Beschreibung klassischer Codes als Goppa-Codes

von Tobias Jacob

betreut von Prof. Dr. R.-P. Holzapfel

im Wintersemester 2003/2004

Vorwort

Im täglichen Leben gewinnt die Kommunikation zunehmend an Bedeutung. Immer größere Datenmengen in zumeist digitalisierter Form sollen immer schneller von einem Ort an den anderen gelangen. Eines der wichtigsten Qualitätsmerkmale stellt die Fehlerfreiheit der erhaltenen Information dar, denn bei der Übertragung von Daten, sei es über Kabel oder mittels einer Funkverbindung, treten häufig Störungen auf. Durch diese wird die gesendete Nachricht verändert und ohne geeignete Gegenmaßnahmen für den Empfänger unbrauchbar.
Die vorliegende Arbeit will nun einige dieser Maßnahmen vorstellen und erläutern, wie eine Nachricht schon vor dem Absenden bearbeitet werden kann, damit auftretende Fehler nach dem Empfang korrigiert werden können. Im ersten Kapitel werden zunächst Grundbegriffe bereitgestellt, die notwendig sind, um einige ausgewählte Klassen von elementaren Codes zu verstehen. Dazu gehören neben Klassikern, wie dem Hamming-Code, auch zyklische Codes, wie sie beispielsweise bereits in den 1960er Jahren von Reed und Solomon entwickelt wurden. Vor allem wollen wir hier die allgemeineren BCH-Codes untersuchen, die den Übergang zum zweiten Kapitel bilden. Dort werden dann rationale Codes behandelt, die nach ihrem Entdecker, dem russischen Mathematiker Goppa, benannt sind. Diese greifen auf tieferliegende Ideen zurück, so dass ein zweites mal einige Grundbegriffe zum Verständnis vorangestellt werden müssen. Die Goppa-Codes werden im folgenden soweit entwickelt, dass am Ende erkennbar ist, wie der Residuensatz eine Kontrolle der rationalen Goppa-Codes ermöglicht. Im dritten und letzten Kapitel wollen wir schließlich untersuchen, in wie weit sich klassische Codes als Goppa-Codes darstellen lassen. Dort wollen wir auch eventuelle Vor- und Nachteile der neueren Goppa-Codes gegenüber ihren klassischen Vorgängern diskutieren. Die Beispiele sind größtenteils aus tatsächlichen technischen Anwendungen gewählt, wie z. B. Satellitenkommunikation, Mobilfunk oder Compact Discs. Dadurch soll die Bedeutung des Themas für den Alltag unterstrichen werden.

Inhaltsverzeichnis

Kapitel 1

Elementare Codes

1.1 Grundbegriffe I

Der Vollständigkeit halber sollen zu Beginn einige wichtige Begriffe definiert werden. Sie sind fundamental für die Codierungstheorie und sollen deshalb hier nicht fehlen.

Definition 1.1.1. Es seien p eine Primzahl, $m \in \mathbb{N}$ und $q := p^m$. Dann ist $\mathbb{F} := \mathbb{F}_q = \mathbb{F}_{p^m}$ ein endlicher Körper mit Charakteristik p und $\#\mathbb{F} = q$ Elementen.

1. Die Elemente der Menge $\mathbb{F}$ sind die *Zeichen* oder *Buchstaben* des *Alphabets*.
2. Die Vektoren (n-Tupel) des Vektorraums $\mathbb{F}^n$ mit $n \in \mathbb{N}$ bilden die *Worte* der Länge n.
3. Ein *linearer Code* C ist ein bzgl. $\mathbb{F}$ linearer k-dimensionaler Unterraum von $\mathbb{F}^n$. Seine Elemente werden *Codewörter* genannt.

