Skriptum

2021/22

Wozu ist Mathe gut?

Drei exemplarische Anwendungen

Gerald und Susanne Teschl

Gerald Teschl
Universität Wien
Fakultät für Mathematik
Oskar-Morgenstern-Platz 1
1090 Wien, Österreich
Gerald.Teschl@univie.ac.at
http://www.mat.univie.ac.at/~gerald/

Susanne Teschl
Fachhochschule Technikum Wien
Höchstädtplatz 5
1200 Wien, Österreich
Susanne.Teschl@technikum-wien.at
http://staff.technikum-wien.at/~teschl/

Ziel dieses Skriptums ist es exemplarisch anhand dreier Beispiele aufzuzeigen, wo Mathematik überall Verwendung findet.

Mathematica

Begleitend ist ein Mathematica-Notebook verfügbar. Dabei geht es darum zu belegen, dass die im Skriptum vorgestellten Methoden auch wirklich so in der Praxis funktionieren. Natürlich kann auch jedes andere Computeralgebrasystem verwendet werden. Zum Öffnen des Mathematica-Notebooks (`*.nb`) benötigen Sie Mathematica oder CDF Player (kostenlos). Ein Teil der Funktionalität von Mathematica ist auch über die Webseite WolframAlpha frei verfügbar.

Eine Bitte...

Druckfehler sind wie Unkraut. Soviel man auch jätet, es bleiben immer ein paar übrig und so sind auch in diesem Skriptum trotz aller Sorgfalt sicher noch ein paar unentdeckte Fehler. Wir bitten Sie daher, uns diese mitzuteilen (auch wenn sie noch so klein sind).

Danke

Für die Rückmeldung von Druckfehlern in Vorversionen bedanken wir uns bei: Florian Jandl, Hubert Koizar, Johanna Rohnefeld, Stefan Weichhart

Version: 1. März 2023

Inhaltsverzeichnis

Kapitel 1

Geheimschriften knacken für Anfänger

1.1 Monoalphabetische Verschlüsselung

Kryptographie ist eine uralte Wissenschaft, die bis zur Mitte des 20. Jahrhunderts hauptsächlich für militärische Zwecke eingesetzt worden ist. Heute werden Verträge, Kundendaten, und andere vertrauliche Informationen auf Rechnern gespeichert und über das Internet ausgetauscht. Der Datenaustausch interessiert nicht nur Geheimdienste, sondern auch Konkurrenzfirmen. Die Sicherheit von Firmennetzen gegenüber Angriffen von außen wird aber noch stark vernachlässigt, obwohl Wissen und Technik der Datensicherheit heute öffentlich zugänglich sind. Heute stehen mit der in den Siebzigerjahren erfundenen Public-Key-Kryptographie und den modernen Blockchiffren starke und mittlerweile bewährte Werkzeuge zur Sicherung von Daten und Vertraulichkeit bereit. Diese Methoden beruhen teilweise auf genial einfachen Ideen der Mathematik — beginnend bei der modularen Arithmetik bis hin zu endlichen Körpern. Wir schränken uns hier auf klassische Verfahren ein und zeigen wie sie mit etwas Mathematik leicht geknackt werden können.

Eine Verschlüsselung ist eine umkehrbare Vorschrift die einen *Klartext* in einen *Geheimtext* überführt. Eine monoalphabetische Verschlüsselung E ist eine umkehrbare Abbildung von Klartextbuchstaben in Geheimtextbuchstaben. Gehen wir nun einfachheitshalber davon aus, dass unser *Alphabet* gleich der Menge der Buchstaben $\{\mathtt{A}, \ldots, \mathtt{Z}\}$ (ohne Sonderzeichen, Umlaute, etc.) ist:

$$\begin{aligned} E \quad : \quad & \{\mathtt{A}, \ldots, \mathtt{Z}\} \to \{\mathtt{A}, \ldots, \mathtt{Z}\} \\ & x \mapsto y = E(x) \end{aligned}$$

Eine Wertetabelle der Verschlüsselungsvorschrift könnte dann zum Beispiel so aussehen:

Klartextbuchstabe x:	A	B	C	D	E	F	G	...	W	X	Y	Z
Geheimtextbuchstabe y:	Q	W	E	R	T	Y	U	...	V	B	N	M

Also jedes `A` im Klartext wird zu einem `Q` im Geheimtext, jedes `B` zu einem `W`, etc.

Für die mathematische Beschreibung bzw. die Implementierung der Verschlüsselungsvorschrift im Computer ist es notwendig, die Buchstaben durchzunummerie-

ren und durch die Ziffern 0 bis 25 zu ersetzen; also allgemein $\mathbb{Z}_n = \{0, \ldots, n-1\}$ als Alphabet zu verwenden.

Jede monoalphabetische Verschlüsselung entspricht einer Permutation (Anordnung unter Berücksichtigung der Reihenfolge) der Zeichen des Alphabets. Bei n Buchstaben gibt es $n!$ Permutationen der Zeichen, also $n!$ Möglichkeiten für eine monoalphabetische Verschlüsselung (die identische Abbildung des Alphabets auf sich selbst wird zwar niemand zur Verschlüsselung verwenden, sie ist hier aber mitgezählt). Das Rufzeichen in $n!$ wird als „n Fakultät“ gelesen und ist als Produkt der Zahlen von 1 bis n definiert: $n! = 1 \cdot 2 \cdots (n-1) \cdot n$.

Warum? — Für den ersten Buchstaben im Geheimalphabet haben Sie die freie Wahl und somit n Möglichkeiten. Für den zweiten Buchstaben fällt eine Möglichkeit weg (einen haben Sie ja schon vergeben), also bleiben $n-1$ Möglichkeiten. Für den dritten Buchstaben gibt es $n-2$ Möglichkeiten usw. bis für den letzten Buchstaben nur noch einer übrig ist. Die Anzahl der Möglichkeiten insgesamt ist nun das Produkt der Möglichkeiten für die einzelnen Buchstaben, also $n \cdot (n-1) \cdots 1 = n!$.

Für $n = 26$ Zeichen sind das zum Beispiel $26! \approx 4 \cdot 10^{26}$ monoalphabetische Chiffren. Trotz dieser großen Zahl sind monoalphabetische Chiffren für natürliche Sprachen (also deutsch, englisch, …) relativ leicht zu knacken. Natürliche Sprachen weisen nämlich immer eine bestimmte *Häufigkeitsverteilung* der Buchstaben auf, die sich auf den Geheimtext überträgt. Wir werden monoalphabetische Verfahren, obwohl sie unsicher sind, aber trotzdem näher besprechen, da man an ihrem Beispiel einen guten Einblick in die Kunst der kryptoanalytischen Angriffe bekommen kann.

Bereits Julius Cäsar hat für vertrauliche Nachrichten eine monoalphabetische Verschlüsselung verwendet: Er hat das Alphabet zyklisch um drei Stellen verschoben:

Klartextbuchstabe x:	A	B	C	D	E	F	G	…	W	X	Y	Z
Geheimtextbuchstabe y:	D	E	F	G	H	I	J	…	Z	A	B	C

Zur Verschlüsselung wird jeder Buchstabe durch den darunterliegenden, zur Entschlüsselung durch den darüberliegenden ersetzt.

Allgemein nennt man einen Algorithmus, bei dem ein Buchstabe durch den um e Stellen weiter liegenden Buchstaben ersetzt wird, eine **Verschiebechiffre** oder **Cäsarverschiebung**. Dabei kann e als der Schlüssel aufgefasst werden. Für die mathematische Beschreibung muss zunächst jeder Buchstabe des Alphabets durch die entsprechende Ziffer in $\mathbb{Z}_{26}$ codiert werden (A = 0, B = 1, …, Z = 25). Dann lautet die Verschlüsselungsvorschrift:

$$y = (x + e) \bmod 26$$

Das nachgesetzte „mod 26“ bedeutet, dass hier modulo 26 zu rechnen ist. Jede Zahl die größer als 25 ist wird also durch ihren Rest bei Division mit 26 ersetzt. Z.B. ist $26 \bmod 26 = 0$, $27 \bmod 26 = 1$, $28 \bmod 26 = 2$, usw. Das entspricht genau der gewünschten zyklischen Verschiebung.

Entschlüsselt wird, indem in dieselbe Richtung um $d = 26 - e$ geschoben wird:

$$x = (y + d) \bmod 26.$$

Beispiel 1.1 (→CAS) Cäsarverschiebung
Verschlüsseln Sie `DIESPINNENDIEGALLIER` mittels einer Verschiebechiffre um $e = 3$ Stellen.

Lösung zu 1.1 Dazu codieren wir zunächst den Klartext in Zahlen aus $\mathbb{Z}_{26}$: `D` $= 3$, `I` $= 8$, usw., letztendlich:

$$\{3, 8, 4, 18, 15, 8, 13, 13, 4, 13, 3, 8, 4, 6, 0, 11, 11, 8, 4, 17\}$$

Die Verschlüsselung ist nun einfach. Für jede Klartextzahl x ist $y = (x+3) \bmod 26$ die zugehörige Geheimtextzahl:

$$\{6, 11, 7, 21, 18, 11, 16, 16, 7, 16, 6, 11, 7, 9, 3, 14, 14, 11, 7, 20\}$$

Das wollen wir wieder in Buchstaben sehen:

`GLHVSLQQHQGLHJDOOLHU`

Beachten Sie, dass – wie bei jeder monoalphabetischen Verschlüsselung – das mehrfache Auftreten eines Buchstaben im Klartext sich auf den Geheimtext überträgt: `NN` im Klartext wird z.B. zu `QQ` im Geheimtext. ■

Sie kennen vielleicht aus dem Internet die Verschiebechiffre **ROT13**, die einen Buchstaben aus dem Alphabet $\{a, \ldots, z\}$ durch den um genau $e = 13$ Stellen verschobenen Buchstaben ersetzt. Diese einfache Verschlüsselung ist nicht zum Schutz von Daten gedacht, sondern soll vor einem versehentlichen Mitlesen schützen – so wie eine Zeitung die Auflösung eines Rätsels oft kopfstehend druckt. Warum wird gerade um 13 Stellen verschoben? — Weil dann mit demselben Schlüssel wieder entschlüsselt werden kann: $d = 26 - e = 13$.

Wechseln wir nun die Seiten und machen uns an die Kryptoanalyse. Gehen wir zunächst davon aus, dass wir einen Geheimtext vor uns haben, von dem wir wissen, dass er durch Verschiebe-Verschlüsselung eines deutschen Textes entstand, zum Beispiel `INJXJWYJCYNXYSNHMYRJMWLJMJNR`.

Es gibt 26 mögliche Verschiebe-Verschlüsselungen. Eine erste Möglichkeit wäre, alle Schlüssel e durchzuprobieren (Brute-Force), und zu prüfen, für welche Verschiebung sich aus dem Geheimtext ein sinnvoller Klartext ergibt. Das ist für einen Computer nicht aufwändig, aber es geht noch einfacher. Die Buchstaben eines deutschen Textes treten ja mit bestimmten Häufigkeiten auf:

Mittlere Buchstabenhäufigkeiten (Prozent) der deutschen Sprache

a	6.51	b	1.89	c	3.06	d	5.08	e	17.40	f	1.66
g	3.01	h	4.76	i	7.55	j	0.27	k	1.21	l	3.33
m	2.53	n	9.78	o	2.51	p	0.79	q	0.02	r	7.00
s	7.27	t	6.15	u	4.35	v	0.67	w	1.89	x	0.03
y	0.04	z	1.13								

So ist also zum Beispiel `E` im Mittel der häufigste Buchstabe in einem deutschen Text. Daraus können wir schließen, dass in einem Geheimtext der Buchstabe, der dem Klartextbuchstaben `E` entspricht, am häufigsten vorkommen sollte. Diese statistische Methode ist natürlich umso zuverlässiger, je länger der Geheimtext ist, den man untersucht.

Beispiel 1.2 (→CAS) Kryptoanalyse einer Verschiebechiffre
Sie erhalten den Geheimtext `INJXJWYJCYNXYSNHMYRJMWLJMJNR`, von dem Sie wissen, dass es sich um einen deutschen Text handelt, der durch eine Verschiebechiffre $y = (x + e) \bmod 26$ verschlüsselt wurde. Entschlüsseln Sie den Geheimtext.

Lösung zu 1.2 Wenn wir bei einer Verschiebechiffre die Verschlüsselung eines *einzigen* Buchstaben kennen, so kennen wir bereits die Verschiebung e. Der häufigste Buchstabe des Geheimtexts ist `J`. Das legt nahe, dass er dem Klartextbuchstaben `E` entspricht. (Dieser Geheimtext ist nicht sonderlich lang, wir könnten also mit dieser Annahme leicht danebenliegen.) In diesem Fall wäre die Verschiebung $e = 5$, die Verschlüsselungsvorschrift also $y = (x + 5) \bmod 26$, und die Entschlüsselungsvorschrift $x = (y + 21) \bmod 26$. Damit ergibt sich: `DIESERTEXTISTNICHTMEHRGEHEIM`. Da wir einen sinnvollen Text erhalten, ist der Code geknackt. Hätten wir keinen sinnvollen Text erhalten, so hätten wir als nächstes die Zuordnung Geheimtext-`J` ist Klartext-`N` (das ist der Buchstabe, der in einem deutschen Text im Mittel am zweithäufigsten auftritt) probieren können. ■

Das war nicht schwer, denn wir wussten, dass es eine Verschiebechiffre ist. Wenn man nicht weiß, *welcher* monoalphabetische Verschlüsselungsalgorithmus verwendet wurde, dann muss man schon etwas mehr arbeiten (es gibt 26! Möglichkeiten). Man muss dann für jeden Klartextbuchstaben den zugehörigen Geheimtextbuchstaben finden. Wieder hilft Statistik: Die häufigsten Buchstaben des Geheimtextes werden gezählt, und mithilfe der bekannten mittleren Häufigkeiten im deutschen Text kann eine Zuordnung versucht werden. Ein Problem dabei ist, dass in der Regel die gezählten Häufigkeiten von einzelnen Zeichen im Geheimtext nahe beieinander liegen werden, sodass eine eindeutige Zuordnung von Klartext- zu Geheimtextbuchstaben zunächst nicht möglich ist. Abhilfe: Man kann zusätzlich die Häufigkeiten von **Buchstabenpaaren** (sogenannten **Bigrammen**) zu Hilfe nehmen:

Mittlere Bigrammhäufigkeiten (Prozent) der deutschen Sprache

en	3.88	er	3.75	ch	2.75	te	2.26	de	2.00	nd	1.99
ei	1.88	ie	1.79	in	1.67	es	1.52				

In deutschen Texten kommen also zum Beispiel die Bigramme `EN` oder `CH` besonders häufig vor. Beispiel: wenn die Buchstaben `G` und `R` im Geheimtext am häufigsten vorkommen, so kann man vermuten, dass der eine `E` und der andere

`N` bedeutet. Wenn diese Häufigkeiten sehr nahe beieinander liegen, kann man die richtige Zuordnung finden, indem man zusätzlich die Häufigkeiten von `GR` und `RG` im Geheimtext ermittelt. Das häufigere Bigramm wird dann laut obiger Bigramm-Tabelle `EN` entsprechen.

Eine Kombination von Brute-Force-Angriff und statistischer Analyse, die das Entschlüsselungsverfahren aufgrund einer hohen Trefferquote weitestgehend automatisiert, ist das folgende Verfahren: Man zählt die Buchstabenhäufigkeiten des Geheimtexts aus und sucht jenen Schlüssel e (bzw., falls das Verschlüsselungsverfahren nicht bekannt ist, jene Permutation), für den die Summe der quadratischen Abweichungen zu den Buchstabenhäufigkeiten der deutschen Sprache minimal wird.

Beispiel, das die Idee verdeutlichen soll: Angenommen, das Klartext- und Geheimtextalphabet besteht aus den 4 Buchstaben `A, B, C, D`, und in einem typischen Klartext sind die relativen Häufigkeiten von `A, B, C, D` gleich $50, 30, 15$ bzw. 5%. (Das entspricht der Referenztabelle mit den Buchstabenhäufigkeiten der deutschen Sprache.) Auszählung der Buchstaben im Geheimtext ergibt z.B. die Häufigkeiten $14, 52, 4, 30$. Wenn der Computer alle 24 möglichen Zuordnungen durchprobiert, so wird sich herausstellen, dass für die Zuordnung

$$\begin{aligned}\texttt{A} &\mapsto F_e(\texttt{A}) = \texttt{B}\\ \texttt{B} &\mapsto F_e(\texttt{B}) = \texttt{D}\\ \texttt{C} &\mapsto F_e(\texttt{C}) = \texttt{A}\\ \texttt{D} &\mapsto F_e(\texttt{D}) = \texttt{C}\end{aligned}$$

die Summe der (quadratischen) Abweichungen der entsprechenden Häufigkeiten,

$$(50-52)^2 + (30-30)^2 + (15-14)^2 + (5-4)^2,$$

minimal ist. (Dass das höchstwahrscheinlich die richtige Zuordnung ist, sehen wir hier auch durch „Hinsehen“.)

