicht die Konsistenz der Dateiinhalte.

Full-Journaling garantiert die Konsistenz des gesamten Datenträgers, da auch Dateiinhalte mit dem Journaling verwaltet werden.

22.6.2. Dateisystemtypen

Lineare Dateisysteme werden auf Lochstreifen/-karten und Magnetbändern verwendet, die Daten werden direkt hintereinander geschrieben.

Hierarchische Dateisysteme werden auf modernen Datenträgern verwendet. Die Daten werden in einer Ordnerstruktur abgelegt. Es gibt genau ein Wurzelverzeichnis. In jedem Verzeichnis können Dateien und/oder weitere Verzeichnisse liegen.

Netzwerkdateisysteme werden lokal wie normale Dateisysteme verwendet, die Daten liegen jedoch auf einem entfernten Server (bspw. FTP, Netzlaufwerke).

22.6.3. Windows

- Datenträger (Partitionen auf physischen Speichermedien) werden mit einem Buchstaben gefolgt von einem Doppelpunkt und einem Backslash dargestellt (bekannt: C:\).

- Jeder Datenträger hat einen eigenen Verzeichnisbaum und ein eigenes Wurzelverzeichnis.

- Auf NT-Technologie basierende Windows-Systeme arbeiten intern wie ein Unix-System, verbergen dies aber vor dem Anwender.

22.6.4. Unix/Linux

- Es gibt nur einen einzigen Verzeichnisbaum (mit „/“ als Wurzelverzeichnis).

- Datenträger können an beliebigen Stellen des Baums eingehängt werden (auch rekursiv und mit unterschiedlichen Dateisystemen).

22.6.5. Rechteverwaltung am Beispiel von Unix/Linux

Unix hat verschiedene Ebenen der Rechteverwaltung:

1. Es gibt den Systemadministrator „root“. Dieser existiert auf allen Unix/Linux-Systemen und darf per Definition alles.

2. Es gibt den Eigentümer einer Datei. Dieser darf in der Regel Dateien lesen, schreiben und ausführen. Ein Benutzer ist Mitglied in einer oder mehreren Benutzergruppen.

3. Es gibt die Benutzergruppe einer Datei. Ein Benutzer, der auf eine Datei zugreift, die seiner Benutzergruppe gehört, die er aber nicht selbst erstellt hat, kann diese in der Regel nur lesen und ausführen, jedoch nicht verändern.

4. Es gibt dann noch solche Benutzer, die weder Eigentümer einer Datei noch in der Benutzergruppe der Datei sind. Diese Benutzer haben in der Regel nur Lesezugriff.

Somit hat eine Datei folgende Attribute:

Dateiname und -ort

Dateigröße

Erstellungszeitpunkt

Modifikationszeitpunkt

Referenzierungszähler

Eigentümer

Eigentümerrechte I.d.R. lesbar, schreibbar, ausführbar.

Gruppe

Gruppenrechte I.d.R. lesbar, manchmal auch noch ausführbar.

Fremdrechte I.d.R. nur lesbar.

Als Beispiel ein Directory-Listing eines Verzeichnisses meines Linux-Systems:

```
sto@linux-oknc:~/Dokumente/matse> la
insgesamt 11396
drwxr-xr-x 3 sto users    4096  7. Feb 11:47 .
drwxr-xr-x 4 sto users    4096  6. Feb 09:59 ..
-rw-r--r-- 1 sto users     569  7. Feb 09:56 3c.asm
-rw-r--r-- 1 sto users 1333953  6. Feb 08:37 IT-Grundlagen....
-rw-r--r-- 1 sto users  411971  6. Feb 08:37 IT_Grundlagen_...
-rw-r--r-- 1 sto users  369857  6. Feb 08:36 IT_Grundlagen_...
-rw-r--r-- 1 sto users  360684  6. Feb 08:37 IT_Grundlagen_...
-rw-r--r-- 1 sto users  407807  6. Feb 08:36 IT_Grundlagen_...
-rw-r--r-- 1 sto users  311985  6. Feb 08:36 IT_Grundlagen_...
-rw-r--r-- 1 sto users 7471963  6. Feb 08:36 IT_Grundlagen_...
-rw-r--r-- 1 sto users  127652  6. Feb 08:37 IT-Grundlagen_...
-rw-r--r-- 1 sto users  282285  6. Feb 08:37 IT-grundlagen_...
-rw-r--r-- 1 sto users  551371  6. Feb 08:37 Kryptologie.pdf
drwxr-xr-x 2 sto users    4096  7. Feb 11:47 verzeichnis
```

Der erste Buchstabe „`d`“ gibt an, ob es sich um ein Verzeichnis handelt. Dann folgen drei Buchstaben für die Benutzerzugriffsrechte (*Read, Write, eXecute*), anschließend drei Buchstaben für die Gruppenrechte und dann drei Buchstaben für die Fremdzugriffsrechte. In diesem Falle ist „`sto`“ der Benutzername und „`users`“ der Gruppenname.

Die Zahl hinter den Zugriffsrechten gibt an, wie oft die Datei referenziert wurde. So wird das Verzeichnis „`verzeichnis`“ zweimal referenziert: einmal durch dieses Verzeichnis „`matse`“, und einmal durch das „`.`“ im Verzeichnis „`verzeichnis`“. Das „`.`“ in diesem Verzeichnis „`matse`“ wird durch den Verzeichnisnamen im übergeordneten Verzeichnis „`Dokumente`“ referenziert, durch das Verzeichnis „`matse`“ selbst, und durch das „`..`“ im Verzeichnis „`verzeichnis`“.

23. Kryptologie

23.1. Definitionen

Kryptografie ist die Wissenschaft der Verschlüsselung von Informationen.

Kryptoanalyse bezeichnet Methoden zur unbefugten Entschlüsselung von verschlüsselten Informationen.[1]

Klartext M Unverschlüsselte Nachricht.

Geheimtext (Chiffretext) C Verschlüsselte Nachricht.

(De-)Chiffrierung Ent- bzw. Verschlüsselung von Informationen.

Schlüssel Vertrauliche Information zum Ver- bzw. Entschlüsseln.

23.2. Einsatzgebiete und Anforderungen

23.2.1. Einsatzgebiete

- Sichere Kommunikation über ein unsicheres Medium.
- Integrität/Signierung von Nachrichten, Authentifizierung von Kommunikationspartnern.

23.2.2. Anforderungen

Kerckhoffs Prinzip (1883):

- Das System muss unentzifferbar sein.
- Das System selbst darf keine Geheimhaltung erfordern.
- Der Algorithmus muss leicht zu übermitteln sein und ein Mensch muss sich den Schlüssel ohne schriftliche Aufzeichnung merken können.
- Das System sollte (muss) mit telegrafischer Kommunikation kompatibel sein (Telefon, E-Mail, etc.).

[1] Achtung: Es ist eine *Analyse*, d.h. die Entschlüsselung an sich fällt unter Kryptografie.

- Das System muss transportabel sein und die Bedienung darf nicht mehr als eine Person erfordern.
- Das System muss einfach anwendbar sein.

23.3. Klassische Verfahren

... basieren größtenteils auf Alphabeten und werden schon seit der Antike verwendet.

Transpositionschiffre bedeutet eine Umordnung der Zeichen. Einfachstes Beispiel: Die zu verschlüsselnde Nachricht rückwärts schreiben.

Substitutionschiffre bedeutet eine Ersetzung der Zeichen. Einfachstes Beispiel: Die Buchstaben einer Nachricht durch die im Alphabet folgenden Buchstaben ersetzen.

Skytale ist ein Transpositionschiffre.

Man nehme ein Pergament und wickle es spiralförmig um einen Stab und schreibe die Nachricht quer auf das Pergament, sodass jedes Wort mehrere Streifen belegt.

Hier ist der Klartext das um den Holzstab gewickelte Pergament, der Geheimtext das Pergament ohne den Stab und der Schlüssel ist der Durchmesser des Stabes.

Monoalphabetisches Substitutionschiffre Jedem Buchstaben im Eingabealphabet ist bijektiv ein anderer Buchstabe im Geheimalphabet zugeordnet. Der Schlüssel ist das Geheimalphabet.

Cäsarchiffre ist ein monoalphabetisches Substitutionschiffre bei dem zwei zunächst identische Alphabete um n Zeichen zueinander rotiert werden. Der Schlüssel ist das n.

Eine Variante hiervon ist die Erstellung eines zufällig erzeugten Geheimalphabets.

Polyalphabetisches Substitutionschiffre ist ein Substitutionschiffre mit mehreren Geheimalphabeten. Jedem Zeichen wird abhängig von seiner Position ein anderes Geheimalphabet zugeordnet.