Wir betrachten in dieser Arbeit nur lineare Codes. Mit dieser Definition haben wir die Möglichkeit uns die einzelnen Codeworte als Punkte im Raum vorzustellen. Weiterhin ist es von Interesse auf dem Raum $\mathbb{F}^n$ eine Metrik, die so genannte Hamming-Metrik, zu definieren, um den Abstand der einzelnen Punkte und damit den Abstand der Worte kontrollieren zu können. Dazu erklären wir zunächst das *Hamming-Gewicht* eines Wortes $a := (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}^n$ in der folgenden Weise:

$$w(a) := \#\{i \in \{0, 1, \ldots, n-1\}; a_i \neq 0\}. \tag{1.1}$$

Setzen wir nun noch $d(a, b) := w(a - b) = \#\{i; a_i - b_i \neq 0\}$ für zwei beliebige Worte $a, b \in \mathbb{F}^n$, so haben wir eine Abbildung, die die drei Eigenschaften

einer Metrik erfüllt.
Für alle $a, b, c \in \mathbb{F}^n$ gilt:

1. $d(a,b) = 0 \Leftrightarrow a = b$
2. $d(a,b) = d(b,a)$ Symmetrie
3. $d(a,b) + d(b,c) \geq d(a,c)$ Dreiecksungleichung

Von besonderer Bedeutung für die Korrigierbarkeit von Codeworten eines Codes C ist der kleinste Abstand zweier Codeworte:

$$d := d_C := min\{d(a,b); a \neq b, a, b \in C\} \tag{1.2}$$

Dieser wird die *Minimaldistanz* des Codes C genannt.

Definition 1.1.2. Ist $C \subseteq \mathbb{F}^n$ ein Code mit $\dim_{\mathbb{F}} C = k$ und Minimaldistanz $d = d_C$, so heißt das Tripel (n, k, d) der *Typ* des Codes C.

Alle Codewörter haben - wie alle anderen Wörter im Sinne von Definition 1.1.1 - die Länge n, denn jedes Codewort ist ein Wort, aber nicht jedes Wort ist Codewort. Wir werden im nächsten Abschnitt sehen, wie sich die Dimension des Codes k als Länge der enthaltenen Information interpretieren lässt. Zuvor kommen wir jedoch noch einmal auf die Bedeutung des Minimalabstandes zurück. Ist nämlich C ein Code vom (n, k, d) -Typ, so ist C für jede natürliche Zahl $t \leq \frac{d-1}{2}$ ein *t-fehlerkorrigierender* Code. Der Code heißt *t-fehlererkennend*, wenn $t \leq \frac{d}{2}$. Der Wert $e := \left[\frac{d-1}{2}\right]$ wird auch als *Fehlerkorrekturindex* bezeichnet. Wir sehen, dass der Parameter d von zentraler Bedeutung für die Güte eines Codes ist. Je größer der Minimalabstand eines Codes ist, umso mehr Fehler können erkannt und korrigiert werden. Wir werden jedoch später sehen, dass d durch die anderen Parameter des Codes beschränkt ist.

1.2 Klassische Codes

Dieser Abschnitt soll dazu dienen, am konkreten Beispiel zweier sehr elementarer Codes die Funktionsweise von Codierung, Decodierung und die Möglichkeit einer Fehlerkorrektur zu veranschaulichen. Es sei bereits an dieser Stelle bemerkt, dass für spätere anspruchsvollere Codes auch die Algorithmen für die Verarbeitung, insbesondere die Fehlerkorrekturalgorithmen entsprechend aufwendiger werden.
Bevor wir zur praktischen Umsetzung übergehen können, benötigen wir noch folgende

Definition 1.2.1. Ist C ein (n, k, d)-Code wie oben, dann besitzt C eine k-elementige Basis über dem Körper $\mathbb{F}$. Eine $k \times n$ Matrix G mit Einträgen aus $\mathbb{F}$ heißt *Generatormatrix* von C, wenn ihre Zeilen eine Basis von C bilden.