Diese Strategie funktioniert auch bei Sprachen, bei denen es keinen eindeutig häufigsten Buchstaben gibt, und auch bei relativ kurzen Texten. Der Preis, den man dafür bezahlt, ist der höhere Aufwand.

Beispiel 1.3 (→CAS) Minimale quadratische Abweichung der Buchstabenhäufigkeiten
Entschlüsseln Sie den Geheimtext

```
PUQMZFIADFMGRPUQRDMSQZMOTPQYXQNQZPQYGZU
HQDEGYGZPPQYSMZLQZDQEFUEFLIQUGZPHUQDLUS
```

von dem Sie wissen, dass es sich um einen deutschen Text handelt, der mit einer Verschiebechiffre verschlüsselt wurde.

Lösung zu 1.3 Mithilfe von `Mathematica` finden wir die minimale Abweichung bei $e = 12$. Versuchen wir, mit dem zugehörigen $d = 26 - 12 = 14 \pmod{26}$ zu entschlüsseln, so erhalten wir

```
DIEANTWORTAUFDIEFRAGENACHDEMLEBENDEMUNIVERSUMUNDDEMGANZEN
RESTISTZWEIUNDVIERZIG
```

Da wir einen *sinnvollen* Text erhalten, wurde der Text also tatsächlich um 12 Stellen verschoben. Wäre der Text sinnlos gewesen, hätten wir als nächstes die Verschiebung mit der zweitkleinsten quadratischen Abweichung probieren können. ■

Übrigens kann man die Buchstabenhäufigkeiten auch dazu verwenden um zu erkennen, dass ein Text monoalphabetisch verschlüsselt wurde: Bei einer monoalphabetischen Verschlüsselung werden die Buchstaben ja nur durchgemischt und somit werden auch die Buchstabenhäufigkeiten nur durchgemischt (aber nicht verändert). Somit kann man die (am besten geordneten) Buchstabenhäufigkeiten im Geheimtext mit den Buchstabenhäufigkeiten der vermuteten Sprache vergleichen. Das ist in Abbildung 1.1 zu sehen. Der linke Text wurde monoalphabetisch verschlüsselt und der rechte Text mit einer polyalphabetischen Verschlüsselung wie wir sie im nächsten Abschnitt besprechen werden. Bei der polyalphabetischen Verschlüsselung werden mehrere Buchstaben zusammengefasst und dadurch sind die Buchstabenhäufigkeiten im Geheimtext weniger ausgeprägt. Das ist in Abbildung 1.1 zum Beispiel am deutlich kleineren Wert für den häufigsten Buchstaben zu sehen.

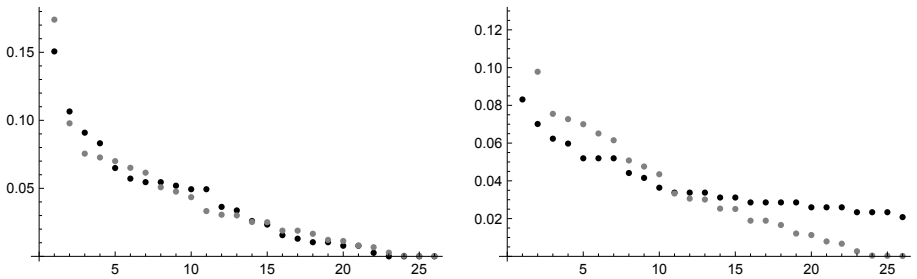

Abbildung 1.1: Vergleich der Buchstabenhäufigkeiten bei einer monoalphabetischen Verschlüsselung (links) und einer polyalphabetischen Verschlüsselung (rechts). Zusätzlich sind jeweils grau die deutschen Buchstabenhäufigkeiten eingezeichnet.

Quantitativ kann man die Abweichung mit Hilfe des **Koinzidenzindex** (auch **Friedman-Charakteristik** nach dem russisch-US-amerikanischen Kryptologen William F. Friedman (1891–1969) und seinem Schüler dem amerikanischen Mathematiker und Kryptologen Solomon Kullback (1907–1994))

$$K = \sum_{i=0}^{25} h_i^2 = h_0^2 + h_1^2 + \cdots + h_{25}^2$$

beschreiben, wobei h_i die relative Häufigkeit des i'ten Buchstaben im untersuchten Text ist (also Anzahl des i'ten Buchstaben im Text geteilt durch die Länge des

Texts). Der Koinzidenzindex liegt zwischen $\frac{1}{26}$ und 1 und ist um so größer, um so ausgeprägter die Häufigkeiten sind, d.h., um so weiter sie von einer Gleichverteilung entfernt sind.

Warum? — Unter Verwendung von $\sum_{i=0}^{25} h_i = 1$ und $\sum_{i=0}^{25} 1 = 26$ erhalten wir $\sum_{i=0}^{25}(h_i - \frac{1}{26})^2 = \sum_{i=0}^{25} h_i^2 - \frac{2}{26}\sum_{i=0}^{25} h_i + \frac{1}{26^2}\sum_{i=0}^{25} 1 = \sum_{i=0}^{25} h_i^2 - \frac{1}{26}$ also

$$\sum_{i=0}^{25} h_i^2 = \frac{1}{26} + \sum_{i=0}^{25} \left(h_i - \frac{1}{26}\right)^2.$$

Daraus ersieht man $K \geq \frac{1}{26}$ und die Abweichung entspricht genau der quadratischen Abweichung von der Gleichverteilung, bei der jeder Buchstabe mit der Häufigkeit $\frac{1}{26}$ auftritt. Wegen $h_i^2 \leq h_i$ (beachte $0 \leq h_i \leq 1$) folgt $\sum_{i=0}^{25} h_i^2 \leq \sum_{i=0}^{25} h_i = 1$ mit Gleichheit genau dann wenn $h_i^2 = h_i$ für alle i gilt. Letzteres impliziert $h_i \in \{0, 1\}$ und das kann nur eintreten, wenn der Text aus genau einem Buchstaben besteht.

Mit den Buchstabenhäufigkeiten der deutschen Sprache erhalten wir den Erwartungswert für einen deutschen Text:

$$K_d = 0.076.$$

Für den linken Text aus Abbildung 1.1 erhält man $K = 0.075$ und für den rechten $K = 0.054$.

Diese statistischen Methoden sind auch für jede andere natürliche Sprache anwendbar, bzw. allgemein für jeden Klartext, bei dem zu erwarten ist, dass die Klartextzeichen im Mittel mit bestimmten Häufigkeiten auftreten. Bei Anwendungen im Computerbereich wird man sich zum Beispiel nicht auf die Buchstaben `A` bis `Z` beschränken. Da alle Daten üblicherweise als eine Folge von Bytes vorliegen, bietet es sich an, die Zahlen von 0 bis 255, also $\mathbb{Z}_{256}$, als Alphabet zu nehmen. Solange die Verteilung dieser Klartextzeichen (also der Bytes) nicht zufällig ist, lassen sich die oben beschriebenen statistischen Methoden zur Kryptoanalyse anwenden. Die zu erwartenden mittleren Häufigkeiten der einzelnen Klartextzeichen werden ermittelt, indem ein repräsentativer Musterklartext ausgezählt wird. Bei einem C-Programm werden zum Beispiel die geschwungenen Klammern „{“ und „}“ besonders häufig auftreten.

Da vielen Softwareentwicklern bitweise logische Operationen vertrauter sind als modulare Arithmetik, wird oft das logische xor zur monoalphabetischen Verschlüsselung verwendet. Das Klartextzeichen $x \in \mathbb{Z}_{256}$ wird dabei mithilfe des Schlüssels $e \in \mathbb{Z}_{256}$ nach der Vorschrift

$$y = x \oplus e$$

verschlüsselt. Entschlüsselt wird mit $x = y \oplus e$.

Beispiel 1.4 (→CAS) Monoalphabetische Verschlüsselung mit xor
Verschlüsseln Sie den Klartext

Streng geheim!

mittels xor mit dem Schlüssel $e = 127$.

Lösung zu 1.4 Der Klartext lautet im ASCII-Code: $83, 116, 114,\ 101, 110, 103,\ 32, 103, 101,\ 104, 101, 105, 109, 33$. Zur Verschlüsselung müssen wir alle Zahlen ins Dualsystem umwandeln und dann bitweise das logische xor bilden: $x = 83 = (1010011)_2$ und $e = 127 = (1111111)_2$ ergibt $y = 83 \oplus 127 = (0101100)_2 = 44$ (wegen $0 \oplus 0 = 1 \oplus 1 = 0$ und $1 \oplus 0 = 0 \oplus 1 = 1$). Analog für die weiteren Buchstaben, sodass der Geheimtext im ASCII-Code lautet: $44, 11, 13,\ 26, 17, 24,\ 95, 24, 26,\ 23, 26,\ 22, 18, 94$. Eine Umwandlung des Geheimtexts in einen String ist hier natürlich nicht sinnvoll, da nicht jedes mögliche Byte einem darstellbaren Zeichen entspricht. ■

Wie bei den Verschiebechiffren genügt bei diesem monoalphabetischen Verfahren die Zuordnung eines einzigen Geheimtextzeichens zum zugehörigen Klartextzeichen, um den Schlüssel zu bestimmen und damit die Verschlüsselung zu knacken.

Zusammenfassend kann man sagen, dass monoalphabetische Verschlüsselungen eines Klartextes, bei dem die einzelnen Klartextzeichen mit bestimmten Häufigkeiten auftreten, mit statistischen Methoden gebrochen werden können, und daher keinerlei Sicherheit bieten.

1.2 Polyalphabetische Verschlüsselung

Bei einer polyalphabetischen Verschlüsselung wird ein Klartextzeichen nicht stets zu demselben Geheimtextzeichen verschlüsselt. Dadurch übertragen sich die Häufigkeiten der Klartextbuchstaben nicht eins zu eins auf die der Geheimtextbuchstaben, und dadurch wird einem Angreifer die Arbeit erschwert.

Ein bekanntes, auch heute noch hin und wieder verwendetes (leider — da unsicher) polyalphabetisches Verfahren ist die so genannte **Vigenère-Verschlüsselung**. Sie wurde im 16. Jahrhundert vom französischen Diplomaten Blaise de Vigenère (1523–1596) vorgeschlagen und ist eine periodische Verwendung mehrerer Verschiebechiffren. Die Verschiebung eines Klartextzeichens hängt dabei von dessen Position im Klartext ab.

Die Idee ist einfach — gleich ein Beispiel: Wir verwenden als Schlüssel das Wort `VIGENERE`. Es wird periodisch fortgesetzt über den Klartext geschrieben. Jeder Buchstabe im Klartext wird nun mit einer anderen Verschiebechiffre verschlüsselt, und zwar mit jener, die zum darüberstehenden Buchstaben des Schlüsselwortes gehört:

Schlüsselwort	VIGENEREVIGENEREVIG
Klartext	DIESERTEXTISTGEHEIM
Geheimtext	YQKWRVKISBOWGKVLZQS

Der erste Buchstabe, D, wird also um V=21 Stellen auf Y verschoben. Der zweite Buchstabe I wird um I=8 Stellen auf Q verschoben, etc.

Das Verfahren kann so mathematisch beschrieben bzw. implementiert werden: Nachdem wir die Buchstaben in Zahlen umgewandelt haben, gilt $y_i = (x_i + e_i) \bmod 26$. Der Unterschied zur Verschiebechiffre ist also, dass jetzt für jeden Buchstaben x_i eine eigene Verschiebung e_i verwendet wird. Ist k die Schlüssellänge, und beginnen wir bei 0 zu zählen, so ist für den Klartextbuchstaben x_k wieder e_0 zu verwenden. Wir erhalten somit die **Vigenère-Verschlüsselungsvorschrift**

$$y_i = (x_i + e_{i \bmod k}) \bmod 26.$$

Entschlüsselt wird wieder mit diesem Vigenère-Algorithmus, nur mit einem anderen Schlüsselwort:

$$x_i = (y_i + d_{i \bmod k}) \bmod 26,$$

wobei $d_i = -e_i \pmod{26}$ gilt.

Beispiel 1.5 (→CAS) Vigenère-Verschlüsselung
Verschlüsseln Sie DIESERTEXTISTGEHEIM mit dem Schlüsselwort VIGENERE und entschlüsseln Sie danach wieder.

Lösung zu 1.5 Mit der Hand für den ersten Buchstaben: $x_0 = 3$ wird mit $e_0 = 21$ zu $y_0 = 24$ verschlüsselt, was dem Buchstaben Y entspricht. Analog wird $x_1 = 8$ mit $e_1 = 8$ zu $y_1 = 16$ verschlüsselt, also zu Q. Insgesamt ergibt sich

YQKWRVKISBOWGKVLZQS

Für die Entschlüsselung berechnen wir zunächst das Schlüsselwort zum Entschlüsseln. Zunächst mit der Hand: $d_0 = 26 - e_0 = 26 - 21 = 5$, also Buchstabe F. Analog $d_1 = 26 - e_1 = 26 - 8 = 18$, also Buchstabe S. Insgesamt ergibt sich FSUWNWJW und damit erhält man auch wieder den ursprünglichen Klartext. ■

Nun zur **Kryptoanalyse der Vigenère-Verschlüsselung**: Wenn für den Schlüssel bis zu n Stellen zur Verfügung stehen, so ergibt das bei einem 26-elementigen Alphabet $26 + 26^2 + \ldots, +26^n = 26\frac{26^n-1}{25} \approx 26^n$ mögliche Schlüssel. Für genügend große Schlüssellänge n wird es daher auch mithilfe von Computern nicht mehr möglich sein, alle Schlüsselwörter *durchzuprobieren*.

Wie sieht es mit einer statistischen Analyse aus? Ein bestimmter Klartextbuchstabe wird nun nicht stets auf den gleichen Geheimtextbuchstaben abgebildet. Zum Beispiel wird das erste E im obigen Beispiel auf K, das zweite E aber auf R abgebildet. Es sieht also auf den ersten Blick so aus, als ob tatsächlich ein brauchbarer Algorithmus gefunden wäre, der nicht so leicht geknackt werden kann. Und so

dauerte es auch in der Tat einige Zeit bis im Jahre 1863 der preußische Infanteriemajor und Kryptograph Friedrich Wilhelm Kasiski (1805–1881) eine Methode zur Entschlüsselung veröffentlichte (die Methode war zuvor auch schon dem englischen Mathematiker Charles Babbage (1791–1871) bekannt, der sie aber geheim hielt). Unsere Krypotanalyse basiert auf späteren Arbeiten von W. Friedman: Wiederum hilft die Häufigkeitsverteilung des Klartexts.

Überlegen wir zunächst, dass alles, was wir brauchen, die Schlüsselwortlänge k ist. Kennen wir diese, dann ist die Kryptoanalyse auf die einer gewöhnlichen Verschiebverschlüsselung reduziert, wobei k Geheimtexte separat analysiert werden müssen. Warum? Betrachten wir nochmals unser Beispiel mit Schlüsselwort `VIGENERE`. Es hat die Länge $k = 8$. Das heißt aber, dass das 1., das 9., das 17. usw. Klartextzeichen mit derselben Verschiebung verschlüsselt wird:

DIESERTE**X**TISTGEH**E**IM

Man braucht also nur den Geheimtext in diese, den $k = 8$ Verschiebungen entsprechenden, Teiltexte zu zerstückeln (Teiltext 1 wäre also in unserem Beispiel der 1., 9., und 17. Buchstabe des Geheimtexts, also `YSZ`, Teiltext 2 wäre der 2., 10., und 18. Buchstabe des Geheimtexts, also `QBQ`, etc.). Jeder dieser Teiltexte ist monoalphabetisch mit einer simplen Verschiebung verschlüsselt, und kann daher mit den im letzten Abschnitt besprochenen Verfahren entschlüsselt werden.

Wie finden wir aber nun die Länge des Schlüsselwortes? Das können wir wieder mit dem Koinzidenzindex bewerkstelligen. Dazu zerlegen wir den Text wie eben beschrieben, wobei wir alle möglichen Werte für k durchprobieren. Liegt der Koinzidenzindex aller Teile nahe am Wert eines deutschen Texts, so ist das ein Hinweis, dass die Teile monoalphabetisch verschlüsselt wurden. So können wir erkennen, ob wir das richtige k gefunden haben!

Im Prinzip kann dafür jedes Teilstück verwendet werden, bei kurzen Texten werden die Teilstücke aber schnell kürzer und es bietet sich der Mittelwert über alle Teilstücke an, um das Ergebnis zu verbessern.