Viginère-Verschlüsselung ist ein polyalphabetisches Substitutionschiffre.

Ein Schlüsselwort bestimmt, wie viele und welche Geheimalphabete verwendet werden. Die einzelnen Geheimalphabete leiten sich aus der Cäsarchiffre ab: Das erste n_1 wird so bestimmt, dass das A durch den ersten Buchstaben c_1 des Geheimalphabets substituiert wird. n_2 wird so bestimmt, dass A durch c_2 substituiert wird, usw..

23.4. Moderne Verfahren

... basieren auf mathematischen Problemstellungen und werden seit Mitte des 20. Jahrhunderts verwendet.

Symmetrische Verfahren sind Block- und Stromchiffren. Zum Ver– und Entschlüsseln wird derselbe Schlüssel verwendet.

Asymmetrische Verfahren sind Hybridverfahren. Zum Ver– und Entschlüsseln werden verschiedene Schlüssel verwendet.

23.4.1. Symmetrische Verfahren

Vorteile Geringer Rechenaufwand zur Ver– und Entschlüsselung.

Nachteile Der geheime Schlüssel muss über ein unsicheres Medium transportiert werden. Für n Personen benötigt man $\frac{n\cdot(n-1)}{2}$ verschiedene Schlüssel.

Bekannte Verfahren DES (Data Encryption Standard), AES (Advanced Encryption Standard), Blowfish, ...

Blockchiffre Der Klartext wird in Blöcke gleicher Größe unterteilt und blockweise mit dem gleichen Schlüssel verschlüsselt. Der letzte Block muss daher ggf. mit Füllbits gefüllt werden.

Stromchiffre Der Klartext wird zeichenweise anhand eines *Schlüsselstroms* verschlüsselt. Schlüsselstrom, Klartext und Geheimtext haben dieselbe Länge.

23.4.2. Asymmetrische Verfahren

Es gibt zwei Schlüssel: einen *privaten Schlüssel* zum Entschlüsseln und einen *öffentlichen Schlüssel* zum Verschlüsseln.

Vorteile Der private Schlüssel muss nicht übertragen werden, und es werden für n Personen nur noch $2n$ Schlüssel benötigt.

Nachteile Hoher Rechenaufwand (ca. Faktor 10^3).

Bekannte Verfahren RSA

23.4.3. RSA

- Basiert auf Primfaktorzerlegung.
- Am häufigsten genutztes asymmetrisches Verfahren.

23.4.3.1. Verfahren

Schlüsselpaare Privat (d, n) und öffentlich (e, n).

Verschlüsselung $C = M^e \bmod n$

Entschlüsselung $M = C^d \bmod n$

Schlüsselerstellung:

1. Wähle zwei Primzahlen p, q mit $p \neq q$.
2. Berechne $N = p \cdot q$ und $\varphi(N) = (p-1) \cdot (q-1)$.
3. Wähle ein e so dass $e \nmid \varphi(N) \land 1 < e < \varphi(N)$.
4. Berechne d mit $e \cdot d \equiv 1 \pmod{\varphi(N)}$.
5. $p, q, \varphi(N)$ werden nicht mehr benötigt und müssen *sicher* vernichtet werden.
6. Die Schlüssel sind nun (d, N) und (e, N).

23.5. Anwendungsgebiete moderner Verfahren

23.5.1. Signierung von Nachrichten

1. Berechnung der Hashsumme einer Nachricht (bspw. mit MD5).
2. Verschlüsselung der Hashsumme mit dem privaten Schlüssel.
3. Der Empfänger entschlüsselt die übermittelte verschlüsselte Hashsumme mit dem öffentlichen Schlüssel.
4. Der Empfänger vergleicht die selbst errechnete Hashsumme mit der entschlüsselten Hashsumme.

23.5.2. Hybridverfahren zur Verschlüsselung von Daten

1. Der verhältnismäßig kurze Schlüssel S eines symmetrischen Verfahrens wird mit dem öffentlichen Schlüssel des Empfängers in S_e verschlüsselt.
2. Die Daten D an sich werden mit dem symmetrischen Verfahren und dem Schlüssel S nach $D_e$verschlüsselt.
3. Der Sender übermittelt D_e und S_e.
4. Der Empfänger entschlüsselt S_e mit seinem privaten Schlüssel und kann damit nun D_e entschlüsseln.

23.6. Kryptoanalyse

... wird oftmals mit kriminellem Hintergrund genutzt um geheime Informationen auszuspähen.

Angriffsarten:

Brute Force Ausprobieren aller möglichen Schlüssel, das Ergebnis pro Schlüssel muss jeweils inhaltlich geprüft werden.

Wörterbuchangriff Annahme, dass der Schlüssel einem Muster unterliegt. Schneller als reines Brute Force, schlägt aber fehl, wenn der Schlüssel nicht im Wörterbuch vorhanden ist.

Häufigkeitsanalyse Funktioniert nur bei monoalphabetischen Chiffren. Es wird die Häufigkeit der Buchstaben in einer Sprache ausgenutzt.

Durch wachsende Rechenleistung schrumpft die Sicherheit heutiger Algorithmen, es müssen daher neue Verfahren erdacht werden.

24. Datensicherheit

Datenschutz ist der Schutz von Daten vor Missbrauch, unbefugter Einsicht/-Verwendung und vor Änderung/Verfälschung.

Datensicherung beschreibt die Menge aller Maßnahmen gegen Verlust, Manipulation und unbefugten Zugriff von Daten, die durch menschliches/-technisches Versagen, Katastrophen und/oder vorsätzliche Taten hervorgerufen werden können.

Datensicherheit ist ein durch die Datensicherung angestrebter Zustand. Absolute Datensicherheit gibt es nicht.

24.1. Sicherheitsrisiken

- Diebstahl
- Zerstörung durch Feuer, Sabotage, Naturkatastrophen, Wassereinbruch, Transportschäden, Betriebsdefekten, ...
- Unbefugter Zugriff (Hacker, oder einfach eine offen gelassene Tür)
- Menschliche Fehler (Bedienungsfehler, Nachlässigkeit)
- Malware („Ach schau mal an, ein USB-Stick, mal sehen was da drauf ist...“)
- Viren, Würmer, Trojaner, Backdoors, Spyware, Phishing

24.2. Sicherheitsmaßnahmen

- Bauliche Maßnahmen (Feuerschutz, Diebstahlsicherung, physische Abschottung)
- Redundanz der Geräte erhöhen
- Gewissenhafte Backups und Verschlüsselung
- Ausschließlich autorisierte Software verwenden
- Arbeitsplatzrechner ohne Nutzbarkeit externer Datenträger
- Gesundes Misstrauen und Aufmerksamkeit

24.3. Bundesdatenschutzgesetz

Quelle: [Bun90]

> <u>§ 9 Technische und organisatorische Maßnahmen</u>
> Öffentliche und nicht-öffentliche Stellen, die selbst oder im Auftrag personenbezogene Daten erheben, verarbeiten oder nutzen, haben die technischen und organisatorischen Maßnahmen zu treffen, die erforderlich sind, um die Ausführung der Vorschriften dieses Gesetzes, insbesondere die in der Anlage zu diesem Gesetz genannten Anforderungen, zu gewährleisten. Erforderlich sind Maßnahmen nur, wenn ihr Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht.

25. Assembler

> Eine Assemblersprache (oft abgekürzt als ASM bzw. asm) ist eine spezielle Programmiersprache, welche die Maschinensprache einer spezifischen Prozessorarchitektur in einer für den Menschen lesbaren Form repräsentiert. Jede Computerarchitektur hat folglich ihre eigene Assemblersprache.
>
> *Quelle: [Wik12b]*

Im Folgenden betrachten wir die Intel x86 Assemblersprache.

25.1. Grundlagen

Beim Datentransfer gilt zu beachten:

1. Man kann Daten zwischen Registern transferieren.
2. Man kann Daten zwischen Registern und Speicher transferieren.
3. Es ist keine direkter Transfer im Arbeitsspeicher möglich, man muss den Umweg über die Register gehen.
4. Jede arithmetische oder logische Operation setzt die sog. Flags, die mit den bedingten Sprungbefehlen benutzt werden können.

25.1.1. Register

Die Intel x86 (32 Bit) Architektur besitzt folgende Register:

al, ah, ax, eax *Accumulator* – Akkumulator, wird für mathematische/logische Operationen benutzt.

bl, bh, bx, ebx *Base* – Basis, wird für die dynamische Berechnung einer Speicheradresse benutzt.

cl, ch, cx, ecx *Counter* – Zähler, wird für einfache Schleifen benutzt.

dl, dh, dx, edx *Data* – Datenregister, wird für Datentransfers über rechnerinterne Ports[1] benutzt.