Die Generatormatrix erzeugt aus einer vorgegebenen Information der Länge k ein Codewort der Länge n. Ist etwa $a = (a_0, a_1, \ldots, a_{k-1})$ vorgegeben und G eine $k \times n$ Matrix, dann gewinnt man ein Codewort $c = (c_0, c_1, \ldots, c_{n-1})$ der Länge n, indem man G auf a wirken lässt:

$$\mathbb{F}^k \to \mathbb{F}^n,\ a \mapsto a \cdot G =: c.$$

Es wird also zusätzliche Information (*Redundanz*) in a eingebaut. Das Codewort c wird gesendet und eventuell beschädigt, d.h. in einer oder mehreren Komponenten verändert. Dies lässt sich als vektorielle Addition $v = c + e$ interpretieren, da $\mathbb{F}^n$ ein linearer Raum ist. Auf der Empfängerseite muss entschieden werden, ob ein empfangenes Wort v ein Codewort ist und damit als unbeschädigt gewertet wird ($e = 0 \Rightarrow v = c$) oder ob es kein Codewort ist. Im zweiten Fall soll dann versucht werden den Fehlervektor e zu bestimmen und mit $c = v - e$ das ursprüngliche Codewort wiederherzustellen. Prinzipiell ist es denkbar, dass ein gesendetes Codewort c durch einen Fehlervektor so verändert wird, dass sich wieder ein zum Code gehöriges Wort $c+e = v = \tilde{c} \in C$ ergibt. Dieses wird vom Empfänger nicht erkannt. In diesem Fall sprechen wir von einem *Decodierfehler*.
Wir brauchen eine Abbildung, die prüft, ob ein Wort zum Code gehört. Dazu hilft uns eine weitere

Definition 1.2.2. Mit dem Standard-Skalarprodukt[1] zweier Vektoren $\langle \cdot, \cdot \rangle$: $\mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F},\ (a, b) \mapsto a \cdot b^T = \sum_{i=0}^{n-1} a_i b_i$ wird definiert:

1. a ist *orthogonal* zu b $(a \perp b) :\Leftrightarrow \langle a, b \rangle = 0$,
2. $C^\perp := \{u \in \mathbb{F}^n;\ \langle u, c \rangle = 0\ \forall c \in C\}$ heißt der zu C *duale Code*
3. und ist $C^\perp = C$, so heißt C *selbstdual*.

Ist C ein (n, k)-Code, so ist $C^\perp$ vom Typ $(n, n-k)$, da $\dim C^\perp = n - \dim C = n - k$. Wenn H eine $(n - k) \times n$ -Generatormatrix von $C^\perp$ mit den $n - k$ Zeilenvektoren $u_0, u_1, \ldots, u_{n-k-1} \in \mathbb{F}^n$ ist, dann gilt für jedes $c \in C = (C^\perp)^\perp$ nach Definition $c \perp C^\perp \Leftrightarrow c \perp u_i\ \forall 0 \leq i \leq n - k - 1$ und damit auch

$$H \cdot c^T = 0. \tag{1.3}$$

[1]Beim Rechnen in $\mathbb{F}_{p^m}$ ist die Definitheitseigenschaft nicht sinnvoll, z.B. ist $\langle (1, 1), (1, 1) \rangle = 2 = 0$ in $\mathbb{F}_2$, also $\langle \cdot, \cdot \rangle$ indefinit.

Man sagt, die Matrix H löscht oder annuliert c, wenn c Codewort ist. Ist $H \cdot v^T =: s_H(v) \neq 0$, so heißt $s_H(v)$ das *H-Syndrom* des Wortes v.

Definition 1.2.3. Es sei C ein (n,k)-Code. Ist H eine Generatormatrix des dualen Codes $C^\perp$, dann heißt H *Kontrollmatrix* oder *Prüfmatrix* von C.

Jede Generatormatrix ist gleichzeitig auch Kontrollmatrix genau dann, wenn C selbstdual ist. Denn dann ist eine beliebige Generatormatrix G von C auch Generatormatrix des dualen Codes, da nach Definition $C = C^\perp$ gilt.
Besonders anschaulich lässt sich die Funktionsweise des eben erläuterten am Beispiel der nach R.W. Hamming benannten Klasse von Codes beschreiben (siehe [Ham87], [Hol99]).
Zielstellung: Es sei $\mathbb{F} = \mathbb{F}_2 = \{0,1\}$ der Grundkörper über dem der Code definiert wird, also C ein *binärer Code*. Der Code soll 1-fehlerkorrigierend sein. Geben wir uns $r \in \mathbb{N}$ vor, so sind damit gerade $2^r - 1$ von Null verschiedene natürliche Zahlen (mögliche Fehlerpositionen) binär darstellbar. Folglich setzen wir $n = 2^r - 1$ als Länge des Codes. Dann ist die Kontrollmatrix H eine $r \times (2^r - 1)$-Matrix mit Einträgen aus $\mathbb{F}_2$.