Beispiel 1.6 (→CAS) Kryptoanalyse einer Vigenère-Verschlüsselung
Entschlüsseln Sie den mit Vigenère verschlüsselten Geheimtext

```
LWAGZRUHUWVLMVRRWTOLFVZIAOQYVZTLQNHMEQMKAEIIPTAVMZFSLVU
YYMFSYENDRKVIZGDIICOEXUXMAJVRJHLQKWXEZGYGUJLJLQPGWTOALP
HPVNBLLUDCHORBRRMTOIEYVPTRQVLLSKXUKERHXLLBVUKMMRQAYUTVE
YYMFSYENHZHYFMVUIISOIHYETFNFRSDQXLVAGCIEOECPBRLQZREKEEK
HPBJNRRJHMHYJWYFIEKSKXYWMVMDZLLVUWECFDRCVSSLZYVGDRWPECX
UHPVDVMLRCRYIZQTVTOSVLUWBEDFVUACVSIBMSIJUUEDBGPANGYKAJO
LFMAZRULMVUNEVMAIJVNUHYWPVMKULRSOLGPFBLRKEEZHVVHQKVYIEJ
```

Lösung zu 1.6 Als erstes bestimmen wir die Schlüssellänge. Dazu zerlegen wir den Text wie oben beschrieben in $k = 1, 2, 3, \ldots$ Teilstücke mit um k versetzten

Buchstaben und berechnen den mittleren Koinzidenzindex der Teilstücke. Lassen wir uns K für $k = 1, 2, \ldots, 35$ vom Computer berechnen und stellen das Ergebnis graphisch dar (siehe Abbildung 1.2), so sehen wir, dass die Werte für $k = 11, 22, 33$ deutlich näher bei dem für einen deutschen Text zu erwartenden Wert von 0.076 liegen als alle anderen Werte. Das ist ein starkes Indiz für eine Schlüssellänge von 11! Der Anstieg ergibt sich, da die einzelnen Texte immer kürzer werden (ab 16 Teiltexten gibt es z.B. schon weniger als 26 Buchstaben und immer mehr Häufigkeiten müssen Null sein, was den Wert nach oben treibt).

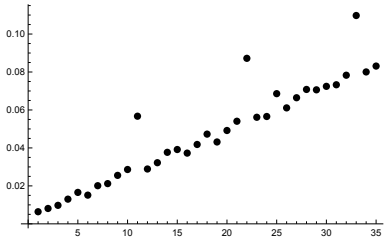

Abbildung 1.2: Für $k = 11, 22, 33$ wurden die Teilstücke vermutlich monoalphabetisch verschlüsselt.

Schauen wir nun, ob wir mit der Schlüssellänge 11 richtig liegen. Dazu müssen wir den Geheimtext wie oben beschrieben in 11 Teiltexte zerlegen: Text 1 besteht also aus dem 1., dem 12., dem 23., usw. Buchstaben des Geheimtext; Text 2 aus dem 2., dem 13., usw. Buchstaben des Geheimtextes, usw. Jeder dieser elf Texte ist (falls 11 die Schlüssellänge ist) mit derselben Verschiebechiffre verschlüsselt. Wir entschlüsseln also jeden Teiltext mit der Kryptoanalyse für die Cäsarverschiebung und erhalten folgende Werte: $20, 4, 8, 13, 25, 4, 17, 7, 0, 17, 3$ Damit haben wir einen Kandidaten für das Schlüsselwort: `UEINZERHARD`. Das zugehörige Schlüsselwort zum Entschlüsseln wäre: `GWSNBWJTAJX`. Versuchen wir, damit zu entschlüsseln:

```
RSSTANDAUFSRINESSCHLOSFESBRUESTUNTDERRITTERFVPSINVOLLER
EUESTUNGDAHBERTEERVONUATENKRACHUNQSPRACHZUSIPHICHSCHAUM
NLNACHUNDLEUNTESICHINVBLLERRUESTUAGWEITUEBERQIEERWHNTEB
EUESTUNGHIEEBEIVERLOREEALSOBALDZURRSTDENHELMHNDDANNDENH
NLTWONACHVEEFOLGENDSTUESEINZIELERCAUSENLOSBIFUNTENFIELU
ADHIERVERLOEERDURCHSEIASTREBENALSYETZTESNUNAHCHNOCHDASL
RBENANDEMERTANZBESONDEESHINGDERBLRCHSCHADENWNRNURGERING
```

Das ist aber leider noch kein ganz sinnvoller Text! Es scheint so, als ob der erste Buchstabe des Schlüsselwortes falsch ist. Damit der Klartext Sinn ergibt, sollte der erste Buchstabe im Klartext wohl ein `E` sein. Das würde dem Schlüsselwort `HEINZERHARD` entsprechen. Damit erhalten wir nun auch einen sinnvollen Klartext[1]. Diese kleine *Panne* ist nicht verwunderlich, denn der Geheimtext besteht nur aus

[1] Bei dem Text handelt es sich um das Gedicht „Ritter Fips“ von Heinz Erhardt. Mit freundlicher Genehmigung das Lappan Verlags, http://www.lappan.de

385 Buchstaben, und daher besteht jeder der 11 Teile nur aus 35 Buchstaben, was für eine zuverlässige statistische Auswertung eben nicht ausreicht. ■

Wir erwähnen, dass es noch weitere (bessere) Verfahren (z.B. den **Zeichenkoinzidenzindex**) gibt um die Schlüssellänge zu bestimmen, die aber auf ähnlichen Überlegungen beruhen. Uns geht es hier nur darum die wesentliche Idee zu veranschaulichen und wir gehen daher nicht weiter darauf ein.

Wird das Alphabet $\mathbb{Z}_{256}$ verwendet, so ist die **xor-Vigenère-Verschlüsselung** möglich:

$$y_i = x_i \oplus e_{i \bmod k}$$

wobei $e_i \in \mathbb{Z}_{256}$ die Zeichen des Schlüssels der Länge k sind. Beispiel: Schlüssel `KEY` der Länge 3: Der erste Buchstabe des Klartextes wird dann mit `K` xor-verknüpft, der zweite mit `E`, der dritte mit `Y`, der vierte wieder mit `K`, usw. Auch die xor-Vigenère-Verschlüsselung kann mit den oben beschriebenen statistischen Methoden geknackt werden. Allgemein funktioniert der Angriff unabhängig von der verwendeten monoalphabetischen Verschlüsselung.

Die ersten Verfahren zum Brechen der Vigenère-Chiffre wurden wie schon erwähnt in der Mitte des 19. Jahrhunderts publiziert und man könnte meinen, dass seitdem genug Zeit vergangen ist, damit sich die Schwächen der Vigenère-Chiffre auch bis zu großen Softwarekonzernen durchsprechen. Aber weit gefehlt, Vigenère-Chiffren (in verschiedenen Varianten) finden sich in einer großen Anzahl von Softwareprodukten. In Microsoft Word 2.0 wurde zum Beispiel ein 16 Byte Schlüsselwort verwendet, das analog wie bei der Vigenère-Chiffre mit jedem Byte im Klartext mittels xor verknüpft wird. Unsere Kryptoanalyse ist also problemlos anwendbar. Es kommt aber noch besser: es wurde auch der ganze Dateiheader mitverschlüsselt, und da in diesem an einer bestimmten Stelle immer 16 Nullbytes stehen, kann man dort das Schlüsselwort im Klartext ablesen (a xor $0 = a$)! Man hat also quasi die Tür versperrt und den Schlüssel stecken gelassen. In der Version Word 6.0 wurde deshalb die Dateistruktur komplizierter gemacht und ein Klartextbyte wird nur mit xor verknüpft, wenn es selbst oder das Ergebnis von Null verschieden ist. Da aber die Dokumentzusammenfassung sowohl verschlüsselt als auch unverschlüsselt in der Datei enthalten ist, ist es weiterhin möglich (auch ohne statistische Analyse) das Schlüsselwort zu finden (Known-Plaintext-Angriff). Dementsprechend gibt es im Internet auch eine Reihe von Programmen, die mit Word 6.0 verschlüsselte Dateien entschlüsseln.

Solche statistischen Verfahren sind natürlich nur eine von vielen Möglichkeiten der Kryptoanalyse.

1.3 One-Time Pad

Die Kryptoanalyse aus dem letzten Abschnitt hat gezeigt, dass auch eine Vigenère-Chiffre durch eine statistische Analyse entschlüsselt werden kann.

Stellen Sie sich nun vor, dass Sie den Geheimtext `QKEZFJRVHC` abgefangen haben. Sie vermuten, dass es sich um eine Vigenère-Verschlüsselung handelt, und probieren alle möglichen Schlüssel mit bis zu 10 Zeichen Länge durch und prüfen, ob sich ein sinnvoller Klartext ergibt. Dabei stoßen Sie also unter anderem auch

auf die Schlüssel `GWSNBWJTAJ` und `GWSNBCNBOY`. Welche Klartexte erhalten Sie für diese beiden Fälle?

Schlüsselwort	`GWSNBWJTAJ`
Geheimtext	`QKEZFJRVHC`
Klartext	`KOMMENICHT`

und

Schlüsselwort	`GWSNBCNBOY`
Geheimtext	`QKEZFJRVHC`
Klartext	`KOMMEHEUTE`

Welcher ist nun der richtige Klartext? Es kommt sogar noch schlimmer: Sie können hier zu *jeder* vorgegebenen Klartextnachricht (mit Länge 10) einen Schlüssel finden, sodass die Geheimtextnachricht `QKEZFJRVHC` zu genau dieser Klartextnachricht entschlüsselt wird.

Warum ist das möglich? — Der Grund liegt darin, dass hier das **Schlüsselwort gleich lang ist wie der zu verschlüsselnde Klartext**. Aus der vorgegebenen Geheimtextnachricht erhalten Sie durch Anwendung der 26^{10} möglichen Schlüssel alle 26^{10} möglichen Klartextnachrichten (= alle möglichen Aneinanderreihungen von 10 Buchstaben).

Eine solche Vigenère-Chiffre, bei der der Schlüssel gleich lang ist wie die Nachricht, wird **One-Time Pad** genannt. Das One-Time-Pad wurde 1917 vom amerikanischen AT&T-Ingenieur Gilbert S. Vernam (1890–1960) erfunden. Der Schlüssel darf nur einmal verwendet werden, denn wenn man zwei Klartexte mit demselben Schlüssel verschlüsselt, so ist das dasselbe, wie wenn man einen Klartext der doppelten Länge mit diesem Schlüssel verschlüsselt. Man spricht daher auch von einem **Wegwerfschlüssel**. Zu Beginn wurden die Schlüsselbuchstaben auf einen Abreißblock geschrieben. Sobald ein Buchstabe verwendet worden war, wurde das Blatt abgerissen.

Wird mit einem One-Time-Pad verschlüsselt, so enthält der Geheimtext (außer der Länge) keinerlei Informationen über den Klartext mehr. Ein solches Verschlüsselungsverfahren wird als **perfekt** oder als **uneingeschränkt sicher** bezeichnet.

In der Praxis sind heute sowohl der Klartext als auch der Schlüssel durch eine Folge von 0 und 1 gegeben, die bitweise mittels xor verknüpft werden. Es handelt sich also um die Vigenère-Verschlüsselung eines Klartextes $x_0x_1x_2\ldots$ mit einem Schlüssel $r_0r_1r_2\ldots$ über dem Alphabet $\{0,1\}$ (der für den Schlüssel verwendete Buchstabe r steht für „random“):

$$y_i = (x_i + r_i) \bmod 2 = x_i \oplus r_i.$$

Der Nachteil des One-Time-Pads ist, dass der Schlüssel immer genauso lang wie die Nachricht sein muss. (Wenn Sie also eine Festplatte verschlüsseln wollen, so

brauchen Sie eine zweite, gleichgroße um den Schlüssel zu speichern.) Das ist der Preis, den man für die absolute Sicherheit zahlen muss. Der Aufwand lohnt sich daher nur für sehr sicherheitskritische Anwendungen. Der heiße Draht zwischen Moskau und Washington soll mit einem One-Time-Pad verschlüsselt worden sein.

Damit die Sicherheit auch wirklich gegeben ist, muss man bei der Schlüsselerzeugung eines beachten: Denkt man sich einfach nur ein leicht merkbares System zur Schlüsselerzeugung aus, so ist die Wahrscheinlichkeit recht groß, dass dieses System irgendwann durchsickert, und die Sicherheit ist dahin. Außerdem können dann wiederum Regelmäßigkeiten des Schlüssels bei der Kryptoanalyse verwendet werden. Der einzige Ausweg ist, für den Schlüssel ein Folge von **Zufallszahlen** zu nehmen. Aber auch hier ist wiederum Vorsicht geboten, denn Zufallszahlen, die auf Computern generiert werden, sind nicht zufällig! In der Regel wird ein Anfangswert („seed“) gewählt, aus dem dann alle weiteren mittels einer Funktion berechnet werden. Man spricht daher von **Pseudozufallszahlen**. Gibt es für den Anfangswert nur wenige (z.B. 2^{16}) Möglichkeiten, so können leicht alle „Zufallsfolgen“ durchprobiert werden. Schlimmer noch, oft wird als Anfangswert die Uhrzeit genommen. Weiß man daher ungefähr, wann der Schlüssel erzeugt wurde, so lässt sich die Anzahl der Möglichkeiten weiter einschränken! Solche Pseudozufallszahlen sind also für kryptographische Anwendungen unbrauchbar. Zur Erzeugung echter Zufallszahlen müssen physikalische Prozesse verwendet werden, deren Verhalten nicht vorhersagbar ist (z.B. thermisches Rauschen eines Widerstandes).

Kapitel 2

Vom Lotfällen zur Bildkompression

Dieser Teil enthält leicht adaptierte Ausschnitte aus unserem Buch [**teschl1**; **teschl2**].

2.1 Skalarprodukt und orthogonale Projektion

Sie kennen Vektorräume meist als $\mathbb{R}^2$ oder $\mathbb{R}^3$ aus der Schule. In diesem Fall können wir uns die Vektoren als Pfeile bzw. Punkte (Ortsvektoren) in der Ebene oder im Raum grafisch veranschaulichen. In der Mathematik hat man es gerne etwas allgemeiner und betrachtet den $\mathbb{R}^n$ (mit $n \in \mathbb{N}$ einer beliebigen natürlichen Zahl) obwohl man sich Vektoren im Fall $n \geq 4$ nicht mehr anschaulich vorstellen kann (ich kann mir zumindest keine vierte Dimension vorstellen;-). Aber es kommt noch schlimmer, in der Mathematik bezeichnet man alles als Vektor solange man es nur addieren und skalieren kann wobei die „üblichen“ Rechenregeln erfüllt sind. Man bezeichnet also eine Menge von Objekten als Vektorraum, wenn für die Objekte eine Addition und eine Skalarmultiplikation definiert ist, die das Kommutativ-, Assoziativ- und Distributivgesetz erfüllen. Wir wollen hier nicht näher darauf eingehen sondern begnügen uns mit der Tatsache, dass man mit abstrakten Vektoren genauso rechnen kann wie mit den Vektoren aus dem $\mathbb{R}^2$ oder $\mathbb{R}^3$; denn genau das ist der mathematische Sinn der Definition eines Vektorraums! Somit können wir zum Beispiel auch die Menge der reell-wertigen Funktionen, die auf einem Intervall I definiert sind, als Vektoren auffassen, da man sie (punktweise) addieren und (punktweise) skalieren kann. Die notwendigen Rechengesetze erben sie dabei natürlich von $\mathbb{R}$.

Die Vektoren im $\mathbb{R}^2$ und $\mathbb{R}^3$ haben aber noch eine weitere wichtige Eigenschaft: Sie besitzen eine Länge und zwischen zwei Vektoren ist ein Winkel definiert. Insbesondere können zwei Vektoren aufeinander normal stehen. Für unser Beispiel der reell-wertigen Funktionen ist allerdings nicht klar was die *Länge* einer Funktion sein soll, geschweige denn, was der *Winkel* zwischen zwei Funktionen sein soll. Gelingt es aber, diese Begriffe auf Funktionen zu verallgemeinern, so ergeben sich mit einem Schlag zuvor ungeahnte Möglichkeiten!