[1] Achtung: Gemeint sind nicht die Internet-Ports. Die Ports hier sind Datenkanäle um mit der rechnerinternen Peripherie (Soundkarte, DMA, Tastatur, etc.) kommunizieren zu können.

cs *Code Segment* – `eip` ist relativ zu diesem Segment (vgl. 20.6.2 auf Seite 80).

ss *Stack Segment* – `esp` ist relativ zu diesem Segment (vgl. 20.6.2 auf Seite 80).

ds, es, fs, gs *Data Segment, Extra Segments* – `esi`, `edi` sind relativ zu diesem Segment (vgl. 20.6.2 auf Seite 80).

eip *Instruction Pointer* – Zeigt auf die nächste auszuführende Anweisung.

esp *Stack Pointer* – Zeigt auf die Adresse im Stack, auf der die nächsten Daten gepusht werden.

ebp *Base Pointer* – Wird bei Funktionsaufrufen benutzt um Stack-Frames anlegen zu können.

esi *Source Index* – Quelladresse für sog. String-Opcodes, relativ zu `ds`.

edi *Destination Index* – Zieladresse für sog. String-Opcodes, relativ zu `es`.

<table>
<tr><td colspan="3">e*x (32 Bit)</td></tr>
<tr><td></td><td colspan="2">*x (16 Bit)</td></tr>
<tr><td></td><td>*h (8 Bit)</td><td>*l (8 Bit)</td></tr>
</table>

Tabelle 25.1.: Aufbau Register

25.1.2. Mnemonics

Mnemonics sind menschenlesbare Darstellungen für die Opcodes. Die Opcodes sind die Binärdarstellungen der Maschinensprache.

25.2. Datenbefehle

Ein Beispiel:

```
mov eax, 123
mov [0], eax
mov eax, [4]
mov word ptr [8], 123
xchg eax, edx
xchg eax, [9]
```

1. Ins Register `eax` wird 123 geschrieben.
2. An die Speicherstelle 0 werden die 4 Bytes aus `eax` kopiert.
3. In das Register `eax` werden 4 Bytes von der Speicherstelle 4 kopiert.
4. An die Speicherstelle 8 werden 2 Bytes geschrieben. Die Angabe `word ptr` muss angegeben werden, da der Assemblierer sonst nicht weiß, wie viele Bytes geschrieben werden sollen.
5. Inhalt von `eax` und `edx` vertauschen.
6. Inhalt von `eax` mit den 4 Bytes an Speicherstelle 9 vertauschen.

25.3. Mathematische Befehle

```
mul eax, 4
imul eax, -3
idiv eax, -3
div eax, 4
add eax, 9
sub eax, -3
mul eax, edx
```

1. `eax` mit 4 multiplizieren, das Ergebnis steht im Registerpaar `edx:eax`, da `eax` bei einer Multiplikation überlaufen kann.

2. `eax` vorzeichenbehaftet mit -3 multiplizieren (i steht für *int*). Vorzeichenbehaftet deswegen, weil die -3 sonst als 0xFFFFFFFD interpretiert werden würde.

3. `eax` vorzeichenbehaftet durch -3 dividieren. Das Divisionsergebnis steht in `eax`, das Moduloergebnis in `edx`.

4. `eax` durch 4 dividieren. Das Divisionsergebnis steht in `eax`, das Moduloergebnis in `edx`.

5. 9 zu `eax` addieren, es wird bei einem Überlauf das *Carry-Flag* gesetzt.

6. -3 von `eax` subtrahieren, es wird bei einem Überlauf das *Carry-Flag* gesetzt.

7. `eax` mit `edx` multiplizieren.

25.4. Sprünge

cmp Zwei Werte vergleichen (Speicher und Register, Speicher und Wert, Register und Wert, Register und Register).

jmp Sprung ohne Bedingung (wird in jedem Fall ausgeführt).

jc Sprung wenn das Carry-Flag gesetzt ist.

jz, je Zero-Flag gesetzt (wenn beide Werte gleich sind).

ja, jb Above, Below (bei vorzeichenbehafteten Vergleichen).

jae, jbe Above or Equal, Below or Equal (bei vorzeichenbehafteten Vergleichen).

jg, jl Greater, Less (ohne Vorzeichenbeachtung).

jge, jle Greater or Equal, Less or Equal (ohne Vorzeichenbeachtung).

Zusätzlich kann hinter das *j* ein *n* (not) angehängt werden um die Bedeutung des Vergleiches umzukehren (*jnae* und *jb* sind bspw. somit identisch).

25.5. Beispiele

Zählt die Bits in `eax` und gibt die Anzahl der Bits in `eax` zurück.

25.5.1. Muster von Dr. Alexander Voß

```
mov eax,0x001a ; -> 0001 1010 -> 3 Bits
call A03c
mov eax, 0
call A03c
mov eax, 0xffffffff ; -> 32 Bits
call A03c
jmp end
;-----------------------------------------------------------

A03c:
    push ebx ; Register auf dem Stack sichern
    push ecx
    mov ecx, 32 ; Anzahl Schleifendurchläufe
    mov ebx, 0 ; Bit-Zähler
A03c_loop:
    shr eax, 1 ; eax um 1 Bit nach rechts schieben, das
              ; herausgeschobene Bit steht im Carry-Flag
    jnc A03c_next ; Carry nicht gesetzt?
    inc ebx ; Bit-Zähler um 1 erhöhen (inkrementieren)
A03c_next:
    cmp eax, 0 ; eax mit 0 vergleichen
    jz A03c_end ; Wenn eax==0 ist, dann brauchen wir nicht
                ; weiter zu zählen
    dec ecx ; Durchlaufzähler-- (dekrementieren)
    jnz A03c_loop ; falls ecx!=0: nächster Durchlauf
A03c_end:
    mov eax, ebx ; Bit-Zähler nach eax kopieren
    pop ecx ; Register wiederherstellen
    pop ebx
    ret ; Rücksprungadresse vom Stack holen und nach
        ; eip schreiben

end:
```

25.5.2. Alternative 1

Hier werden die Befehle `loop` und `adc` benutzt um den Code zu verkürzen.

```
; ...
A03c:
    push ebx ; Register auf dem Stack sichern
    push ecx
    mov ecx, 32 ; Anzahl Schleifendurchläufe
    mov ebx, 0 ; Bit-Zähler
A03c_loop:
    shr eax, 1 ; eax um 1 Bit nach rechts schieben, das
              ; herausgeschobene Bit steht im Carry-Flag
    adc ebx, 0 ; 0+CarryFlag auf ebx aufaddieren
    loop A03c_loop ; loop führt ein "dec ecx" durch und
                   ; springt zum angegebenen Label, wenn
                   ; ecx!=0
A03c_end:
    mov eax, ebx ; Bit-Zähler nach eax kopieren
    pop ecx ; Register wiederherstellen
    pop ebx
    ret ; Rücksprungadresse vom Stack holen und nach
        ; eip schreiben
```

25.5.3. Alternative 2

Hier wird die Schleifenvariable `ecx` gespart.

```
; ...
A03c:
    push ebx ; Register auf dem Stack sichern
    mov ebx, 0 ; Bit-Zähler
A03c_loop:
    shr eax, 1 ; eax um 1 Bit nach rechts schieben, das
              ; herausgeschobene Bit steht im Carry-Flag
    adc ebx, 0 ; 0+CarryFlag auf ebx aufaddieren
    cmp eax, 0
    jne A03c_loop
A03c_end:
    mov eax, ebx ; Bit-Zähler nach eax kopieren
    pop ebx
    ret ; Rücksprungadresse vom Stack holen und nach
        ; eip schreiben
```

25.6. Compiler-Optimierungen

Vom GCC optimierte Version:

```
A03c: ; hier ist die zu untersuchende Zahl in edi
    xor eax, eax ; xor geht schneller als mov (hier: =0)
    test edi, edi ; test ist ein Bittest (hier: ==0)
    jz A03c_end ; Parameter ist 0
    nop ; no-operation: eingefügt, weil der
        ; Prozessor bei gewissen "alignments"
        ; die Anweisungen schneller verarbeiten kann
    ; ... es folgen noch 9 nop’s für 16-Byte Alignment
A03c_loop:
    mov edx, edi
    and edx, 1
    add eax, edx ; eax ist der Bitzähler
    shr edi, 1
    jnz A03c_loop ; Zero-Flag ist nur gesetzt, wenn
                  ; edi beim shr auf 0 gesetzt wurde
A03c_end:
    repz ret
```

Original C++ Code:

```
unsigned int xxx(unsigned int num) {
    int res = 0;
    while(num != 0) {
        res += num&1;
        num >>= 1;
    }
    return res;
}
```

Dieser kurze Code ist übrigens nicht wirklich optimierbar, bei komplexeren Algorithmen kommen die Optimierungsalgorithmen im GCC wesentlich besser zum Tragen, allerdings sind diese optimierten Codes zu lang und zu kompliziert für diesen kurzen Ausflug.