Definition 1.2.4. Es seien $\mathbb{F} = \mathbb{F}_2 = \{0,1\}$, $r \in \mathbb{N}$ und $n = 2^r - 1$. Ein linearer $(n, n-r)$-Code heißt *binärer Hamming-Code* $Ham(r)$, falls die n Spalten der Kontrollmatrix H in natürlicher Reihenfolge aus den Ziffern-Vektoren der binären Darstellungen der Zahlen 1 bis $2^r - 1$ bestehen und

$$Ham(r) := \{c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}^n;\ H \cdot c^T = 0\}$$

ist.

Wir wollen nun das soeben erläuterte an einem einfachen Beispiel verdeutlichen.

Beispiel 1.2.1. Es sei $r = 3$ und damit $n = 2^3 - 1 = 7$. Als eine Kontrollmatrix finden wir

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Für ein empfangenes Wort $v = c + e$, c ein Codewort und e ein Fehlervektor berechnen wir mit der Kontrollmatrix

$$H \cdot v^T = H \cdot (c+e)^T = H \cdot c^T + H \cdot e^T = H \cdot e^T = H \cdot e_i^T,$$

wobei e_i den i-ten Einheitsvektor bezeichnet.
Die Fehlerposition i kann hier einfach durch das Skalarprodukt (in $\mathbb{Z}^3$)

$$(2^0, 2^1, 2^2) \cdot s_H(v) = i$$

bestimmt werden.
Da wir in $\mathbb{F}_2$ rechnen, ist mit dem Auffinden der Fehlerstelle auch die Korrektur ein leichtes: wir addieren an der entsprechenden Stelle i eine eins.
Wie sieht die Generatormatrix unseres $Ham(3)$-Codes aus? Aus der Kontrollmatrix lesen wir ab, dass für jedes Codewort $c = (c_1, c_2, \ldots, c_7) \in Ham(3)$ gelten muss:

$$\begin{array}{rcl} c_1 + c_3 + c_5 + c_7 & = & 0 \\ c_2 + c_3 + c_6 + c_7 & = & 0 \\ c_4 + c_5 + c_6 + c_7 & = & 0 \end{array} . \tag{1.4}$$

Wie man sofort sieht, sind die drei Gleichungen linear unabhängig. Sie erzeugen als Lösungsraum des Gleichungssystems einen Unterraum von $\mathbb{F}^n = \mathbb{F}^7$ der Dimension $k = 4$. Die freien Unbekannten des Gleichungssystem sind c_3, c_5, c_6, c_7. Ferner gilt (in $\mathbb{F}^2$) äquivalent zu (1.4)

$$\begin{array}{rcl} c_1 & = & c_3 + c_5 + c_7 \\ c_2 & = & c_3 + c_6 + c_7 \\ c_4 & = & c_5 + c_6 + c_7 \end{array} . \tag{1.5}$$

Ist also $a = (a_1, a_2, a_3, a_4) \in \mathbb{F}_2^4$ zu codieren, so bildet eine Matrix G dieses Wort auf ein Codewort von $Ham(3)$ ab, wenn sie die folgende Gestalt hat:

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

Die Matrix G identifiziert (a_1, a_2, a_3, a_4) mit (c_3, c_5, c_6, c_7). G ist dann eine Generatormatrix von $Ham(3)$.
Wir haben also insgesamt folgenden Codier-Decodier-Ablauf:

$$\begin{array}{ccccccc} \mathbb{F}^k & \xrightarrow{G} & C \subseteq \mathbb{F}^n = \mathbb{F}^{2^r-1} & \xrightarrow{+e} & \mathbb{F}^n & \xrightarrow{H} & \mathbb{F}^r \\ a = (a_1, \ldots, a_k) & \mapsto & c = (c_1, \ldots, c_n) & \mapsto & v = c + e & \mapsto & s = (s_0, \ldots, s_{r-1}) \end{array}$$