Von Johann Wolfgang von Goethe stammt der Ausspruch „Die Mathematiker sind eine Art Franzosen: Redet man zu ihnen, so übersetzen sie es in ihre Sprache, und dann ist es alsbald etwas anderes.“. Und in der Tat ist der erste Schritt in der Mathematik oft die **Abstraktion**, also ein Problem zunächst auf seine grundlegendsten Bestandteile zu reduzieren und jeden Ballast zu entfernen, der bei der Lösung stören könnte. Dabei geht natürlich auch der unmittelbare Bezug zur Wirklichkeit verloren und viele Menschen verbinden daher mit Abstraktion etwas Negatives. Doch darum geht es nicht. Es geht darum, den Blick auf das *Wesentliche* frei zu machen und in einem Problem Strukturen und Analogien zu einem anderen Problem zu erkennen, dass in der Regel keinerlei Bezug zum Ausgangsproblem hat. Hat man z.B. erkannt, dass Funktionen Vektoren sind, dann kann ich mir Funktionen als Vektoren vorstellen und ein Problem über Funktionen mit meiner geometrischen Anschauung lösen. Es geht also nicht darum, Dinge als abstrakte Vektoren aufzufassen um Nicht-Mathematiker auszuschließen, sondern darum, dass mit einem Schlag das Wissen über Vektoren im $\mathbb{R}^n$ zugänglich gemacht wird. So kann man z.B. ein Bild im Computer als einen Vektor auffassen und dann einen Vektor in einem kleineren Unterraum suchen, der möglichst kurzen Abstand zum Originalvektor hat. Wenn man das geschickt macht wird der Betrachter den Unterschied nicht merken. Ist der Unterraum kleiner als der ursprüngliche Raum, so braucht man weniger Komponenten abzuspeichern und erreicht somit eine Datenkompression! Mit den Details wollen wir uns in diesem Abschnitt befassen.

Das **Skalarprodukt** oder auch **innere Produkt** $\langle \mathbf{a}, \mathbf{b} \rangle$ zweier Vektoren $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ und $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ im $\mathbb{R}^n$ ist definiert als

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{j=1}^{n} a_j b_j = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n.$$

Das Ergebnis dieser Multiplikation ist also kein Vektor, sondern ein Skalar, daher auch der Name. Den Begriff eines Skalarprodukts gibt es nicht nur im $\mathbb{R}^n$, sondern er kann auch für allgemeine Vektorräume definiert werden:

Definition 2.1 Ist V ein reeller Vektorraum, so nennt man eine Abbildung $\langle ., . \rangle : V \times V \to \mathbb{R}$ ein **Skalarprodukt**, falls sie für alle $\mathbf{a}, \mathbf{b} \in V$ und $k, h \in \mathbb{R}$ folgende Eigenschaften erfüllt:

$$\begin{aligned} \langle \mathbf{a}, \mathbf{a} \rangle &> 0, \quad \text{wenn } \mathbf{a} \neq \mathbf{0} \quad \text{(Positivität)} \\ \langle \mathbf{a}, \mathbf{b} \rangle &= \langle \mathbf{b}, \mathbf{a} \rangle \quad \text{(Symmetrie)} \\ \langle \mathbf{a}, k\mathbf{b} + h\mathbf{c} \rangle &= k\langle \mathbf{a}, \mathbf{b} \rangle + h\langle \mathbf{a}, \mathbf{c} \rangle \quad \text{(Linearität)} \end{aligned}$$

Diese Eigenschaften sind im Fall des $\mathbb{R}^n$ für das eingangs definierte Skalarprodukt leicht zu überprüfen. In der Definition 2.1 ist nur die Linearität im 2. Argument angegeben. Aus der Eigenschaft der Symmetrie folgt, dass das Skalarprodukt auch linear im ersten Argument ist, d.h.: $\langle k\mathbf{b} + h\mathbf{c}, \mathbf{a} \rangle = k\langle \mathbf{b}, \mathbf{a} \rangle + h\langle \mathbf{c}, \mathbf{a} \rangle$.

Die Länge eines Vektors $\mathbf{a} = (a_1, \ldots, a_n)$ kann mithilfe des Skalarprodukts ausgedrückt werden:

$$\|\mathbf{a}\|^2 = \langle \mathbf{a}, \mathbf{a} \rangle = |a_1|^2 + \ldots + |a_n|^2.$$

Für einen Einheitsvektor $\mathbf{e}$ gilt daher insbesondere immer $\langle \mathbf{e}, \mathbf{e} \rangle = 1$. Allgemein definiert man:

Definition 2.2 Ist V ein reeller Vektorraum mit einem Skalarprodukt $\langle ., . \rangle$, so ist die **Länge** oder **Norm** eines Vektors definiert durch

$$\|\mathbf{a}\|^2 = \langle \mathbf{a}, \mathbf{a} \rangle.$$

Was kann man sich unter dem Skalarprodukt vorstellen? Es sagt etwas über die Lage der beiden Vektoren relativ zueinander aus:

Satz 2.3 Für $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ gilt:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\varphi),$$

wobei $\varphi \in [0, \pi]$ der (kleinere) Winkel zwischen $\mathbf{a}$ und $\mathbf{b}$ in der von den beiden Vektoren aufgespannten Ebene ist.

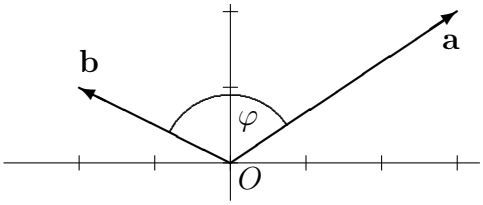

Abbildung 2.1: Winkel zwischen zwei Vektoren

Den Zusammenhang zwischen Winkel und Skalarprodukt im $\mathbb{R}^2$ kann man sich geometrisch überlegen. Betrachten wir dazu Abbildung 2.1. Ist α der Winkel zwischen $\mathbf{a}$ und der x-Achse, so kann $\mathbf{a}$ geschrieben werden als $\mathbf{a} = (\|\mathbf{a}\| \cos(\alpha), \|\mathbf{a}\| \sin(\alpha))$. Analog ist $\mathbf{b} = (\|\mathbf{b}\| \cos(\beta), \|\mathbf{b}\| \sin(\beta))$, wobei β wieder der Winkel zwischen $\mathbf{b}$ und der x-Achse ist. Der Winkel φ zwischen $\mathbf{a}$ und $\mathbf{b}$ ist in Abbildung 2.1 gleich $\varphi = \beta - \alpha$. Damit berechnen wir nun das Skalarprodukt $\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\alpha) \cos(\beta) + \|\mathbf{a}\| \|\mathbf{b}\| \sin(\alpha) \sin(\beta) = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\alpha - \beta)$, wobei im letzten Schritt das Additionstheorem $\cos(\alpha) \cos(\beta) + \sin(\alpha) \sin(\beta) = \cos(\alpha - \beta)$ für den Kosinus verwendet wurde.

Allgemein kann je nach Lage der Vektoren $\varphi = |\beta - \alpha|$ (falls $|\beta - \alpha| \leq \pi$) oder $\varphi = 2\pi - |\beta - \alpha|$ (falls $\pi \leq |\beta - \alpha| < 2\pi$) auftreten. Wegen $\cos(|\beta - \alpha|) = \cos(2\pi - |\beta - \alpha|) = \cos(\beta - \alpha)$ ist unser Ergebnis in allen Fällen richtig.

Beispiel 2.4 Winkel zwischen zwei Vektoren des $\mathbb{R}^2$
Berechnen Sie den Winkel zwischen
a) $\mathbf{a} = (3, 2)$ und $\mathbf{b} = (-2, 1)$ b) $\mathbf{a} = (1, 2)$ und $\mathbf{b} = (2, -1)$

Lösung zu 2.4

a) Es ist $\cos\varphi = \frac{\langle \mathbf{a},\mathbf{b}\rangle}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{3\cdot(-2)+2\cdot 1}{\sqrt{13}\sqrt{5}} = -\frac{4}{\sqrt{65}}$ und somit $\varphi = \arccos(-\frac{4}{\sqrt{65}}) = 2.09$ (in Radiant) $\approx 120°$. Die Vektoren sind in Abbildung 2.1 dargestellt.

b) Da $\langle \mathbf{a}, \mathbf{b}\rangle = 0$, folgt $\cos\varphi = 0$ und damit $\varphi = \frac{\pi}{2} = 90°$. ■

Im $\mathbb{R}^n$ schließen zwei Vektoren $\mathbf{a}$ und $\mathbf{b}$ genau dann einen rechten Winkel ein, wenn ihr Skalarprodukt $\langle \mathbf{a}, \mathbf{b}\rangle = 0$ ist (siehe auch letztes Beispiel). Allgemein definiert man:

Definition 2.5

- Zwei Vektoren $\mathbf{a}, \mathbf{b} \in V$ heißen **orthogonal**, wenn $\langle \mathbf{a}, \mathbf{b}\rangle = 0$ ist. Man schreibt dafür $\mathbf{a} \perp \mathbf{b}$.
- Zwei Vektoren $\mathbf{a}, \mathbf{b} \in V$ heißen **parallel**, wenn $\mathbf{a} = k\mathbf{b}$ oder $\mathbf{b} = k\mathbf{a}$ mit irgendeinem Skalar k.

Man sagt anstelle „orthogonal“ auch, dass $\mathbf{a}$ und $\mathbf{b}$ **normal** oder **senkrecht** aufeinander stehen.

Wegen der Symmetrie des Skalarprodukts gilt $\langle \mathbf{a}, \mathbf{b}\rangle = 0$ genau dann, wenn $\langle \mathbf{b}, \mathbf{a}\rangle = 0$.

Für zwei orthogonale Vektoren folgt nun der

Satz 2.6 (Pythagoras) Ist $\mathbf{a} \perp \mathbf{b}$, so folgt

$$\|\mathbf{a} + \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2.$$

Um diesen Satz in die vertraute Form zu bringen, zeichnen Sie zwei orthogonale Vektoren $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$. Wenn Sie auch $\mathbf{a} + \mathbf{b}$ einzeichnen, so ergibt sich ein rechtwinkliges Dreieck.

Der Satz von Pythagoras kann leicht nachgerechnet werden: $\|\mathbf{a} + \mathbf{b}\|^2 = \langle \mathbf{a} + \mathbf{b}, \mathbf{a} + \mathbf{b}\rangle = \langle \mathbf{a}, \mathbf{a}\rangle + \langle \mathbf{a}, \mathbf{b}\rangle + \langle \mathbf{b}, \mathbf{a}\rangle + \langle \mathbf{b}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2$. (Zunächst wurde dabei die Länge durch ein Skalarprodukt ausgedrückt (siehe Definition 2.2), dann wurde die Eigenschaft der Linearität (Definition 2.1) des Skalarprodukts verwendet, zuletzt wieder das Skalarprodukt als Länge ausgedrückt.)

Wir kommen nun zum wichtigen Begriff der *orthogonalen Projektion* eines Vektors in eine vorgegebene Richtung: Ein beliebiger Vektor $\mathbf{a}$ kann in Bezug auf eine Richtung, die durch einen Einheitsvektor $\mathbf{e}$ bestimmt ist, in zwei Anteile (Komponenten) zerlegt werden: $\mathbf{a} = \mathbf{a}_\parallel + \mathbf{a}_\perp$, wobei die Komponente $\mathbf{a}_\parallel = \langle \mathbf{e}, \mathbf{a}\rangle\mathbf{e}$ parallel zu $\mathbf{e}$ ist (d.h. ein Vielfaches von $\mathbf{e}$ ist) und die Komponente $\mathbf{a}_\perp = \mathbf{a} - \langle \mathbf{e}, \mathbf{a}\rangle\mathbf{e}$ orthogonal zu $\mathbf{e}$ ist.

Dass $\mathbf{a}_\perp$ orthogonal zu $\mathbf{e}$ (und damit orthogonal zu $\mathbf{a}_\parallel$) ist, dass also $\langle \mathbf{e}, \mathbf{a}_\perp\rangle = 0$ gilt, kann man folgendermaßen nachrechnen (wieder mithilfe von Definitionen 2.1 und 2.2): $\langle \mathbf{e}, \mathbf{a}_\perp\rangle = \langle \mathbf{e}, \mathbf{a}\rangle - \langle \mathbf{e}, \langle \mathbf{e}, \mathbf{a}\rangle\mathbf{e}\rangle = \langle \mathbf{e}, \mathbf{a}\rangle - \langle \mathbf{e}, \mathbf{a}\rangle\langle \mathbf{e}, \mathbf{e}\rangle = 0$ (hier haben wir $\langle \mathbf{e}, \mathbf{e}\rangle = 1$ verwendet).

Das ist in Abbildung 2.2 für $\mathbf{a} \in \mathbb{R}^2$ veranschaulicht. Zusammenfassend gilt:

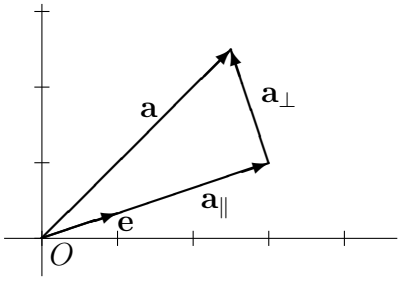

Abbildung 2.2: Orthogonale Projektion

Satz 2.7 Sei $\mathbf{e}$ ein Einheitsvektor. Jeder Vektor $\mathbf{a} \in V$ kann bezüglich $\mathbf{e}$ in zwei zueinander orthogonale Komponenten zerlegt werden:

$$\mathbf{a} = \mathbf{a}_{\parallel} + \mathbf{a}_{\perp},$$

wobei

$$\mathbf{a}_{\parallel} = \langle \mathbf{e}, \mathbf{a} \rangle \mathbf{e}$$

die **(orthogonale) Projektion** von $\mathbf{a}$ in Richtung von $\mathbf{e}$ und

$$\mathbf{a}_{\perp} = \mathbf{a} - \langle \mathbf{e}, \mathbf{a} \rangle \mathbf{e}$$

das **orthogonale Komplement** von $\mathbf{a}$ in Richtung von $\mathbf{e}$ genannt wird.

Beispiel 2.8 Orthogonale Projektion
Berechnen Sie die Komponenten von $\mathbf{a} = (1, 3)$ parallel und orthogonal zu $\mathbf{e} = \frac{1}{\sqrt{2}}(1, 1)$.

Lösung zu 2.8 Die Projektion $\mathbf{a}_{\parallel}$ in Richtung von $\mathbf{e}$ ist

$$\mathbf{a}_{\parallel} = \langle \mathbf{e}, \mathbf{a} \rangle \mathbf{e} = \langle \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix} \rangle \mathbf{e} = \frac{4}{\sqrt{2}} \mathbf{e} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

Daraus berechnen wir

$$\mathbf{a}_{\perp} = \mathbf{a} - \mathbf{a}_{\parallel} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

■

Mit dem Satz von Pythagoras folgt die Beziehung $\|\mathbf{a}\|^2 = \|\mathbf{a}_{\parallel}\|^2 + \|\mathbf{a}_{\perp}\|^2$ (siehe auch Abbildung 2.2) und deshalb insbesondere die Abschätzung

$$\|\mathbf{a}_{\parallel}\| \leq \|\mathbf{a}\|.$$

Das heißt, die Länge der orthogonalen Projektion $\mathbf{a}_\parallel$ ist kleiner oder gleich als die Länge von $\mathbf{a}$. Das ist deshalb interessant, weil die orthogonale Projektion $\mathbf{a}_\parallel$ in der Praxis oft als Näherung für $\mathbf{a}$ verwendet wird – mehr dazu in Kürze.

Daraus können wir eine wichtige Abschätzung herleiten: Wenn wir einen Vektor $\mathbf{b}$ mithilfe des Einheitsvektors $\mathbf{e}$ in seine Richtung als $\mathbf{b} = \|\mathbf{b}\|\mathbf{e}$ schreiben, dann erhalten wir (wieder mithilfe der Linearität aus Definition 2.1): $|\langle \mathbf{a}, \mathbf{b}\rangle| = |\langle \mathbf{a}, \|\mathbf{b}\|\mathbf{e}\rangle| = \|\mathbf{b}\||\langle \mathbf{a}, \mathbf{e}\rangle| = \|\mathbf{b}\|\|\mathbf{a}_\parallel\| \leq \|\mathbf{b}\|\|\mathbf{a}\|$, also:

Satz 2.9 (Cauchy-Schwarz-Ungleichung) Für beliebige Vektoren $\mathbf{a}, \mathbf{b} \in V$ gilt

$$|\langle \mathbf{a}, \mathbf{b}\rangle| \leq \|\mathbf{a}\|\|\mathbf{b}\|$$

(mit Gleichheit genau dann, wenn $\mathbf{a}$ und $\mathbf{b}$ parallel sind).

Damit folgt auch leicht die Dreiecksungleichung (siehe Definition 2.2): $\|\mathbf{a} + \mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|$, denn $\|\mathbf{a} + \mathbf{b}\|^2 = \langle \mathbf{a} + \mathbf{b}, \mathbf{a} + \mathbf{b}\rangle = \langle \mathbf{a}, \mathbf{a}\rangle + \langle \mathbf{a}, \mathbf{b}\rangle + \langle \mathbf{b}, \mathbf{a}\rangle + \langle \mathbf{b}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + 2|\langle \mathbf{a}, \mathbf{b}\rangle| + \|\mathbf{b}\|^2 \leq \|\mathbf{a}\|^2 + 2\|\mathbf{a}\|\|\mathbf{b}\| + \|\mathbf{b}|^2 = (\|\mathbf{a}\| + \|\mathbf{b}\|)^2$.