Es ist aber interessant, dass der GCC für folgenden C++ Code exakt denselben Assembler-Code erzeugt:

```
unsigned int xxx(unsigned int num) {
    int res = 0;
    while(num != 0) {
        int tmp = num & 1;
        res = res + num;
        num = num / 2;
    }
    return res;
}
```

Zum Weiterlesen empfiehlt sich [Wik12a] und unter `http://jasmin.sourceforge.net/` findet sich ein in Java geschriebener x86-Simulator, mit dem sich sehr schön die Funktionsweise von Assembler nachvollziehen lässt.

Teil VI.

Anhänge

A. Formelsammlung

A.1. Potenzreihen

$$\exp(x) = \sum_{n=0}^{\infty} \frac{1}{n!} x^n \mid x \in \mathbb{R}$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \mid x \in \mathbb{R}$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \mid x \in \mathbb{R}$$

$$\sinh(x) = \sum_{n=0}^{\infty} \frac{1}{(2n+1)!} x^{2n+1} \mid x \in \mathbb{R}$$

$$\cosh(x) = \sum_{n=0}^{\infty} \frac{1}{(2n)!} x^{2n} \mid x \in \mathbb{R}$$

$$\arctan(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1} \mid |x| \leq 1$$

$$\ln(1+x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n \mid x \in (-1, 1]$$

$$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{1}{n} x^n \mid x \in [-1, 1)$$

$$\sqrt{1+x} = \sum_{n=0}^{\infty} \binom{\frac{1}{2}}{n} x^n \mid |x| \leq 1$$

$$\frac{1}{\sqrt{1+x}} = \sum_{n=0}^{\infty} \binom{-\frac{1}{2}}{n} x^n \mid |x| < 1$$

A.2. Andere Reihen

Geometrische Reihe $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x} \mid |x| < 1$

Endliche geometrische Reihe $\sum_{n=0}^{k} x^n = \frac{1-x^{k+1}}{1-x} \mid x \neq 1$

Harmonische Reihe $\sum_{n=1}^{\infty} x^{-n}$ konvergiert falls $x > 1$

Binomische Reihe $\sum_{n=0}^{\infty} \binom{r}{n} x^n = (1+x)^r$ nur wenn $|x| \leq 1 \wedge r > 0$ oder $|x| < 1 \wedge r < 0$

A.3. Wichtige Grenzwerte

Jeweils für $n \to \infty$:

$$\sqrt[n]{a} \to 1$$

$$\sqrt[n]{n} \to 1$$

$$\sqrt[n]{n!} \to \infty$$

$$\left(1 - \frac{1}{n}\right)^n = \frac{1}{n}\sqrt[n]{n!} \to e^{-1}$$

$$\left(1 - \frac{x}{n}\right)^n \to e^{-x}$$

$$\left(\frac{n+1}{n}\right)^n = \left(1 + \frac{1}{n}\right)^n = \frac{n}{\sqrt[n]{n!}} \to e$$

$$\left(1 + \frac{x}{n}\right)^n \to e^x$$

$$\binom{a}{n} \to 0 \mid a > -1$$

$$\frac{a^n}{n!} \to 0$$

$$\frac{n^n}{n!} \to \infty$$

$$\frac{a^n}{n^k} \to \infty \mid a > 1 \wedge k = \text{const.}$$

$$a^n n^k \to 0 \mid |a| < 1 \wedge k = \text{const.}$$

A.4. Integrale und Ableitungen

A.4.1. Regeln

Produktregel $(uv)' = u'v + uv'$, $(uvw)' = u'vw + uv'w + uvw'$, etc., gleiches für Vektoren (sowohl Skalarprodukt als auch Kreuzprodukt).

Quotientenregel $\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$

Kettenregel $(y(x(t)))' = y'(x(t)) \cdot x'(t)$, $(\vec{u}(\lambda(t)))' = \vec{u}'(\lambda(t)) \cdot \lambda'(t)$

Substitutionsregel $\int f(x)\,\mathrm{d}x = \int f(g(u))g'(u)\,\mathrm{d}u$ mit $x = g(u)$ und $\mathrm{d}x = g'(u)\,\mathrm{d}u$

Partielle Integration $\int u'v\,\mathrm{d}x = uv - \int uv'\,\mathrm{d}x$

$$\int x^n\,\mathrm{d}x = \frac{1}{n+1}x^{n+1} \mid n \neq 1$$

$$\int \frac{f'(x)}{f(x)}\,\mathrm{d}x = \ln(|f(x)|)$$

A.4.2. Einfache Integrale

$$\int nx^{n-1}\,\mathrm{d}x = x^n$$

$$\int \frac{-n}{x^{n+1}}\,\mathrm{d}x = x^{-n}$$

$$\int \frac{1}{x+a}\,\mathrm{d}x = \ln(|x+a|)$$

$$\int \frac{1}{2\sqrt{x}}\,\mathrm{d}x = \sqrt{x}$$

$$\int \frac{1}{n\sqrt[n]{x^{n-1}}}\,\mathrm{d}x = \sqrt[n]{x}$$

A.4.3. Exponentialfunktionen

$$\int e^{ax}\,\mathrm{d}x = \frac{1}{a}e^{ax}$$

$$\int xe^{ax}\,\mathrm{d}x = \frac{ax-1}{a^2}e^{ax}$$

$$\int a^x \ln(a)\,\mathrm{d}x = a^x$$

$$\int x^x(1+\ln(x))\,\mathrm{d}x = x^x$$

A.4.4. Logarithmusfunktionen

$$\int \ln(x)\,\mathrm{d}x = x\ln(x) - x$$

$$\int x\ln(x)\,\mathrm{d}x = x^2\left(\frac{\ln(x)}{2} - \frac{1}{4}\right)$$

$$\int \ln^2(x)\,\mathrm{d}x = x\ln^2(x) - 2x\ln(x) + 2x$$

A.4.5. Trigonometrische Funktionen

$$\int \cos(x)\,\mathrm{d}x = \sin(x)$$

$$\int -\sin(x)\,\mathrm{d}x = \cos(x)$$

$$\int \tan(x)\,\mathrm{d}x = -\ln(|\cos(x)|)$$

$$\int \sin^2(ax)\,\mathrm{d}x = \frac{1}{2}x - \frac{1}{4a}\sin(2ax)$$

$$\int \cos^2(ax)\,\mathrm{d}x = \frac{1}{2}x + \frac{1}{4a}\sin(2ax)$$

$$\int \sin(ax)\cos(ax)\,\mathrm{d}x = \frac{1}{2a}\sin^2(ax)$$

$$\int \frac{1}{\sin(ax)\cos(ax)}\,\mathrm{d}x = \frac{1}{a}\ln(|\tan(ax)|)$$

$$\int x\sin(ax)\,\mathrm{d}x = \frac{1}{a^2}\sin(ax) - \frac{x}{a}\cos(ax)$$

$$\int x\cos(ax)\,\mathrm{d}x = \frac{1}{a^2}\cos(ax) + \frac{x}{a}\sin(ax)$$

$$\int e^{ax}\sin(bx)\,\mathrm{d}x = \frac{e^{ax}}{a^2+b^2}\left(a\sin(bx) - b\cos(bx)\right)$$

$$\int e^{ax}\cos(bx)\,\mathrm{d}x = \frac{e^{ax}}{a^2+b^2}\left(a\cos(bx) + b\sin(bx)\right)$$

$$\int \frac{1}{\cos^2(x)}\,\mathrm{d}x = \tan(x)$$

$$\int \frac{-1}{\sin^2(x)}\,\mathrm{d}x = \cot(x)$$

$$\int \frac{1}{\sqrt{1-x^2}}\,\mathrm{d}x = \arcsin(x)$$

$$\int \frac{-1}{\sqrt{1-x^2}}\,\mathrm{d}x = \arccos(x)$$

$$\int \frac{1}{1+x^2}\,\mathrm{d}x = \arctan(x)$$

$$\int \frac{-1}{1+x^2}\,\mathrm{d}x = \operatorname{arccot}(x)$$

A.4.6. Hyperbolische Funktionen

$$\int \cosh(x)\,\mathrm{d}x = \sinh(x)$$

$$\int \sinh(x)\,\mathrm{d}x = \cosh(x)$$

$$\int \frac{1}{\cosh^2(x)}\,\mathrm{d}x = \tanh(x)$$

$$\int \frac{-1}{\sinh^2(x)}\,\mathrm{d}x = \coth(x)$$

$$\int \frac{1}{\sqrt{x^2+1}}\,\mathrm{d}x = \operatorname{arsinh}(x)$$

$$\int \frac{1}{\sqrt{x^2-1}}\,\mathrm{d}x = \operatorname{arcosh}(x) \mid x > 1$$