Diese Ungleichung wurde zuerst vom russischen Mathematiker Wiktor Jakowlewitsch Bunjakowski (1804–1889), einem Schüler des französischen Mathematikers Augustin Louis Cauchy (1789–1857), veröffentlicht. Fünfzig Jahre später wurde sie vom deutschen Mathematiker Hermann Amandus Schwarz (1843–1921) wiederentdeckt.

Die Bedeutung der orthogonalen Projektion begründet sich nun unter anderem in folgender Eigenschaft: Gegeben ist ein Vektor $\mathbf{a}$, der durch einen Vektor aus der linearen Hülle $\mathrm{LH}\{\mathbf{e}\} = \{k\mathbf{e} | k \in \mathbb{R}\}$ von $\mathbf{e}$ (also durch einen Vektor, der ein Vielfaches von $\mathbf{e}$ ist) angenähert werden soll. Unter allen diesen Vielfachen von $\mathbf{e}$ ist gerade die orthogonale Projektion $\mathbf{a}_\parallel$ die beste Approximation von $\mathbf{a}$. Genau meint man mit „der besten“ Approximation:

Satz 2.10 Sei $\mathbf{e}$ ein normierter Vektor, d.h. $\|\mathbf{e}\| = 1$. Für jeden Vektor $\mathbf{x} \in \mathrm{LH}\{\mathbf{e}\}$ ist $\|\mathbf{a} - \mathbf{x}\| \geq \|\mathbf{a}_\perp\|$. Gleichheit gilt genau dann, wenn $\mathbf{x} = \mathbf{a}_\parallel$.

In Worten bedeutet dieser Satz: Für jeden Vektor $\mathbf{x}$ aus der linearen Hülle von $\mathbf{e}$ ist der Abstand zwischen $\mathbf{a}$ und $\mathbf{x}$ (das ist der „Fehler“, wenn $\mathbf{a}$ durch $\mathbf{x}$ approximiert wird) größer oder gleich der Länge von $\mathbf{a}_\perp$. *Minimal* ist der Abstand für $\mathbf{x} = \mathbf{a}_\parallel$.

Geometrisch ist das im $\mathbb{R}^2$ nach einem Blick auf Abbildung 2.2 klar: Stellen Sie sich Vektoren $\mathbf{x}$ in Richtung von $\mathbf{e}$ vor, und den zugehörigen Abstand $\|\mathbf{a} - \mathbf{x}\|$. Wenn Sie $\mathbf{x} = \mathbf{a}_\parallel$ nehmen (wie in der Abbildung dargestellt), dann ist der Abstand gerade die Länge $\|\mathbf{a}_\perp\|$. In diesem Sinn ist in der „eindimensionalen Welt“ der Geraden, die durch $\mathbf{e}$ aufgespannt wird, der Vektor $\mathbf{a}_\parallel$ die beste Approximation von $\mathbf{a}$.

Ein allgemeiner Beweis von Satz 2.10: Ist $k\,\mathbf{e}$ ein beliebiger Vektor auf der durch $\mathbf{e}$ aufgespannten Geraden, so ist das Quadrat des Abstands zu $\mathbf{a} = \mathbf{a}_\parallel + \mathbf{a}_\perp$ gegeben durch

$$\|\mathbf{a} - k\,\mathbf{e}\|^2 = \|(\mathbf{a}_\parallel - k\,\mathbf{e}) + \mathbf{a}_\perp\|^2 = \|\mathbf{a}_\parallel - k\,\mathbf{e}\|^2 + \|\mathbf{a}_\perp\|^2.$$

Das folgt mit dem Satz von Pythagoras, weil $(\mathbf{a}_\parallel - k\,\mathbf{e}) \perp \mathbf{a}_\perp$. Der Abstand ist minimal, wenn $k\,\mathbf{e} = \mathbf{a}_\parallel$.

Mehr zur Approximation wird im Abschnitt 2.2 folgen. Nun zu einer anderen Anwendung der orthogonalen Projektion:

Mithilfe der Zerlegung eines Vektors $\mathbf{a}$ in die beiden Komponenten $\mathbf{a}_\parallel$ und $\mathbf{a}_\perp$ können wir auch den Abstand einer Geraden vom Ursprung bestimmen.

Betrachten wir die Abbildung 2.3. Gegeben ist der Ortsvektor $\mathbf{a}$ irgendeines

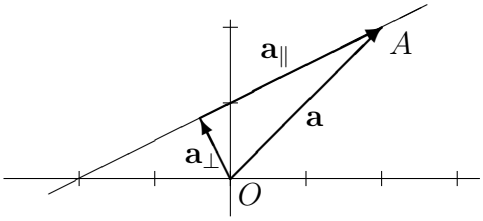

Abbildung 2.3: Abstand einer Geraden vom Ursprung

Punktes A auf der Geraden. Der Ortsvektor wird nun zerlegt in seine Projektion $\mathbf{a}_\parallel$ in Richtung $\mathbf{e}$ der Geraden, und in $\mathbf{a}_\perp$. Interessant ist, dass nicht nur für den konkreten gezeichneten Punkt A, sondern für den Ortsvektor $\mathbf{a}$ *jedes* beliebigen Punktes auf der Geraden die Komponente $\mathbf{a}_\perp$ gleich ist. Wegen Satz 2.10 ist die Länge $\|\mathbf{a}_\perp\|$ der (minimale) **Abstand der Geraden vom Ursprung**.

Beispiel 2.11 Abstand einer Geraden vom Ursprung
Gegeben ist die Gerade in Abbildung 2.3

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{a} + k\mathbf{e}, \qquad \text{mit} \quad \mathbf{a} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad \mathbf{e} = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$

wobei $\mathbf{a}$ der Ortsvektor des Punktes $A = (2, 2)$ auf der Geraden ist. Berechnen Sie den Abstand der Geraden vom Ursprung.

Lösung zu 2.11 Der gesuchte Abstand ist die Länge $\|\mathbf{a}_\perp\|$. Es ist $\mathbf{a}_\perp = \mathbf{a} - \mathbf{a}_\parallel$, wir müssen also zuerst die Projektion $\mathbf{a}_\parallel$ berechnen:

$$\mathbf{a}_\parallel = \langle \mathbf{e}, \mathbf{a} \rangle \mathbf{e} = \frac{1}{\sqrt{5}}(4+2)\,\mathbf{e} = \frac{6}{5} \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$

und damit ist

$$\mathbf{a}_\perp = \mathbf{a} - \mathbf{a}_\parallel = \frac{1}{5} \begin{pmatrix} 10 \\ 10 \end{pmatrix} - \frac{1}{5} \begin{pmatrix} 12 \\ 6 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} -2 \\ 4 \end{pmatrix}.$$

Die gesuchte Länge von $\mathbf{a}_\perp$ ist daher

$$\|\mathbf{a}_\perp\| = \sqrt{\frac{4+16}{25}} = \frac{2}{\sqrt{5}}.$$

■

2.1.1 Anwendung: Matched-Filter und Vektorraum-basierte Informationssuche

Die Idee der Bestapproximation in einem Vektorraum kann man z. B. auch bei der Dokumentsuche verwenden. Nehmen wir an, wir wollen eine Suchmaschine schreiben, die auf Webprogrammierung spezialisiert ist. Sie durchsucht Webseiten nur nach einigen wenigen vorgegebenen Stichworten, z. B.,

Einführung, Schnellkurs, Referenz, HTML, XML, PHP, Java,

und erstellt für jedes Dokument einen Vektor, dessen j-te Komponente angibt, ob und wo das Dokument das j-te Stichwort enthält. Zum Beispiel: 3. . . Stichwort kommt im Titel vor, 2. . . Stichwort ist im Dokument hervorgehoben (Fettdruck, Überschrift, etc.), 1. . . Stichwort kommt im Text vor, 0. . . Stichwort kommt nicht vor. Die Vektoren einiger Webseiten könnten dann wie folgt aussehen:

$$\begin{aligned} \mathbf{a}_1 &= (3,0,0,3,2,0,1) \\ \mathbf{a}_2 &= (0,0,3,1,0,3,2) \\ \mathbf{a}_3 &= (0,3,0,0,0,0,3) \\ &\vdots \end{aligned}$$

Sucht nun ein Benutzer nach den Stichworten „HTML Referenz“, so ordnen wir dieser Anfrage den Suchvektor

$$\mathbf{q} = (0,0,1,1,0,0,0)$$

zu und berechnen die Winkel zwischen den Dokumentvektoren und dem Suchvektor:

$$\cos(\varphi_j) = \frac{\langle \mathbf{a}_j, \mathbf{q} \rangle}{\|\mathbf{a}_j\| \, \|\mathbf{q}\|}, \qquad j = 1, 2, 3, \ldots .$$

Die Übereinstimmung ist umso besser, je näher der Winkel φ_j bei 0 liegt, also je größer $\cos(\varphi_j)$ ist (für $\varphi = 0$ wären die Vektoren ja parallel).

Das ist aber nur der Gipfel des Eisbergs. Die gleiche Idee kann man natürlich in einem beliebigen Vektorraum verwenden, um nach der besten Übereinstimmung zwischen einem Suchvektor $\mathbf{q}$ und gegebenen Vektoren $\mathbf{a}_j$ zu suchen. Da die Cauchy-Schwarz-Ungleichung in einem beliebigen Vektorraum mit Skalarprodukt gilt, und Gleichheit genau bei parallelen Vektoren eintritt, brauchen wir nur nach dem Maximum von

$$\frac{|\langle \mathbf{a}_j, \mathbf{q} \rangle|}{\|\mathbf{a}_j\| \, \|\mathbf{q}\|}$$

zu suchen. Zum Beispiel kann man auf dem Vektorraum der reellen Funktionen ein Skalarprodukt mithilfe des Integrals erklären und diese Idee verwenden, um in einem Audiosignal nach einem bestimmten Teilstück zu suchen. Oder wir können damit ein vorgegebenes Objekt in einem Bild suchen. Dieses Verfahren ist als **Matched-Filter** bekannt.

2.2 Orthogonalentwicklungen

Eine Orthogonalzerlegung $\mathbf{a} = \mathbf{a}_{\parallel} + \mathbf{a}_{\perp}$ kann man nicht nur bezüglich eines Vektors $\mathbf{e}$ (der eine Gerade aufspannt), sondern auch bezüglich mehrerer Vektoren $\mathbf{u}_1, \ldots, \mathbf{u}_m$ durchführen. (Diese spannen einen Teilraum auf; z. B. im Fall von zwei linear unabhängigen Vektoren $\mathbf{u}_1, \mathbf{u}_2$ eine Ebene.) Analog zum letzten Abschnitt fragt man wieder nach der besten Approximation von $\mathbf{a}$ in dem durch $\mathbf{u}_1, \ldots, \mathbf{u}_m$ aufgespannten Teilraum.

Die Überlegungen in diesem Abschnitt gelten für einen beliebigen Vektorraum V. Stellen Sie sich aber, damit es anschaulicher wird, zum Beispiel $V = \mathbb{R}^n$ bzw. noch konkreter $V = \mathbb{R}^3$ vor:

Definition 2.12 Gegeben sind beliebige Vektoren $\mathbf{a}_1, \ldots, \mathbf{a}_m$ aus V. Die Menge aller Linearkombinationen von $\mathbf{a}_1, \ldots, \mathbf{a}_m$ heißt die **lineare Hülle** dieser Vektoren. Schreibweise:

$$\mathrm{LH}\{\mathbf{a}_1, \ldots, \mathbf{a}_m\} = \{\sum_{j=1}^{m} k_j \mathbf{a}_j \mid k_j \in \mathbb{R}\} \subseteq V.$$

Die lineare Hülle von $\mathbf{a}_1, \ldots, \mathbf{a}_m$ besteht also aus allen Vektoren aus V, die sich in der Form

$$k_1\mathbf{a}_1 + k_2\mathbf{a}_2 + \ldots + k_m\mathbf{a}_m$$

schreiben lassen. Die lineare Hülle eines Vektors (ungleich dem Nullvektor) ist z.B. eine Gerade durch den Ursprung. Die lineare Hülle zweier Vektoren die nicht parallel sind ist eine Ebene durch den Ursprung.

Definition 2.13 Die Vektoren $\mathbf{u}_1, \ldots, \mathbf{u}_m \in V$ werden als **Orthonormalsystem** bezeichnet, falls sie die Länge 1 haben und paarweise orthogonal sind, wenn also

$$\langle \mathbf{u}_j, \mathbf{u}_k \rangle = \delta_{jk}, \quad \text{wobei} \quad \delta_{jk} = \begin{cases} 0, & \text{falls } j \neq k \\ 1, & \text{falls } j = k \end{cases}$$

gilt.

In einem n-dimensionalen Vektorraum V bildet jedes Orthonormalsystem aus n Vektoren eine Basis von V, die Vektoren spannen also ganz V auf. Man spricht in diesem Fall von einer **Orthonormalbasis**. Die Standardbasis $\mathbf{e}_1 = (1, 0, \ldots, 0)$, $\mathbf{e}_2 = (0, 1, 0, \ldots, 0)$, $\ldots$, $\mathbf{e}_n = (0, \ldots, 0, 1)$ ist zum Beispiel eine Orthonormalbasis.

Satz 2.14 (Orthogonalentwicklung) Ist $\mathbf{u}_1, \ldots, \mathbf{u}_n \in V$ eine Orthonormalbasis, so lässt sich jeder Vektor $\mathbf{a} \in V$ als

$$\mathbf{a} = \sum_{j=1}^{n} \langle \mathbf{u}_j, \mathbf{a} \rangle \mathbf{u}_j$$

schreiben.

Beispiel 2.15 Orthonormalbasis
Zeigen Sie, dass

$$\mathbf{u}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, \qquad \mathbf{u}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\end{pmatrix}$$

eine Orthonormalbasis des $\mathbb{R}^2$ bilden und berechnen Sie die Orthogonalentwicklung von $\mathbf{a} = (1, 3)$.

Lösung zu 2.15 Wir berechnen zunächst die folgenden Skalarprodukte: $\langle \mathbf{u}_1, \mathbf{u}_2\rangle = 0$, $\langle \mathbf{u}_1, \mathbf{u}_1\rangle = \langle \mathbf{u}_2, \mathbf{u}_2\rangle = 1$. Also bilden $\mathbf{u}_1, \mathbf{u}_2$ ein Orthonormalsystem, und da es 2 Vektoren sind, handelt es sich um eine Orthonormalbasis des $\mathbb{R}^2$. Die Entwicklungskoeffizienten von $\mathbf{a}$ bezüglich dieser Orthonormalbasis lauten

$$\langle \mathbf{u}_1, \mathbf{a}\rangle = \frac{1}{\sqrt{2}}(1+3) = 2\sqrt{2}, \qquad \langle \mathbf{u}_2, \mathbf{a}\rangle = \frac{1}{\sqrt{2}}(-1+3) = \sqrt{2},$$

und somit (siehe Abbildung 2.4)

$$\mathbf{a} = 2\sqrt{2}\,\mathbf{u}_1 + \sqrt{2}\,\mathbf{u}_2.$$

Probe: Setzen Sie $\mathbf{u}_1, \mathbf{u}_2$ ein und prüfen Sie nach, ob wir tatsächlich durch diese Linearkombination $\mathbf{a}$ erhalten! ■

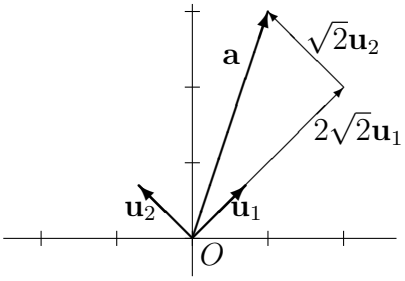

Abbildung 2.4: Orthogonalentwicklung: $\mathbf{a} = 2\sqrt{2}\mathbf{u}_1 + \sqrt{2}\mathbf{u}_2$

Wenn das Orthonormalsystem in einem n-dimensionalen Vektorraum V aus $m < n$ Vektoren besteht, so spannt es nicht ganz V, sondern nur einen Teilraum von V auf.

Zum Beispiel spannt ein Orthonormalsystem aus zwei Vektoren nicht den ganzen $\mathbb{R}^3$, sondern nur eine Ebene auf.

Wieder können wir in diesem Fall, analog wie im Abschnitt 2.1, einen Vektor $\mathbf{a} \in V$ in zwei Komponenten zerlegen: eine Komponente, die im Teilraum liegt, und eine, die orthogonal zum Teilraum ist.

Satz 2.16 Gegeben ist ein Orthonormalsystem $\mathbf{u}_1, \ldots, \mathbf{u}_m$. Jeder Vektor $\mathbf{a} \in V$ lässt sich in der Form $\mathbf{a} = \mathbf{a}_\parallel + \mathbf{a}_\perp$ schreiben, wobei

$$\mathbf{a}_\parallel = \sum_{j=1}^{m} \langle \mathbf{u}_j, \mathbf{a} \rangle \mathbf{u}_j$$

in $\mathrm{LH}\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ liegt, und

$$\mathbf{a}_\perp = \mathbf{a} - \mathbf{a}_\parallel$$

orthogonal zu jedem Vektor in $\mathrm{LH}\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ ist. Der Vektor $\mathbf{a}_\parallel$ heißt die **(orthogonale) Projektion** von $\mathbf{a}$ auf $\mathrm{LH}\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$.