$$\int \frac{1}{1-x^2}\,\mathrm{d}x = \operatorname{artanh}(x) \mid |x| < 1$$

$$\int \frac{1}{1-x^2}\,\mathrm{d}x = \operatorname{arcoth}(x) \mid |x| > 1$$

$$u = \sqrt{x^2+a^2}: \quad \int u\,\mathrm{d}x = \frac{1}{2}\left(xu + a^2\operatorname{arsinh}\left(\frac{x}{a}\right)\right) = \frac{1}{2}\left(xu + a^2\ln\left(x+u\right)\right)$$

$$u = \sqrt{x^2-a^2}: \quad \int u\,\mathrm{d}x = \frac{1}{2}\left(xu - a^2\operatorname{arcosh}\left(\frac{x}{a}\right)\right) = \frac{1}{2}\left(xu\right)$$

$$u = \sqrt{a^2-x^2}: \quad \int u\,\mathrm{d}x = \frac{1}{2}\left(xu + a^2\arcsin\left(\frac{x}{a}\right)\right)$$

Des Weiteren mit

- $X = ax^2 + bx + c$
- $\Delta = 4ac - b^2$
- $a \neq 0$

$$\int \frac{1}{X}\,\mathrm{d}x = \begin{cases} \frac{2}{\sqrt{\Delta}} \arctan\left(\frac{2ax+b}{\sqrt{\Delta}}\right) & \Delta > 0 \\ \frac{-2}{\sqrt{-\Delta}} \operatorname{artanh}\left(\frac{2ax+b}{\sqrt{-\Delta}}\right) & \Delta < 0 \\ \frac{1}{\sqrt{-\Delta}} \ln\left(\frac{2ax+b-\sqrt{-\Delta}}{2ax+b+\sqrt{-\Delta}}\right) & \Delta < 0 \\ \frac{-2}{2ax+b} & \Delta = 0 \end{cases}$$

$$\int \frac{1}{X^2}\,\mathrm{d}x = \frac{2ax+b}{\Delta X} + \frac{2a}{\Delta} \int \frac{1}{X}\,\mathrm{d}x$$

$$\int \frac{x}{X}\,\mathrm{d}x = \frac{1}{2a} \ln(|X|) - \frac{b}{2a} \int \frac{1}{X}\,\mathrm{d}x$$

B. Tabellen

	0.005	0.01	0.025	0.05	0.10	0.90	0.95	0.975	0.99	0.995
1	0.00004	0.00016	0.00098	0.00393	0.01579	2.70554	3.84146	5.02389	6.63490	7.87944
2	0.01003	0.02010	0.05064	0.10259	0.21072	4.60517	5.99146	7.37776	9.21034	10.59663
3	0.07172	0.11483	0.21580	0.35185	0.58437	6.25139	7.81473	9.34840	11.34487	12.83816
4	0.20699	0.29711	0.48442	0.71072	1.06362	7.77944	9.48773	11.14329	13.27670	14.86026
5	0.41174	0.55430	0.83121	1.14548	1.61031	9.23636	11.07050	12.83250	15.08627	16.74960
6	0.67573	0.87209	1.23734	1.63538	2.20413	10.64464	12.59159	14.44938	16.81189	18.54758
7	0.98926	1.23904	1.68987	2.16735	2.83311	12.01704	14.06714	16.01276	18.47531	20.27774
8	1.34441	1.64650	2.17973	2.73264	3.48954	13.36157	15.50731	17.53455	20.09024	21.95495
9	1.73493	2.08790	2.70039	3.32511	4.16816	14.68366	16.91898	19.02277	21.66599	23.58935
10	2.15586	2.55821	3.24697	3.94030	4.86518	15.98718	18.30704	20.48318	23.20925	25.18818
11	2.60322	3.05348	3.81575	4.57481	5.57778	17.27501	19.67514	21.92005	24.72497	26.75685
12	3.07382	3.57057	4.40379	5.22603	6.30380	18.54935	21.02607	23.33666	26.21697	28.29952
13	3.56503	4.10692	5.00875	5.89186	7.04150	19.81193	22.36203	24.73560	27.68825	29.81947
14	4.07467	4.66043	5.62873	6.57063	7.78953	21.06414	23.68479	26.11895	29.14124	31.31935
15	4.60092	5.22935	6.26214	7.26094	8.54676	22.30713	24.99579	27.48839	30.57791	32.80132
16	5.14221	5.81221	6.90766	7.96165	9.31224	23.54183	26.29623	28.84535	31.99993	34.26719
17	5.69722	6.40776	7.56419	8.67176	10.08519	24.76904	27.58711	30.19101	33.40866	35.71847
18	6.26480	7.01491	8.23075	9.39046	10.86494	25.98942	28.86930	31.52638	34.80531	37.15645
19	6.84397	7.63273	8.90652	10.11701	11.65091	27.20357	30.14353	32.85233	36.19087	38.58226
20	7.43384	8.26040	9.59078	10.85081	12.44261	28.41198	31.41043	34.16961	37.56623	39.99685
22	8.64272	9.54249	10.98232	12.33801	14.04149	30.81328	33.92444	36.78071	40.28936	42.79565
24	9.88623	10.85636	12.40115	13.84843	15.65868	33.19624	36.41503	39.36408	42.97982	45.55851
26	11.16024	12.19815	13.84390	15.37916	17.29188	35.56317	38.88514	41.92317	45.64168	48.28988
28	12.46134	13.56471	15.30786	16.92788	18.93924	37.91592	41.33714	44.46079	48.27824	50.99338
30	13.78672	14.95346	16.79077	18.49266	20.59923	40.25602	43.77297	46.97924	50.89218	53.67196
40	20.70654	22.16426	24.43304	26.50930	29.05052	51.80506	55.75848	59.34171	63.69074	66.76596
50	27.99075	29.70668	32.35736	34.76425	37.68865	63.16712	67.50481	71.42020	76.15389	79.48998
60	35.53449	37.48485	40.48175	43.18796	46.45889	74.39701	79.08194	83.29767	88.37942	91.95170
70	43.27518	45.44172	48.75756	51.73928	55.32894	85.52704	90.53123	95.02318	100.42518	104.21490
80	51.17193	53.54008	57.15317	60.39148	64.27784	96.57820	101.87947	106.62857	112.32879	116.32106
90	59.19630	61.75408	65.64662	69.12603	73.29109	107.56501	113.14527	118.13589	124.11632	128.29894
100	67.32756	70.06489	74.22193	77.92947	82.35814	118.49800	124.34211	129.56120	135.80672	140.16949

Tabelle B.1.: Chi-Quadrat-Verteilung

$\Phi(y.yx) = 0.z$ $\quad\Phi(-u) = 1 - \Phi(u)$ $\quad\Phi(u) \cong 1 \mid u \geq 4$

	0	1	2	3	4	5	6	7	8	9
0.0	*500000*	503989	507978	511966	515953	519939	523922	527903	531881	535856
0.1	539828	543795	547758	551717	555670	559618	563559	567495	571424	575345
0.2	579260	583166	587064	590954	594835	598706	602568	606420	610261	614092
0.3	617911	621720	625516	629300	633072	636831	640576	644309	648027	651732
0.4	655422	659097	662757	666402	670031	673645	677242	680822	684386	687933
0.5	691462	694974	698468	701944	705401	708840	712260	715661	719043	722405
0.6	725747	729069	732371	735653	738914	742154	745373	748571	751748	754903
0.7	758036	761148	764238	767305	770350	773373	776373	779350	782305	785236
0.8	788145	791030	793892	796731	799546	802337	805105	807850	810570	813267
0.9	815940	818589	821214	823814	826391	828944	831472	833977	836457	838913
1.0	*841345*	843752	846136	848495	850830	853141	855428	857690	859929	862143
1.1	864334	866500	868643	870762	872857	874928	876976	879000	881000	882977
1.2	884930	886861	888768	890651	892512	894350	896165	897958	899727	901475
1.3	903200	904902	906582	908241	909877	911492	913085	914657	916207	917736
1.4	919243	920730	922196	923641	925066	926471	927855	929219	930563	931888
1.5	933193	934478	935745	936992	938220	939429	940620	941792	942947	944083
1.6	945201	946301	947384	948449	949497	950529	951543	952540	953521	954486
1.7	955435	956367	957284	958185	959070	959941	960796	961636	962462	963273
1.8	964070	964852	965620	966375	967116	967843	968557	969258	969946	970621
1.9	971283	971933	972571	973197	973810	974412	975002	975581	976148	976705
2.0	*977250*	977784	978308	978822	979325	979818	980301	980774	981237	981691
2.1	982136	982571	982997	983414	983823	984222	984614	984997	985371	985738
2.2	986097	986447	986791	987126	987455	987776	988089	988396	988696	988989
2.3	989276	989556	989830	990097	990358	990613	990863	991106	991344	991576
2.4	991802	992024	992240	992451	992656	992857	993053	993244	993431	993613
2.5	993790	993963	994132	994297	994457	994614	994766	994915	995060	995201
2.6	995339	995473	995604	995731	995855	995975	996093	996207	996319	996427
2.7	996533	996636	996736	996833	996928	997020	997110	997197	997282	997365
2.8	997445	997523	997599	997673	997744	997814	997882	997948	998012	998074
2.9	998134	998193	998250	998305	998359	998411	998462	998511	998559	998605
3.0	*998650*	998694	998736	998777	998817	998856	998893	998930	998965	998999
3.1	999032	999065	999096	999126	999155	999184	999211	999238	999264	999289
3.2	999313	999336	999359	999381	999402	999423	999443	999462	999481	999499
3.3	999517	999534	999550	999566	999581	999596	999610	999624	999638	999651
3.4	999663	999675	999687	999698	999709	999720	999730	999740	999749	999758
3.5	999767	999776	999784	999792	999800	999807	999815	999822	999828	999835
3.6	999841	999847	999853	999858	999864	999869	999874	999879	999883	999888
3.7	999892	999896	999900	999904	999908	999912	999915	999918	999922	999925
3.8	999928	999931	999933	999936	999938	999941	999943	999946	999948	999950
3.9	999952	999954	999956	999958	999959	999961	999963	999964	999966	999967

Tabelle B.2.: Standardnormalverteilung

	0.90	0.95	0.975	0.99	0.995
1	3.077684	6.313752	12.706205	31.820516	63.656741
2	1.885618	2.919986	4.302653	6.964557	9.924843
3	1.637744	2.353363	3.182446	4.540703	5.840909
4	1.533206	2.131847	2.776445	3.746947	4.604095
5	1.475884	2.015048	2.570582	3.364930	4.032143
6	1.439756	1.943180	2.446912	3.142668	3.707428
7	1.414924	1.894579	2.364624	2.997952	3.499483
8	1.396815	1.859548	2.306004	2.896459	3.355387
9	1.383029	1.833113	2.262157	2.821438	3.249836
10	1.372184	1.812461	2.228139	2.763769	3.169273
11	1.363430	1.795885	2.200985	2.718079	3.105807
12	1.356217	1.782288	2.178813	2.680998	3.054540
13	1.350171	1.770933	2.160369	2.650309	3.012276
14	1.345030	1.761310	2.144787	2.624494	2.976843
15	1.340606	1.753050	2.131450	2.602480	2.946713
16	1.336757	1.745884	2.119905	2.583487	2.920782
17	1.333379	1.739607	2.109816	2.566934	2.898231
18	1.330391	1.734064	2.100922	2.552380	2.878440
19	1.327728	1.729133	2.093024	2.539483	2.860935
20	1.325341	1.724718	2.085963	2.527977	2.845340
22	1.321237	1.717144	2.073873	2.508325	2.818756
24	1.317836	1.710882	2.063899	2.492159	2.796940
26	1.314972	1.705618	2.055529	2.478630	2.778715
28	1.312527	1.701131	2.048407	2.467140	2.763262
30	1.310415	1.697261	2.042272	2.457262	2.749996
40	1.303077	1.683851	2.021075	2.423257	2.704459
50	1.298714	1.675905	2.008559	2.403272	2.677793
60	1.295821	1.670649	2.000298	2.390119	2.660283
100	1.290075	1.660234	1.983972	2.364217	2.625891
200	1.285799	1.652508	1.971896	2.345137	2.600634
300	1.284380	1.649949	1.967903	2.338842	2.592316
400	1.283672	1.648672	1.965912	2.335706	2.588176
500	1.283247	1.647907	1.964720	2.333829	2.585698
∞	1.281552	1.644854	1.959964	2.326348	2.575829

Tabelle B.3.: t-Verteilung nach Student

C. Beispiele

C.1. Konvergenz einer Folge

Aufgabe
Zur Folge
$$a_{n+1} = 5 - \frac{4}{a_n}$$
ist zu zeigen, dass sie konvergent ist. Außerdem soll der Grenzwert bestimmt werden.

$a_0 = 2$, $a_1 = 3$, daher Vermutung, dass die Folge streng monoton steigend ist. Beweis mittels vollständiger Induktion:

Anfang $a_1 > a_0$

Voraussetzung $\exists n_0 \geq 0 \, \forall n \geq n_0 : a_{n+1} > a_n$

Behauptung $a_{n+2} > a_{n+1}$

Schluss $a_{n+2} = 5 - \frac{4}{a_{n+1}} > 5 - \frac{4}{a_n} = a_{n+1}$

Für den Grenzwert muss gelten:

$$a = 5 - \frac{4}{a} \quad \Leftrightarrow \quad a^2 - 5a + 4 = 0 \quad \Leftrightarrow \quad (a-1)(a-4) = 0$$

Beweis, dass die Folge durch 4 beschränkt ist mittels vollst. Induktion:

Anfang $a_0 < 4$

Voraussetzung $\exists n_0 \geq 0 \, \forall n \geq n_0 : a_n < 4$

Behauptung $a_{n+1} < 4$

Schluss $a_{n+1} = 5 - \frac{4}{a_n} < 5 - \frac{4}{4} = 4$

C.2. Konvergenz einer Reihe

Aufgabe
Analysiere die Konvergenz von

$$\sum_{n=1}^{\infty} \frac{2^n n^n}{(n!)^2}$$

Quotientenkriterium[1]:

$$\begin{aligned}
\lim_{n\to\infty} \frac{2^{n+1}(n+1)^{n+1}}{((n+1)!)^2} \cdot \frac{(n!)^2}{2^n n^n} &= \lim_{n\to\infty} \frac{2(n+1)}{(n+1)^2} \cdot \left(\frac{n+1}{n}\right)^n \\
&= \lim_{n\to\infty} \frac{2}{n+1} \cdot \left(1+\frac{1}{n}\right)^n \\
&= 0 \cdot e < 1
\end{aligned}$$

C.3. Extremwerte einer Funktion mehrerer Veränderlicher

Aufgabe
Berechne die Extremwerte von $f(x,y) = x^4 - 2x^2 + (2x^2-1)y^2$ und bestimme deren Art.

Suche nach kritischen Punkten, die $\operatorname{grad} f = \vec{0}$ erfüllen:

$$\begin{aligned}
f_x &= 4x(x^2 - 1 + y^2) \\
f_y &= 2(2x^2-1)y
\end{aligned}$$

[1] Betragsstriche können weggelassen werden, da die Reihenglieder alle positiv sind.

- $f_y = 0$, $2(2x^2 - 1)y = 0$
  - $y = 0$
    - $f_x = 0$, $4x(x^2 - 1) = 0$
      - $x = 0$, $y = 0$
      - $x = 1$, $y = 0$
      - $x = \text{-}1$, $y = 0$
  - $x = \sqrt{\frac{1}{2}} =: w$
    - $4w\left(y^2 - \frac{1}{2}\right)$
      - $x = w$, $y = w$
      - $x = w$, $y = \text{-}w$
  - $x = \text{-}w$
    - $\text{-}4w\left(y^2 - \frac{1}{2}\right)$
      - $x = \text{-}w$, $y = w$
      - $x = \text{-}w$, $y = \text{-}w$

Ergibt die Kritischen Punkte: $(0,0)$, $(\pm 1, 0)$, $(\pm w, \pm w)$ und $(\pm w, \mp w)$. Überprüfung der Definitheit der HESSE-Matrix nach dem HURWITZ-Kriterium:

$$
\begin{aligned}
Hf &= \begin{pmatrix} 12x^2 - 4 + 4y^2 & 8xy \\ 8xy & 2(2x^2 - 1) \end{pmatrix} \\
(Hf)(0,0) &= \begin{pmatrix} \text{-}4 & 0 \\ 0 & \text{-}2 \end{pmatrix} \\
(Hf)(\pm 1, 0) &= \begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix} \\
(Hf)(\pm w, \pm w) &= \begin{pmatrix} 4 & 4 \\ 4 & 0 \end{pmatrix} \\
(Hf)(\pm w, \mp w) &= \begin{pmatrix} 4 & \text{-}4 \\ \text{-}4 & 0 \end{pmatrix}
\end{aligned}
$$

Damit:

- Maximum bei $(0,0)$
- Minimum bei $(\pm 1, 0)$
- Sattelpunkte bei $(\pm w, \pm w)$ und $(\pm w, \mp w)$

C.4. Vollständige Induktion

Aufgabe
Beweise, dass $\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$ gilt.