Dass $\mathbf{a}_\perp$ orthogonal zu jedem der Vektoren $\mathbf{u}_1, \ldots, \mathbf{u}_m$ ist, können wir direkt nachrechnen:

$$\langle \mathbf{a}_\perp, \mathbf{u}_\ell \rangle = \langle \mathbf{a} - \mathbf{a}_\parallel, \mathbf{u}_\ell \rangle = \langle \mathbf{a}, \mathbf{u}_\ell \rangle - \sum_{j=1}^{m} \langle \mathbf{u}_j, \mathbf{a} \rangle \langle \mathbf{u}_j, \mathbf{u}_\ell \rangle = \langle \mathbf{a}, \mathbf{u}_\ell \rangle - \langle \mathbf{u}_\ell, \mathbf{a} \rangle = 0.$$

Damit ist $\mathbf{a}_\perp$ auch orthogonal zu jeder Linearkombination des Orthonormalsystems, insbesondere also auch zu $\mathbf{a}_\parallel$.

Anschaulich im $\mathbb{R}^3$ erklärt: Gegeben ist ein Vektor $\mathbf{a}$ und das Orthonormalsystem $\mathbf{u}_1, \mathbf{u}_2$, das die Ebene $U = \mathrm{LH}\{\mathbf{u}_1, \mathbf{u}_2\}$ aufspannt. Dann können wir den Vektor in $\mathbf{a} = \mathbf{a}_\parallel + \mathbf{a}_\perp$ zerlegen, wobei die Komponente $\mathbf{a}_\parallel$ in der Ebene liegt ($\mathbf{a}_\parallel$ ist eine Linearkombination von $\mathbf{u}_1, \mathbf{u}_2$) und $\mathbf{a}_\perp$ senkrecht auf die Ebene steht.

Beispiel 2.17 Orthogonale Projektion
Berechnen Sie die orthogonale Projektion $\mathbf{a}_\parallel$ von $\mathbf{a} = (3, -1, 4) \in \mathbb{R}^3$ auf die Ebene, die vom Orthonormalsystem

$$\mathbf{u}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{u}_2 = \frac{1}{\sqrt{6}} \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix}$$

aufgespannt wird, sowie $\mathbf{a}_\perp$.

Lösung zu 2.17 Mit obiger Formel erhalten wir

$$\mathbf{a}_\parallel = \langle \mathbf{u}_1, \mathbf{a} \rangle \mathbf{u}_1 + \langle \mathbf{u}_2, \mathbf{a} \rangle \mathbf{u}_2 = \frac{7}{\sqrt{2}} \mathbf{u}_1 + \frac{1}{\sqrt{6}} \mathbf{u}_2 = \frac{1}{3} \begin{pmatrix} 11 \\ -1 \\ 10 \end{pmatrix}.$$

Damit berechnen wir

$$\mathbf{a}_\perp = \mathbf{a} - \mathbf{a}_\parallel = \frac{2}{3} \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}.$$

Machen Sie die Probe, indem Sie überprüfen, ob dieser Vektor orthogonal zu $\mathbf{u}_1$ und $\mathbf{u}_2$ ist! ■

Wir wissen nun, wie die Projektion auf einen Teilraum definiert ist, wo aber braucht man das?

In der Praxis hat man es oft mit dem Problem zu tun, dass $\mathbf{a}$ die *ideale* Lösung wäre, aus technischen Gründen aber nur Vektoren in einem Teilraum U zulässig sind. In diesem Fall sucht man (analog wie in Abschnitt 2.1) jenen Vektor aus dem Teilraum, für den der Abstand zu $\mathbf{a}$ (also der Fehler) minimal ist.

Dazu wählt man ein Orthonormalsystem $\mathbf{u}_1, \dots, \mathbf{u}_m$, das U aufspannt. Die gesuchte beste Näherung ist dann die Projektion von $\mathbf{a}$ auf $U = \mathrm{LH}\{\mathbf{a}_1, \dots, \mathbf{a}_m\} = \mathrm{LH}\{\mathbf{u}_1, \dots, \mathbf{u}_m\}$:

Satz 2.18 Sei $\mathbf{u}_1, \dots, \mathbf{u}_m \in V$ ein Orthonormalsystem. Dann gilt für jeden Vektor $\mathbf{x} \in \mathrm{LH}\{\mathbf{u}_1, \dots, \mathbf{u}_m\}$, dass $\|\mathbf{a} - \mathbf{x}\| \geq \|\mathbf{a}_\perp\|$, mit Gleichheit genau für $\mathbf{x} = \mathbf{a}_\parallel$.

Im $\mathbb{R}^3$ veranschaulicht: Gesucht ist jener Vektor in einer von zwei linear unabhängigen Vektoren $\mathbf{u}_1$ und $\mathbf{u}_2$ aufgespannten Ebene, der möglichst nahe an einem gegebenen Vektor $\mathbf{a} \in \mathbb{R}^3$ liegt. Geometrisch ist uns klar, dass die beste Approximation genau jener Vektor aus der Ebene ist, für den die Differenz zu $\mathbf{a}$ (= der Fehler, der bei der Approximation gemacht wird) orthogonal auf die Ebene steht. Die beste Näherung ist also die Projektion $\mathbf{a}_\parallel$. Wir finden sie, indem wir die Projektion $\mathbf{a}_\parallel$ auf die Ebene berechnen.

Im Fall $n > 3$ können wir uns nicht mehr veranschaulichen, warum gerade unsere Projektion den minimalen Fehler liefert. Dazu ist ein analytisches Argument notwendig, das ohne eine geometrische Vorstellung auskommt: Ist $\mathbf{x} = \sum_{j=1}^{m} k_j \mathbf{u}_j$ irgendeine Linearkombination, so gilt nach dem Satz von Pythagoras $\|\mathbf{a} - \mathbf{x}\|^2 = \|\mathbf{a}_\parallel + \mathbf{a}_\perp - \mathbf{x}\|^2 = \|\mathbf{a}_\parallel - \mathbf{x}\|^2 + \|\mathbf{a}_\perp\|^2$. Der Abstand wird also genau dann minimal, wenn wir $\mathbf{x} = \mathbf{a}_\parallel$ wählen.

Unsere Überlegungen waren motiviert von unserer Vorstellung im $\mathbb{R}^3$. Alles, was wir verwendet haben, waren aber immer nur die drei Eigenschaften aus Definition 2.1 (Positivität, Symmetrie und Linearität) für das Skalarprodukt. Auf den ersten Blick scheint das nur unnötig abstrakt und kompliziert. Interessant wird das Ganze aber, wenn man beginnt, die mathematische Struktur des $\mathbb{R}^3$ mit seinem Skalarprodukt auch in anderen Objekten zu erkennen! In diesem Sinn ist zum Beispiel die Zerlegung des Tones einer schwingenden Saite in seine Grund- und Oberschwingungen nichts anderes als eine Orthogonalentwicklung. Hat man das erkannt, so lassen sich plötzlich komplizierte Probleme mithilfe geometrischer Anschauung lösen, die zuvor unlösbar erschienen sind.

2.2.1 Anwendung: Bildkompression mit der diskreten Kosinustransformation

Eine für die Praxis besonders wichtige Orthogonalbasis C ist durch

$$\mathbf{u}_k = (u_{1k}, \ldots, u_{nk}), \qquad k = 1, \ldots, n$$

mit

$$u_{jk} = c_k \cos\left(\frac{(2j-1)(k-1)\pi}{2n}\right), \quad \text{und} \quad c_k = \begin{cases} \sqrt{\frac{1}{n}}, & \text{für } k = 1, \\ \sqrt{\frac{2}{n}}, & \text{für } k \neq 1, \end{cases}$$

gegeben. Sie ist als **diskrete Kosinustransformation** (DCT) bekannt. Zu jedem Vektor $\mathbf{x}$ bestimmt man den Bildvektor

$$\mathbf{y} = (\langle \mathbf{u}_1, \mathbf{x} \rangle, \ldots, \langle \mathbf{u}_n, \mathbf{x} \rangle).$$

Er enthält gerade die Entwicklungskoeffizienten bezüglich der Orthonormalbasis. Aus dem Bildvektor $\mathbf{y}$ kann der Originalvektor jederzeit mit

$$\mathbf{x} = y_1\mathbf{u}_1 + \cdots + y_n\mathbf{u}_n$$

zurückerhalten werden. Bei praktischen Anwendungen ist $\mathbf{x} = (x_1, \ldots, x_n)$ zum Beispiel ein Vektor von Signalwerten. Bezeichnen wir den zugehörigen Bildvektor (Koeffizientenvektor) mit $\mathbf{y} = (y_1, \ldots, y_n)$. Mit anderen Worten, $\mathbf{x} = y_1\mathbf{u}_1 + \ldots + y_n\mathbf{u}_n$ ist die Orthogonalentwicklung von $\mathbf{x}$ bezüglich der Orthonormalbasis $\mathbf{u}_1, \ldots, \mathbf{u}_n$. Die Projektion von $\mathbf{x}$ auf den Teilraum, der durch die ersten $m < n$ Basisvektoren aufgespannt wird, gibt eine Approximation des Originalvektors, die für viele Fälle ausreichend ist: $\mathbf{x} \approx y_1\mathbf{u}_1 + \ldots + y_m\mathbf{u}_m$. Diese Approximation wird eindeutig durch die m Entwicklungskoeffizienten $y_1, \ldots, y_m$ charakterisiert.

Man ersetzt in diesem Sinn die n Komponenten des Originalvektors durch $m \leq n$ Entwicklungskoeffizienten, also die Projektion auf die ersten m Komponenten, und erreicht dadurch eine Datenreduktion. Dies ist die Grundidee des JPEG-Verfahrens. Dabei gehen Daten verloren – die JPEG-Kompression ist also *nicht* verlustfrei. Natürlich ist das Wegwerfen von Daten im Allgemeinen kein sehr gutes Kompressionsverfahren. Der Grund, warum es hier trotzdem funktioniert, liegt in der Art und Weise wie das menschliche Auge sieht: Für die Wahrnehmung sind kontinuierliche Änderungen wichtiger als rasche Sprünge. Eine Folge von abwechselnd weißen und schwarzen Bildpunkten wird also (wenn sie dicht nebeneinander liegen) als eine Folge von grauen Punkten wahrgenommen. In diesem Sinn könnte man in einem Graustufenbild jeweils 2x2 Blöcke von Bildpunkten zusammenfassen und durch einen Mittelwert ersetzten. Dadurch werden 4 Bildpunkte durch einen ersetzt und 75% der Originaldaten eingespart. Liegen die Bildpunkte eng nebeneinander, so wird das von unserem Auge kaum wahrgenommen.

In Abbildung 2.5 sind die Basisvektoren für $n = 4$ veranschaulicht und man sieht, dass der erste Koeffizient dem Mittelwert entspricht,

$$y_1 = \langle \mathbf{u}_1, \mathbf{x} \rangle = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} x_j,$$

während die weiteren Koeffizienten die Schwankungen mit immer höher werdender Freqenzen wiedergeben. Da bei einem digitalisierten Bild die Bildpunkte nicht

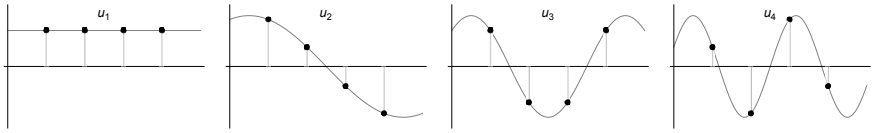

Abbildung 2.5: Veranschaulichung der Basisvektoren der DCT für $n = 4$.

linear sondern rechteckig angeordnet sind, ist es von Vorteil diese Struktur beizubehalten. In diesem Fall ist das Bild als eine Matrix $X = (x_{k_1 k_2})$ gegeben, wobei $x_{k_1 k_2}$ die Graustufe, also eine ganze Zahl zwischen 0 und 255, des Bildpunktes (Pixel) mit vertikalem Index k_1 und horizontalem Index k_2 ist. Die Vektoren in einer Orthonormalbasis werden nun auch durch zwei Indizes gekennzeichnet und man erhält die **zweidimensionale Kosinustransformation**, indem man kreuzweise Produkte aus den Vektoren der eindimensionalen Kosinustransformation bildet (in der Mathematik spricht man vom Tensorprodukt):

$$u_{(j_1 j_2),(k_1 k_2)} = u_{j_1 k_1} u_{j_2 k_2}.$$

Die zugehörigen Basisvektoren sind für $n_1 \times n_2 = 4 \times 4$ als Graustufenbilder in Abbildung 2.6 dargestellt. Ein vorgegebenes Bild wird als Linearkombination (also als Überlagerung) dieser Basisbilder dargestellt. Wieder entspricht der erste (beginnend in der linken oberen Ecke) Koeffizient dem Mittelwert des gesamten Blocks während die höheren Koeffizienten genauere Details enthalten. Je mehr Koeffizienten man weglässt, umso unschärfer wirkt das Bild. Das wird in Abbildung 2.7 veranschaulicht: Ausgegangen wurde von einem Graustufenbild, gegeben durch eine Matrix aus Graustufen. Für diese Matrix wurden mit der diskreten Kosinustransformation die zugehörigen Entwicklungskoeffizienten berechnet. Von den Entwicklungskoeffizienten wurde ein vorgegebener Prozentsatz weggeworfen (es wurde also nur ein vorgegebener Block beginnend links oben behalten) und daraus das Bild rekonstruiert. Mathematisch gesehen wurde also die orthogonale Projektion auf die vorgegebenen Basisbilder berechnet.

Bei einem Farbbild wird jedem Bildpunkt ein Farbwert zugeordnet, also ein Zahlentripel, das den Rot-, Grün-, und Blauanteil angibt. Jeder dieser drei Werte kann separat behandelt werden. Bei JPEG wandelt man dabei zuerst den RGB Wert in ein äquivalentes Tripel YCbCr um, das die Helligkeit Y (Graustufe) und die

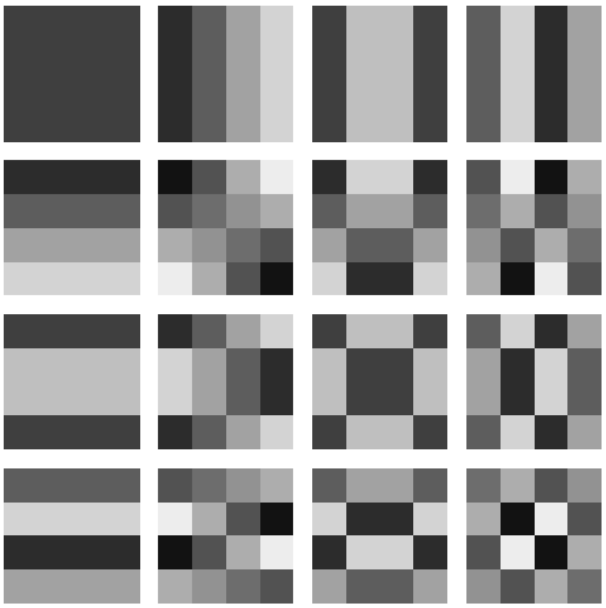

Abbildung 2.6: Die Basisvektoren der 2d DCT für $n = 4 \times 4$.

zwei Farbkomponenten Cb (Blue-Yellow Chrominance) und Cr (Red-Green Chrominance) beschreibt. Da das menschliche Auge viel mehr Helligkeitsrezeptoren als Farbrezeptoren besitzt, sind für die beiden Farbkomponenten weniger Entwicklungskoeffizienten notwendig. Außerdem transformiert man bei JPEG nicht das gesamte Bild sondern zerlegt es in 8×8 Blöcke (für die effektive numerische Berechnung muss n eine Potenz von 2 sein) und, statt die höheren Koeffizienten ganz wegzuwerfen, speichert man sie mit geringerer Genauigkeit (das wird als Quantisierung bezeichnet). Am Ende werden die Koeffizienten nochmals verlustfrei mittels Huffmancodierung komprimiert.

Die Quantisierung wird durch eine sogenannte Quantisierungsmatrix realisiert. Dabei wird jeder Koeffizient der DCT durch den entsprechenden Eintrag in der Quantisierungsmatrix dividiert und das Ergebnis auf die nächste ganze Zahl gerundet. Umso größer der Koeffizient in der Quantisierungsmatrix, umso stärker ist der Effekt der Rundung. Im Extremfall, wenn der Koeffizient klein bzw. der entsprechende Koeffizient in der Quantisierungsmatrix groß ist, so wird zu 0 gerundet.