IA Für $n_0 = 1$ gilt: $1 = \frac{1\cdot(1+1)}{2} = \frac{2}{2} = 2$.

IV $\sum_{i=1}^{n} i = \frac{n\cdot(n+1)}{2}$

IB Wenn die IV für ein beliebiges aber festes n gilt, so gilt sie auch für $n+1$.

IS

$$\begin{aligned}
\sum_{i=1}^{n+1} i &= \frac{(n+1)\cdot(n+2)}{2} \\
\Leftrightarrow (n+1) + \sum_{i=1}^{n} i &= \frac{(n+1)\cdot(n+2)}{2} \\
\overset{IV}{\Leftrightarrow} (n+1) + \frac{n\cdot(n+1)}{2} = \frac{n^2+3n+2}{2} &= \frac{n^2+3n+2}{2} \\
\Rightarrow & \top
\end{aligned}$$

Damit ist die IV bewiesen.

C.5. Schnittmengenbestimmung

> **Aufgabe**
> Es seien $P = (1,\text{-}1,2)$, $Q = (1,1,1)$ und $\vec{n} = (1,2,3)^T$ gegeben. Bestimme den Schnittpunkt der Gerade durch P mit Richtung $\vec{n}$ und der Ebene in Q senkrecht zu $\vec{n}$.

1. Gerade in Parameterform bringen

$$\vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ \text{-}1 \\ 2 \end{pmatrix} + \gamma \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \tag{C.5.1}$$

$$x = 1 + \gamma \tag{C.5.2}$$

$$y = \text{-}1 + 2\gamma \tag{C.5.3}$$

$$z = 2 + 3\gamma \tag{C.5.4}$$

2. Ebene in paramterlose Form bringen

$$\langle \vec{x}, \vec{n} \rangle = \langle \vec{q}, \vec{n} \rangle$$

$$x + 2y + 3z = 6 \tag{C.5.5}$$

3. Einsetzen von C.5.2, C.5.3 und C.5.4 in C.5.5

$$\begin{aligned} 1 + \gamma + 2(\text{-}1 + 2\gamma) + 3(2 + 3\gamma) &= 6 \\ 5 + 14\gamma &= 6 \\ \gamma &= {}^{1}/_{14} \end{aligned}$$

4. Einsetzen von γ in C.5.1

$$\begin{aligned} \vec{x} &= \begin{pmatrix} 1 \\ \text{-}1 \\ 2 \end{pmatrix} + \frac{1}{14} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \\ &= \frac{1}{14} \begin{pmatrix} 15 \\ \text{-}12 \\ 31 \end{pmatrix} \end{aligned}$$

D. LaTeX Symbole

D.1. Griechische Buchstaben

α	\alpha
β	\beta
Γ, γ	\Gamma, \gamma
Δ, δ	\Delta, \delta
ϵ, ε	\epsilon, \varepsilon
ζ	\zeta
η	\eta
Θ, θ, ϑ	\Theta, \theta, \vartheta
ι	\iota
κ	\kappa
Λ, λ	\Lambda, \lambda
μ	\mu
ν	\nu
Ξ, ξ	\Xi, \xi
Π, π, ϖ	\Pi, \pi, \varpi
ρ, ϱ	\rho, \varrho
Σ, σ, ς	\Sigma, \sigma, \varsigma
τ	\tau
Υ, υ	\Upsilon, \upsilon
Φ, ϕ, φ	\Phi, \phi, \varphi
χ	\chi
Ψ, ψ	\Psi, \psi
Ω, ω	\Omega, \omega

D.2. Sonstige Symbole

∂	\partial
∞	\infty
$\imath$	\imath
∇	\nabla
ℓ	\ell

$\ldots, \cdots$	`\dots, \cdots`

D.3. Verhältnisoperatoren

$<$	`<`
$\leq$	`\leq`
$\ll$	`\ll`
$\subset$	`\subset`
$\subseteq$	`\subseteq`
$=$	`=`
$\equiv$	`\equiv`
$\in$	`\in`
$>$	`>`
$\geq$	`\geq`
$\gg$	`\gg`
$\supset$	`\supset`
$\supseteq$	`\supseteq`
$\approx$	`\approx`
$\cong$	`\cong`
$\notin$	`\notin`

D.4. Operatoren

$+, -, \cdot, :, /$	`+, -, \cdot, :, /`
$\pm, \mp$	`\pm, \mp`
$\times$	`\times`
$\vee, \wedge, \lor, \land$	`\vee, \wedge, \lor, \land`
$\neg, \top, \bot$	`\neg, \top, \bot`
$\rightarrow, \leftarrow, \leftrightarrow, \mapsto$	`\rightarrow, \leftarrow, \leftrightarrow, \mapsto`
$\Rightarrow, \Leftarrow, \Leftrightarrow$	`\Rightarrow, \Leftarrow, \Leftrightarrow`
$\oplus, \ominus, \odot, \oslash$	`\oplus, \ominus, \odot, \oslash`
$\circ$	`\circ`
$\otimes$	`\otimes`
$\cap, \cup$	`\cap, \cup`
$\exists, \nexists, \forall$	`\exists, \nexists, \forall`
$\emptyset, \varnothing$	`\emptyset, \varnothing`
$\setminus$	`\setminus`

D.5. Begrenzungen und Klammern

$\vert, \Vert$	`|, |`
$\{, \}$	`{, }`
$\langle, \rangle$	`\langle, \rangle`
$\lceil, \rceil$	`\lceil, \rceil`
$\lfloor, \rfloor$	`\lfloor, \rfloor`

D.6. Große Operatoren, Dekorationen

$\int$	`\int (mit ^ und _ verwenden)`
$\sum$	`\sum (mit ^ und _ verwenden)`
$\prod$	`\prod (mit ^ und _ verwenden)`
$\vec{a}$	`\vec`
a^b	`a^b`
$\frac{a}{b}$	`\frac`
$\lim\limits_{n\to\infty}$	`\underset`
$\overset{!}{=}$	`\overset`
$\underbrace{a+b}$	`\underbrace`
a_b	`a_b`
$\sqrt{a+b}$	`\sqrt`
$\sqrt[n]{a}$	In LyX: Alt+M, R
$\left(\frac{a}{b}\right)$	In LyX: Alt+M, (
$\sphericalangle, \measuredangle$	`\sphericalangle, \measuredangle`
a_c^b	`a^b, (Taste runter), _c`

D.7. Abstände

$a\,b$	`\,`
$a\:b$	`\:`
$a\;b$	`\;`
$a\quad b$	`\quad`
$a\qquad b$	`\qquad`

D.8. Zahlenmengen

$\mathbb{N}$	`\mathbb N`
$\mathbb{Z}$	`\mathbb Z`
$\mathbb{Q}$	`\mathbb Q`
$\mathbb{R}$	`\mathbb R`
$\mathbb{C}$	`\mathbb C`

D.9. Die MathBB-Zeichen im Überblick

$\Game\ \mathbb{A}$	a, A
$\mathbb{B}$	B
$\mathbb{C}$	C
$\mathbb{D}$	D
$\mathbb{E}$	E
$\mho\ \mathbb{F}$	f, F
$\eth\ \mathbb{G}$	g, G
$\eqsim\ \mathbb{H}$	h, H
$\beth\ \mathbb{I}$	i, I
$\gimel\ \mathbb{J}$	j, J
$\daleth\ \mathbb{K}$	k, K
$\lessdot\ \mathbb{L}$	l, L
$\gtrdot\ \mathbb{M}$	m, M
$\ltimes\ \mathbb{N}$	n, N
$\rtimes\ \mathbb{O}$	o, O
$\shortmid\ \mathbb{P}$	p, P
$\shortparallel\ \mathbb{Q}$	q, Q
$\smallsetminus\ \mathbb{R}$	r, R
$\thicksim\ \mathbb{S}$	s, S
$\thickapprox\ \mathbb{T}$	t, T
$\approxeq\ \mathbb{U}$	u, U
$\succapprox\ \mathbb{V}$	v, V
$\precapprox\ \mathbb{W}$	w, W
$\curvearrowleft\ \mathbb{X}$	x, X
$\curvearrowright\ \mathbb{Y}$	y, Y
$\digamma\ \mathbb{Z}$	z, Z
$\nVdash\nvDash\nVDash$	1-3
$\ntrianglerighteq\ntrianglelefteq\ntriangleleft\ntriangleright$	4-7
$\nleftarrow\nrightarrow\nvdash$	8-0

D.10. Beispiele

$$\psi_{j,k}(x) := 2^{j/2}\,\psi(2^j\,x-k)$$

```
\psi_{j,k}(x):=2^{j/2}\,\psi(2^j\,x-k)
```

$$f\mapsto c(f):=\mathcal W[f]=\left(c_{j,k}(f):=\langle f,\,\psi_{j,k}\rangle:\;j,k\in\mathbb{Z}\right)$$

```
f\mapsto c(f):=\mathcal W[f]=
\left(c_{j,k}(f):=\langle f,\,\psi_{j,k}\rangle:
\;j,k\in\mathbb{Z}\right)
```

$$\frac{\sqrt{2\pi}}{T}\sum_{k=-\infty}^{\infty}\delta\left(\omega-k\frac{2\pi}{T}\right)$$

```
\frac{\sqrt{2\pi}}{T}\sum_{k=-\infty}^{\infty}
\delta\left(\omega-k\frac{2\pi}{T}\right)
```

D.10.1. Vergleich eingebettete Formeln und abgesetzte Formeln

Wechsel zwischen den Modi in LyX mit Strg+Shift+M. Unterschiede in den Grenzen und Größen der großen Operatoren beachten.

Eingebettet
$\frac{\sqrt{2\pi}}{T}\sum_{k=-\infty}^{\infty}\delta\left(\omega-k\frac{2\pi}{T}\right)$
$\int_{-\infty}^{\infty}G(f)e^{i2\pi ft}\mathrm{d}f$
$\sum_a^b f(x)\quad\int_a^b f(x)\,\mathrm{dx}$
Abgesetzt

$$\frac{\sqrt{2\pi}}{T}\sum_{k=-\infty}^{\infty}\delta\left(\omega-k\frac{2\pi}{T}\right)$$

$$\int_{-\infty}^{\infty}G(f)e^{i2\pi ft}\mathrm{d}f$$

$$\sum_a^b f(x)\quad\int_a^b f(x)\,\mathrm{dx}$$

E. Stichworte

Mathematik

IT Grundlagen

Glossar

ALU Arithmetic Logic Unit (Seite 72)

amd64 AMD: Bezeichnung für die 64-Bit-Generation der Prozessoren. (Seite 96)

ASM Assembler (Seite 96)

BDSG Bundesdatenschutzgesetz (Seite 95)

CD-ROM Compact Disc – Read Only Memory (Seite 84)

CISC Complex Instruction Set Computer (Seite 75)

CPU Central Processing Unit (Seite 72)

CRC Cyclic Redundancy Check (Seite 70)

DMA Direct Memory Access (Seite 74)

DRAM Dynamic Random Access Memory (Seite 73)

EFM Eight to Fourteen Modulation (Seite 84)

FCFS First Come, First Serve (Seite 79)

FIFO First In, First Out (Seite 79)

HAL Hardware Abstraction Layer (Seite 76)

ia-64 Intel: Bezeichnung für die 64-Bit-Generation der Prozessoren. (Seite 96)

LFU Least Frequently Used (Seite 80)

LGS Lineares Gleichungssystem (Seite 26)

LRU Least Recently Used (Seite 80)

MD5 Message Digest Algorithm 5 (Seite 70)

MLC Multi Level Cells (Seite 83)

NRU Not Recently Used (Seite 80)

PC Program Counter (Seite 73)

RAID Redundant Array of Independent Discs (Seite 84)

RAM Random Access Memory (Seite 72)

RISC Reduced Instruction Set Computer (Seite 75)

ROM Read Only Memory (Seite 72)

SHA Secure Hash Algorithm (Seite 70)

SJF Shortest Job First (Seite 79)

SLC Single Level Cells (Seite 83)

SRAM Static Random Access Memory (Seite 73)

SSD Solid State Disk (Seite 83)

x64 Generelle Bezeichnung für die 64-Bit-Generation der Prozessoren. (Seite 96)

x86 Bezeichnung für die 32-Bit-Generation der Prozessoren. Die ersten 32-Bit-Prozessoren von Intel hatten noch Zahlen als Bezeichnung, der erste seiner Art war der 80386. Es folgten der 80486 und der als Pentium I bekannte 80586. (Seite 96)

Literaturverzeichnis

[Bun90] Bundesregierung: Bundesdatenschutzgesetz Â§9. Version: 1990. http://www.gesetze-im-internet.de/bdsg_1990/__9.html. In: *Bundesdatenschutzgesetz*. Bundesregierung, 1990

[May10] Mayiopoulos, Dipl.-Math. P.: *Analysis I*. 2010 http://www.rz.rwth-aachen.de/aw/cms/rz/Zielgruppen/rz_auszubildende/veranstaltungen/mathematik/pflichtkurse/~rcm/analysis_i/?lang=de

[MW11] Merziger, G. ; Wirth, T.: *Repetitorium höhere Mathematik*. 6. Auflage. Binomi Verlag, 2011. – ISBN 978–3–923923–34–2

[Pap11] Papula, Lothar: *Mathematik für Ingenieure und Naturwissenschaftler Band 3*. Vieweg + Teubner Verlag, 2011 (6. Auflage). – ISBN 978–3–8345–1227–8

[Pfl11] Pflug, Hans J.: *1. Programmiersprache Java*. 2011 http://www.rz.rwth-aachen.de/aw/cms/rz/Zielgruppen/rz_auszubildende/veranstaltungen/programmiersprachen/Pflichtkurse/~qxt/java/?lang=de

[PW11] Pflug, Hans J. ; Willemsen, Benno: *Lineare Algebra 1/2*. 2011 http://www.rz.rwth-aachen.de/aw/cms/rz/Zielgruppen/rz_auszubildende/veranstaltungen/mathematik/pflichtkurse/~rcp/lineare_algebra_i/?lang=de

[VKKS11] Voss, Dr. A. ; Küppers, B. ; Kraft, Prof. Dr. B. ; Sczimarowsky, M.: *IT Grundlagen Semesterkurs*. 2011 http://www.rz.rwth-aachen.de/aw/cms/rz/Zielgruppen/rz_auszubildende/veranstaltungen/informatik/pflichtkurse/~qvw/it_grundlagen/?lang=de

[Wik04] Wikipedia: *Hammingcode — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Hammingcode&oldid=7094873. Version: 2004. – [Online; Stand 20. April 2012]

[Wik12a] WIKIBOOKS: *Assembler-Programmierung für x86-Prozessoren — Wikibooks.* 2012 `http://de.wikibooks.org/w/index.php?title=Assembler-Programmierung_f%C3%BCr_x86-Prozessoren&oldid=631075`. – [Online; Stand 20. April 2012]

[Wik12b] WIKIPEDIA: *Assemblersprache — Wikipedia, Die freie Enzyklopädie.* `http://de.wikipedia.org/w/index.php?title=Assemblersprache&oldid=102180077`. Version: 2012. – [Online; Stand 20. April 2012]

[Wik12c] WIKIPEDIA: *Einerkomplement — Wikipedia, Die freie Enzyklopädie.* `http://de.wikipedia.org/w/index.php?title=Einerkomplement&oldid=100846954`. Version: 2012. – [Online; Stand 20. April 2012]

[Wik12d] WIKIPEDIA: *Festkommazahl — Wikipedia, Die freie Enzyklopädie.* `http://de.wikipedia.org/w/index.php?title=Festkommazahl&oldid=101255745`. Version: 2012. – [Online; Stand 20. April 2012]

[Wik12e] WIKIPEDIA: *Gleitkommazahl — Wikipedia, Die freie Enzyklopädie.* `http://de.wikipedia.org/w/index.php?title=Gleitkommazahl&oldid=101255894`. Version: 2012. – [Online; Stand 20. April 2012]

[Wik12f] WIKIPEDIA: *IEEE 754 — Wikipedia, Die freie Enzyklopädie.* `http://de.wikipedia.org/w/index.php?title=IEEE_754&oldid=101722017`. Version: 2012. – [Online; Stand 20. April 2012]

[Wik12g] WIKIPEDIA: *Zweierkomplement — Wikipedia, Die freie Enzyklopädie.* `http://de.wikipedia.org/w/index.php?title=Zweierkomplement&oldid=101985901`. Version: 2012. – [Online; Stand 20. April 2012]