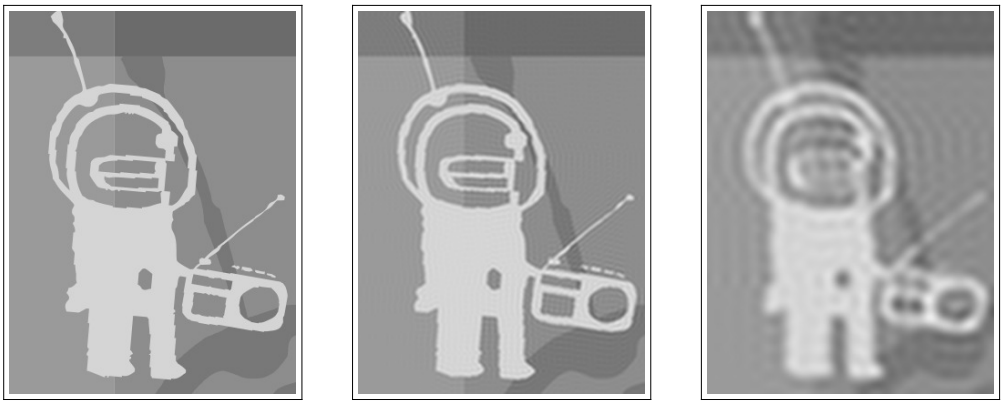

Abbildung 2.7: Effekt der Projektion auf die ersten Komponenten bei der 2d DCT.

Kapitel 3

AlkoMat(h). Modellierung in der Atemgasanalyse

Dieser letzte Abschnitt soll einen kleinen Einblick in die aktuelle Forschung geben und zeigen, wie Mathematik bei der Modellierung medizinischer Probleme verwendet wird. Dieser Teil ist eine leicht gekürzte Fassung von [**oemg2013**].

3.1 Wozu Modellierung?

Bereits zur Zeit des Hippocrates war die Atemanalyse Teil der medizinischen Diagnostik. Die moderne Ära der Atemgasanalyse wurde durch den Nobelpreisträger Linus Pauling eingeleitet, der aufzeigte, dass die menschliche Atemluft über 200 flüchtige organische Verbindungen (volatile organic compounds, VOCs) in picomolarer Konzentration enthält. Der Vorteil von Atemtests im Vergleich zu z.B. Bluttests liegt auf der Hand: Sie sind nicht-invasiv und können daher beliebig oft (sogar kontinuierlich in Echtzeit) durchgeführt werden. Dementsprechend ist die Atemgasanalyse ein wichtiges aktuelles Forschungsgebiet in der Medizin. Österreich ist hier mit dem Institut für Atemgasanalytik der Universität Innsbruck international an vorderster Front vertreten.

Der wohl bekannteste Atemtest ist der Alkomat, der zur Bestimmung der Blutalkoholkonzentration bei Verkehrskontrollen im Straßenverkehr verwendet wird. Die Idee dahinter ist, dass der Alkohol im Blut beim Gasaustausch in den Lungenbläschen (Alveolen) an die eingeatmete Luft abgegeben und danach ausgeatmet wird. Über die Messung der Alkoholkonzentration in der Ausatmungsluft können also Rückschlüsse über die Alkoholkonzentration im Blut gemacht werden. Dass das in der Praxis allerdings nicht ganz so einfach ist wie es klingt, kann man daraus ablesen, dass aus rechtlicher Sicht der Atemtest oft nur einen Anfangsverdacht begründet, der durch einen Bluttest bestätigt werden muss (die Gerichte in Europa sind sich da aber nicht immer ganz einig).

Während es bei einer Reihe von Atemtests ausreicht, nur den Nachweis der Existenz einer bestimmten Substanz zu erbringen, so ist in anderen Situationen, so wie zum Beispiel im eingangs erwähnten Alkoholtest, die Blutkonzentration die entscheidende Größe. Beim Alkoholtest geht man oft von der vereinfachten Annahme aus, dass das Verhältnis zwischen Blut- und Atemkonzentration durch 2100:1 gegeben ist. Dabei handelt es sich aber um einen Durchschnittswert, der

sowohl von der Person als auch vom momentanen physiologischen Zustand abhängt und dessen tatsächlicher Wert erheblich davon abweichen kann (in diversen Studien werden 50% und mehr angegeben).

Da Alkohol etwas schwieriger zu modellieren ist, wollen wir uns hier auf Isopren beschränken, ein Stoff, dem in der Medizin hohe Bedeutung beigemessen wird, der aber immer noch viele Fragen aufwirft. In [**Bajtarevic:2009**] wurde die Isoprenkonzentration in der Atemluft von Patienten mit Lungenkrebs untersucht und festgestellt, dass sie unter dem für gesunde PatientInnen üblichen Grenzwert von circa 80 ppb liegt. Abbildung 3.1 zeigt jedoch, dass die Isoprenkonzentration von gesunden PatientInnen durch Erhöhung der Atemfrequenz zwischen beiden Bereichen problemlos wechseln kann.

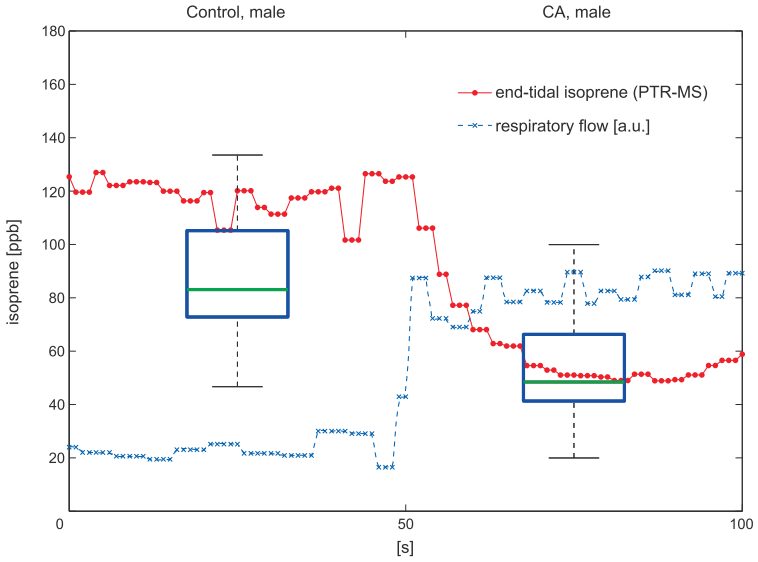

Abbildung 3.1: Zeitlicher Verlauf der Isoprenkonzentration (rote Punkte) in der Ausatemluft bei Hyperventilation eines gesunden Probanden überlagert mit dem Boxplot für gesunde (links) und lungenkranke (rechts) PatientInnen aus [**Bajtarevic:2009**]. Zusätzlich ist noch der zeitliche Verlauf des Atemfluss (blaue Kreuze) dargestellt.

Dieses Beispiel zeigt, dass es wichtig ist den genauen Zusammenhang zwischen Atmung und Atem- bzw. Blutkonzentrationen zu verstehen, um standardisierte Messmethoden zu entwickeln, die diese Probleme vermeiden.

3.2 Ein erster Einblick in die Modellierung

Um diesen Zusammenhang zwischen Atmung und Atem- bzw. Blutkonzentrationen zu modellieren, sehen wir uns zunächst den Atemvorgang aus biologischer Sicht

etwas genauer an. Dabei wird die Umgebungsluft über die Atemwege in die Lunge gesaugt, wo sie dann in den Lungenbläschen, den Alveolen, landet. Die Alveolen sind von einem dichten, Blut durchströmten Kapillarnetz umgeben, sodass es zu einem Gasaustausch durch Diffusion kommt. Einen ersten Einblick erhält man, indem man diesen Gasaustausch mit Hilfe einer Massenbilanzgleichung modelliert (vgl. Abbildung 3.2).

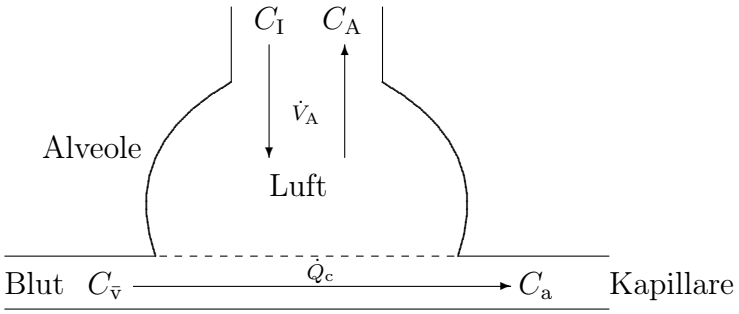

Abbildung 3.2: Schematischer Gasaustausch (symbolisiert durch die strichlierte Linie) in einer Alveole.

Die Isoprenmenge in der Lunge zur Zeit t ist gegeben durch das effektive Lungenvolumen V_A (ca. 4l) multipliziert mit der Konzentration in den Alveolen nach dem Gasaustausch $C_\mathrm{A}(t)$, die also der Konzentration der abgeatmeten Luft entspricht. Diese Menge ändert sich einerseits durch Zutransport aus dem Blut, gegeben durch

$$\dot{Q}_\mathrm{c}(t)\big(C_{\bar{\mathrm{v}}}(t) - C_\mathrm{a}(t)\big),$$

wobei $\dot{Q}_\mathrm{c}$ die Menge des Blutflusses durch die Lunge (Herzzeitvolumen; ca. 5l/min) ist und $C_{\bar{\mathrm{v}}}$ bzw. C_a die mittlere Konzentration im venösen bzw. arteriellen Blut bezeichnet, und andererseits durch Abatmen

$$\dot{V}_\mathrm{A}(t)\big(C_\mathrm{I} - C_\mathrm{A}(t)\big),$$

wobei $\dot{V}_\mathrm{A}$ den Luftfluss durch die Lunge (Alveolarventilation oder Atemzeitvolumen; ca. 6l/min) und C_I die Konzentration in der Umgebungsluft bezeichnet (wird als konstant angenommen).[1] Insgesamt erhalten wir also folgende Massenbilanzgleichung für die Isoprenmenge in der Lunge:

$$\underbrace{V_\mathrm{A}\frac{d}{dt}C_\mathrm{A}(t)}_{\text{zeitliche Änderung}} = \underbrace{\dot{V}_\mathrm{A}(t)\big(C_\mathrm{I} - C_\mathrm{A}(t)\big)}_{\text{Abtransport über die Atemluft}} + \underbrace{\dot{Q}_\mathrm{c}(t)\big(C_{\bar{\mathrm{v}}}(t) - C_\mathrm{a}(t)\big)}_{\text{Zutransport aus dem Blut}}. \tag{3.1}$$

[1] In der Wirklichkeit erfolgt die Atmung bekanntlich nicht durch einen kontinuierlichen Strom wie hier angenommen; die Größen hier sind daher als Mittelwert über einen Atemzug zu verstehen.

Als Nächstes bringen wir etwas Physik ins Spiel: Das Gesetz von Henry besagt, dass die Konzentration eines in einer Flüssigkeit gelösten Gases direkt proportional zum Partialdruck des entsprechenden Gases über der Flüssigkeit ist.[2] Aus der thermischen Zustandsgleichung idealer Gase folgt, dass der Partialdruck P_X aber proportional zur Konzentration C_X ist: $P_X = (R \cdot T)C_X$, wobei T die Temperatur (in Kelvin) und R die spezifische Gaskonstante ist. Gehen wir davon aus, dass die Temperatur in den Alveolen konstant ist, erhalten wir

$$C_\mathrm{a} = \lambda_\mathrm{b:air} C_\mathrm{A}, \tag{3.2}$$

wobei der Partitionskoeffizient $\lambda_\mathrm{b:air}$ für Isopren ca. 0.75 beträgt.

Befindet sich das System im Gleichgewicht,

$$0 = \dot{V}_\mathrm{A}\big(C_\mathrm{I} - C_\mathrm{A}\big) + \dot{Q}_\mathrm{c}\big(C_{\bar{\mathrm{v}}} - C_\mathrm{a}\big),$$

und ist in der Umgebungsluft kein Isopren vorhanden, $C_\mathrm{I} = 0$, so erhalten wir durch Auflösen die Farhi Gleichung (vgl. [**Farhi:1967a**])

$$C_\mathrm{A} = \frac{C_{\bar{\mathrm{v}}}}{\lambda_\mathrm{b:air} + \frac{\dot{V}_\mathrm{A}}{\dot{Q}_\mathrm{c}}}. \tag{3.3}$$

Das Ventilation-Perfusion-Verhältnis $\frac{\dot{V}_\mathrm{A}}{\dot{Q}_\mathrm{c}}$ liegt im Ruhezustand ungefähr bei 1 und kann somit für schwer lösliche Stoffe mit kleinem Partitionskoeffizient wie Isopren nicht vernachlässigt werden.

Aber auch für leicht wasserlösliche Stoffe wie Aceton (mit $\lambda_\mathrm{b:air} \approx 340$) oder Ethanol (Alkohol) (mit $\lambda_\mathrm{b:air} \approx 1750$) ist die Farhi Gleichung keine ausreichende Beschreibung. Außerdem werden diese Stoffe nicht nur in den Alveolen, sondern auch in den oberen Atemwegen über die Schleimhaut ausgetauscht, sodass eine detailreichere Modellierung notwendig ist (vgl. [**King2010a**]).

3.3 Kompartimentmodelle

Für die weitere Modellierung wird der Körper in einzelne Kompartimente zerlegt, zwischen denen wie im letzten Abschnitt Massenbilanzgleichungen aufgestellt werden. Für die Modellierung von Isopren wurde zum Beispiel in [**King:isoprene**] das in Abbildung 3.3 skizzierte Modell vorgeschlagen.

Die Bilanzgleichungen für das Lungenkompartiment kennen wir ja schon,

$$V_\mathrm{A}\frac{dC_\mathrm{A}}{dt} = \dot{V}_\mathrm{A}(C_\mathrm{I} - C_\mathrm{A}) + \dot{Q}_\mathrm{c}(C_{\bar{\mathrm{v}}} - C_\mathrm{a}), \tag{3.4}$$

[2] Hier wird angenommen, dass die Diffusion in den Alveolen ausreichend schnell erfolgt, so dass sich ein Gleichgewicht einstellt.

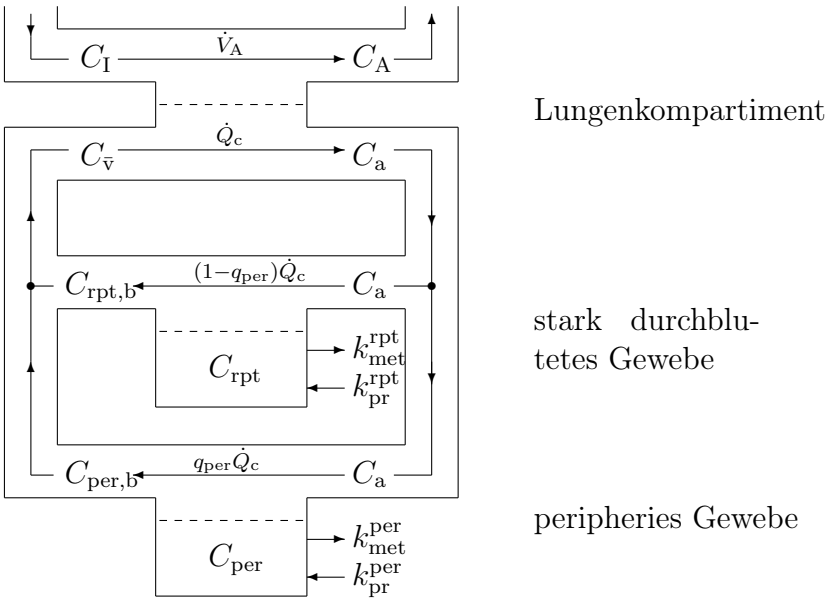

Abbildung 3.3: Drei-Kompartimentmodell für Isopren: Lungenkompartiment (Gasaustausch), stark durchblutetes Gewebe (Metabolismus und Produktion) und peripheres Gewebe (Speicherung, Metabolismus und Produktion).

und für die beiden anderen Kompartiments kann man analog vorgehen: Das arterielle Blut mit der Isoprenkonzentration C_{a} trennt sich in zwei Teile, $q_{\mathrm{per}}\dot{Q}_{\mathrm{c}}$ und $(1-q_{\mathrm{per}})\dot{Q}_{\mathrm{c}}$, die die beiden Gewebekompartiments durchströmen. Der erste Teil $(1-q_{\mathrm{per}})\dot{Q}_{\mathrm{c}}$ tritt in das stark durchblutete Gewebe (innere Organe, u.a. Leber) mit der Konzentration C_{a} ein und verlässt es mit einer Konzentration $C_{\mathrm{rpt,b}} = \lambda_{\mathrm{b:rpt}} C_{\mathrm{rpt}}$, die proportional der Isoprenkonzentration C_{rpt} im Gewebe ist. Der Proportionalitätsfaktor $\lambda_{\mathrm{b:rpt}}$ entspricht wie zuvor einem Partitionskoeffizient. Die über das Blut abtransportierte Menge ist somit $(1-q_{\mathrm{per}})\dot{Q}_{\mathrm{c}}(C_{\mathrm{a}} - \lambda_{\mathrm{b:rpt}} C_{\mathrm{rpt}})$. Zusätzlich gehen wir von einer konstanten Produktionsrate $k^{\mathrm{rpt}}_{\mathrm{pr}}$ und einem metabolischen Abbau $-k^{\mathrm{rpt}}_{\mathrm{met}} C_{\mathrm{rpt}}$ aus, wobei letzterer proportional zur vorhanden Konzentration im Kompartiment angesetzt wird. In Summe erhalten wir also folgende Massenbilanzgleichung für die Isoprenmenge im ersten Gewebekompartiment

$$V_{\mathrm{rpt}} \frac{dC_{\mathrm{rpt}}}{dt} = (1-q_{\mathrm{per}})\dot{Q}_{\mathrm{c}}(C_{\mathrm{a}} - \lambda_{\mathrm{b:rpt}} C_{\mathrm{rpt}}) + k^{\mathrm{rpt}}_{\mathrm{pr}} - k^{\mathrm{rpt}}_{\mathrm{met}} C_{\mathrm{rpt}}, \tag{3.5}$$

wobei V_{rpt} das zugehörige effektive Volumen des Kompartiments ist. Vollkommen analog erhält man für das zweite Gewebekompartiment (Muskeln)

$$V_{\mathrm{per}} \frac{dC_{\mathrm{per}}}{dt} = q_{\mathrm{per}}\dot{Q}_{\mathrm{c}}(C_{\mathrm{a}} - \lambda_{\mathrm{b:per}} C_{\mathrm{per}}) + k^{\mathrm{per}}_{\mathrm{pr}} - k^{\mathrm{per}}_{\mathrm{met}} C_{\mathrm{per}}. \tag{3.6}$$

Die Konzentration im arteriellen Blut ist, wie vorher, durch das Gesetz von Henry (3.2) gegeben und die Konzentration im venösen Blut ergibt sich aus den gemisch-

ten Anteilen der beiden Gewebekompartiments zu

$$C_{\bar{\mathrm{v}}} := (1 - q_{\mathrm{per}})\lambda_{\mathrm{b:rpt}} C_{\mathrm{rpt}} + q_{\mathrm{per}}\lambda_{\mathrm{b:per}} C_{\mathrm{per}}. \tag{3.7}$$

Zuletzt verwendet man noch dass der relative Anteil $q_{\mathrm{per}} \in (0, 1)$ des Blutfluss durch das Muskelkompartiment bei körperlicher Belastung von einem Ruhewert $q_{\mathrm{per}}^{\mathrm{rest}} \approx 0.08$ auf einen Maximalwert $q_{\mathrm{per}}^{\mathrm{max}} \approx 0.7$ steigt und durch

$$q_{\mathrm{per}}(\dot{Q}_{\mathrm{c}}) := q_{\mathrm{per}}^{\mathrm{max}} - (q_{\mathrm{per}}^{\mathrm{max}} - q_{\mathrm{per}}^{\mathrm{rest}}) \exp\big(-\tau \max\{0, \frac{\dot{Q}_{\mathrm{c}} - \dot{Q}_{\mathrm{c}}^{\mathrm{rest}}}{\dot{Q}_{\mathrm{c}}^{\mathrm{rest}}}\}\big), \quad \tau > 0, \tag{3.8}$$

modelliert werden kann, wobei $\dot{Q}_{\mathrm{c}}^{\mathrm{rest}}$ der Ruheblutfluss ist.

Setzt man all diese Informationen in die drei Massenbilanzgleichungen ein, so erhält man ein gekoppeltes System von drei **Differentialgleichungen** (also einen Zusammenhang zwischen Ableitungen und Funktionswerten dieser drei unbekannten Funktionen,

$$\mathbf{x}(t) = \begin{pmatrix} C_{\mathrm{a}}(t) \\ C_{\mathrm{rpt}}(t) \\ C_{\mathrm{per}}(t) \end{pmatrix}, \tag{3.9}$$

die zu jedem Zeitpunkt erfüllt sein muss)

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{g}(t, \mathbf{x}(t)). \tag{3.10}$$

Die Funktionen $\dot{V}_{\mathrm{A}}(t)$ und $\dot{Q}_{\mathrm{c}}(t)$ und die Konstanten $\lambda_{\mathrm{b:air}}$, C_{I}, $\dot{Q}_{\mathrm{c}}^{\mathrm{rest}}$ können gemessen und somit als bekannt vorausgesetzt werden. Nicht direkt gemessen und damit a priori unbekannt sind die Parameter $\lambda_{\mathrm{b:rpt}}$, $\lambda_{\mathrm{b:per}}$, q_{per}, $q_{\mathrm{per}}^{\mathrm{rest}}$, $q_{\mathrm{per}}^{\mathrm{max}}$, V_{A}, V_{rpt}, V_{per}, $k_{\mathrm{pr}}^{\mathrm{rpt}}$, $k_{\mathrm{met}}^{\mathrm{rpt}}$, $k_{\mathrm{pr}}^{\mathrm{per}}$, $k_{\mathrm{met}}^{\mathrm{per}}$. Außerdem kann die Konzentration in der Ausatemluft gemessen werden, die wir ja mit der Konzentration in den Alveolen identifiziert haben:

$$y(t) := C_{\mathrm{meas}}(t) = C_{\mathrm{A}}(t) = \lambda_{\mathrm{b:air}}^{-1} C_{\mathrm{a}}(t). \tag{3.11}$$

Man wird also versuchen die unbekannten Parameter zu bestimmen indem man die Modellgleichungen löst und dann durch Optimieren der Parameterwerte den *Fehler* zwischen Modell und Messung minimiert. Man spricht in diesem Zusammenhang auch von Parameteridentifikation. Am Ende erwartet man ein Modell, das die Messwerte entsprechend genau wiedergibt und bei dem die Parameterwerte innerhalb bestimmter physiologisch sinnvoller Grenzen liegen; man versucht also das Modell zu validieren.

Nun könnte man einwenden, dass dieses Modell immer noch eine viel zu grobe Abbildung der Wirklichkeit ist und man in Anbetracht der Rechenkapazitäten moderner Computer doch leicht noch weitere Kompartiments hinzunehmen könnte. Das Problem ist aber, dass dieses (und damit auch jedes andere Modell) eine Reihe von Größen und Parametern enthält, die nicht (zerstörungsfrei am Patienten) gemessen werden können. Natürlich erhält man mit einem komplexeren Modell mehr

Freiheitsgrade für die Optimierung und somit zwangsläufig eine bessere Übereinstimmung mit dem Experiment. Bei falschen Modellannahmen entfernt man sich aber zu weit von der physiologischen Wirklichkeit und etwaige Schlussfolgerungen aus dem Modell werden wertlos.

Deshalb gilt für das Modell (frei nach Einstein): Es muss so einfach wie möglich sein, aber nicht einfacher.

3.4 Die Experimente

Im Atemgaslabor des Instituts für Atemgasanalytik der Universität Innsbruck können die Konzentrationen mit modernsten Massenspektrometern in Echtzeit gemessen und mit dem Modell verglichen werden. Dabei sitzt der Proband auf einem Ergometer und zusätzlich zu den Atemkonzentrationen werden verschiedene medizinische Parameter gemessen (Abbildung 3.4).

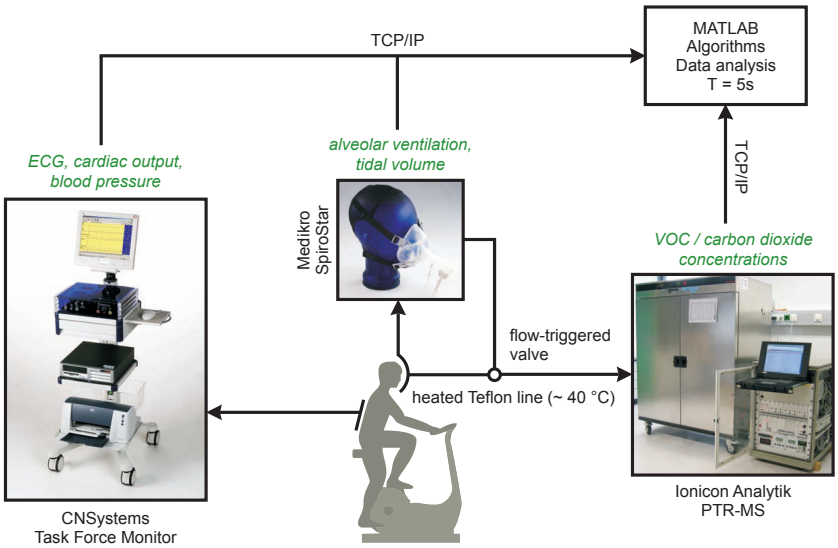

Abbildung 3.4: Schematischer Aufbau eines Ergometerexperiments

Die fehlenden Parameter können dann identifiziert werden und das Modell kann mit der Messung verglichen werden. Abbildung 3.5 zeigt das Ergebnis für die Atemgaskonzentration von Isopren, einem Stoff, der aktuell als potentieller Indikator für verschiedene metabolische Vorgänge im Körper diskutiert wird. Trotz dieses großen Interesses ist der Ursprung und die Funktion von Isopren im menschlichen Körper immer noch nicht ausreichend geklärt. Zur Zeit ist nur die Möglichkeit der Produktion von Isopren in der Leber bekannt, da der zugehörige Mechanismus aber nur langsam abläuft, geht man davon aus, dass es noch weitere Möglichkeiten geben muss. Im Experiment war der Proband zunächst fünf Minuten in Ruhe und Abbildung 3.5 zeigt einen konstanten Isoprenlevel. Danach musste er für ca. 15 Minuten

Rad fahren, gefolgt von 12 Minuten Pause, 15 Minuten Rad fahren, 3 Minuten Pause, 5 Minuten Rad fahren. Das Experiment zeigt, dass es beim Übergang von

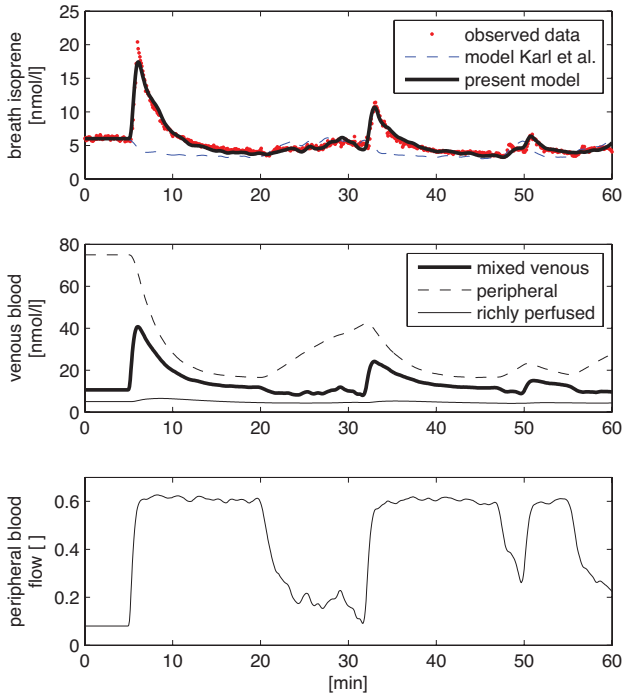

Abbildung 3.5: Messergebnisse für Isoprenkonzentrationen bei einem Ergometerexperiment und Vergleich mit dem älteren Modell aus [**Karl:2001**].

Ruhe auf Aktivitätsphasen zu einer stark erhöhten Isoprenkonzentration kommt, die nicht alleine durch die Änderung des Ventilation-Perfusion-Verhältnis erklärt werden kann (vgl. das strichliert eingezeichnete Modell aus [**Karl:2001**], welches die gemessenen Konzentrationen nicht befriedigend beschreiben kann). Weiters ersieht man, dass bei weiteren Belastungen der Anstieg wesentlich kleiner ausfällt, was durch einen Auswaschungseffekt erklärt werden kann. Die Modellierung liefert einen Hinweis, dass Isopren im Muskelkompartiment erzeugt wird — eine Vermutung, die inzwischen durch weitere Experimente mit Muskeldystrophiepatienten, die eine stark erniedrigte Isoprenkonzentration aufweisen, erhärtet wurde; siehe [**King2012**].

Ein weiteres vielversprechendes Anwendungsgebiet der Atemgasanalyse ist die Anästhesie, wo es wünschenswert ist, die Blutkonzentration des Anästhetikums im Gehirn während einer Operation zu überwachen; siehe [**ieee2011**].

Literatur

[Bajtarevic et al., 2009] Bajtarevic, A., Ager, C., Pienz, M., Klieber, M., Schwarz, K., Ligor, M., Ligor, T., Filipiak, W., Denz, H., Fiegl, M., Hilbe, W., Weiss, W., Lukas, P., Jamnig, H., Hackl, M., Haidenberger, A., Buszewski, B., Miekisch, W., Schubert, J., und Amann, A. (2009). Noninvasive detection of lung cancer by analysis of exhaled breath. *BMC Cancer*, 9:348. doi: 10.1186/1471-2407-9-348

[Farhi, 1967] Farhi, L. E. (1967). Elimination of inert gas by the lung. *Respir Physiol*, 3:1–11. doi: 10.1016/0034-5687(67)90018-7

[Hlastala, 1998] Hlastala, M. P. (1998). The alcohol breath test – a review. *Journal of Applied Physiology*, 84:401–408. http://jap.physiology.org/content/84/2/401

[Hoppensteadt et al., 2002] Hoppensteadt, F. C., und Peskin, C. S. (2002). Modeling and Simulation in Medicine and the Life Sciences. 2te Aufl., Springer Verlag. doi: 10.1007/978-0-387-21571-6

[Karl et al., 2001] Karl, T., Prazeller, P., Mayr, D., Jordan, A., Rieder, J., Fall, R., and Lindinger, W. (2001). Human breath isoprene and its relation to blood cholesterol levels: new measurements and modeling. *J Appl Physiol*, 91:762–70. http://jap.physiology.org/content/91/2/762

[King et al., 2010] King, J., Koc, H., Unterkofler, K., Mochalski, P., Kupferthaler, A., Teschl, G., Teschl, S., Hinterhuber, H., und Amann, A. (2010). Physiological modeling of isoprene dynamics in exhaled breath. *J Theor Biol*, 267:626–37. doi: 10.1016/j.jtbi.2010.09.028

[King et al., 2012] King, J., Mochalski, P., Unterkofler, K., Teschl, G., Klieber, M., Stein, M., Amann, A., und Baumann, M. (2012). Breath isoprene: muscle dystrophy patients support the concept of a pool of isoprene in the periphery of the human body. *Biochem Biophys Res Commun*, 423:526–530. doi: 10.1016/j.bbrc.2012.05.159

[King et al., 2011a] King, J., Unterkofler, K., Teschl, G., Teschl, S., Koc, H., Hinterhuber, H., und Amann, A. (2011a). A mathematical model for breath

gas analysis of volatile organic compounds with special emphasis on acetone. *J Math Biol*, 63:959–999. doi: 10.1007/s00285-010-0398-9

[King et al., 2011b] King, J., Unterkofler, K., Teschl, S., Amann, A., und Teschl, G. (2011b). Breath gas analysis for estimating physiological processes using anesthetic monitoring as a prototypic example. *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, pages 1001–1004. doi: 10.1109/IEMBS.2011.6090232

[King et al., 2013] King, J., Unterkofler, K., Amann, A., Teschl, S., und Teschl, G. (2013). Mathematische Modellierung in der Atemgasanalyse. *Schriftenreihe zur Didaktik der Mathematik der ÖMG* 46, 100–107. http://www.oemg.ac.at/DK/Didaktikhefte/2013%20Band%2046/VortragTeschl.pdf

[Modak, 2010] Modak, A. (2010). Single time point diagnostic breath tests: A review. *J Breath Res*, 4:017002. doi: 10.1088/1752-7155/4/1/017002

[Teschl, 2013] Teschl, S., und Teschl, G. (2013). Mathematik für Informatiker. Band 1: Diskrete Mathematik und Lineare Algebra. 4te Aufl., Springer Verlag. doi: 10.1007/978-3-642-54274-9

[Teschl, 2014] Teschl, S., und Teschl, G. (2014). Mathematik für Informatiker. Band 2: Analysis und Statistik. 3te Aufl., Springer Verlag. doi: 10.1007/978-3-642-37972-7

[Wobst, 2001] Wobst, R. (2001). Abenteuer Kryptologie. Methoden, Risiken und Nutzen der Datenverschlüsselung. 3te Aufl., Addison-Wesley.