Sei nun beispielsweise $a = (1, 0, 1, 1)$ gegeben, dann ist das zugehörige Codewort $a \cdot G = c = (0, 1, 1, 0, 0, 1, 1)$. Wird das Wort c unbeschädigt übermittelt, so ermittelt der Decodierer für $H \cdot c^T = s(c) = (0, 0, 0)$. Es wird die Redundanz entfernt und man erhält die korrekte Information $(1, 0, 1, 1) = a$. Tritt bei der Übermittlung jedoch eine Störung auf und es wird das Wort

$v = (0,1,1,0,0,0,1)$ ergibt die Paritätsprüfung $H \cdot v^T = s(v) = (0,1,1)$ also einen Fehler an der sechsten Stelle $e = (0,0,0,0,0,1,0)$. Daraufhin wird an der sechsten Stelle eins addiert (in $\mathbb{F}_2^7$) und wir erhalten das korrekte Codewort $c = v - e = (0,1,1,0,0,1,1)$ und daraus die korrekt Information wie eben.
Unser Beispiel-Code $Ham(3)$ ist vom Typ $(7,4,3)$, wobei sich die Minimaldistanz $d = 3$ durch einfaches Nachrechnen ergibt.

Der folgende Satz soll uns eine Abschätzung für den Minimalabstand bei bekannter Codelänge und Dimension geben.

Satz 1.2.1 (Singleton-Schranke). *Es sei C ein beliebiger Code vom Typ (n,k,d) über einem Körper $\mathbb{F}$. Dann gilt für den Minimalabstand*

$$d \leq n - k + 1. \tag{1.6}$$

Der Wert $n-k+1$ wird Singleton-Schranke *genannt. Gilt sogar $d = n-k+1$, dann heißt C* MDS-Code *(maximum distance separable) oder auch* optimaler Code.

Beweis. Ist $C \in \mathbb{F}^n$ ein beliebiger linearer (n,k,d)-Code, dann ist die Einschränkung einer Abbildung $\varphi : \mathbb{F}^n \to \mathbb{F}^{n-d+1}$, $(c_0,\ldots,c_{n-1}) \mapsto (c_0,\ldots,c_{n-d})$ auf $C \subseteq \mathbb{F}^n$ injektiv, da C den Minimalabstand d hat und es somit keine zwei Codewörter aus C geben kann, die in den ersten $n-d+1$ Komponenten übereinstimmen. Denn sonst wäre der Minimalabstand kleiner als d. Aus der Injektivität von $\varphi|_C$ folgt

$$\begin{aligned} k = \dim_{\mathbb{F}}(C) = \dim_{\mathbb{F}}(\varphi(C)) &\leq n - d + 1 \\ \Rightarrow d &\leq n - k + 1 \end{aligned}$$

□

Von praktischem Interesse ist das Verhältnis $\frac{k}{n}$ von Information k und Gesamtlänge n in einem Codewort, die so genannte *Informationsrate*. In dem Beispiel beträgt sie $\frac{4}{7}$. Für größer werdendes n geht die Informationsrate $\frac{n-r}{n}$ gegen 1. Wollen wir etwa einen Code der Länge $2^{10} = 1024$ konstruieren, sind $r = 10$ Stellen davon zur Paritätskontrolle notwendig und die Informationsrate ist mit $\frac{1014}{1024}$ bereits deutlich höher. Jedoch kann hier nur ein Fehler in 1024 Bits korrigiert werden, während bei $Ham(3)$ alle 8 Bits ein Fehler korrigierbar ist.
Definition 1.2.4 lässt die Reihenfolge der Anordnung der $2^r - 1$ Spaltenvektoren offen.

Definition 1.2.5. Zwei Codes $C, C' \subseteq \mathbb{F}^n$ heißen *äquivalente Codes*, wenn $\#C = \#C'$ und es eine Permutation $\pi \in \mathfrak{S}_n$ der Indexmenge $\{0,1,\ldots,n-1\}$

gibt, so dass sich jedes Codewort $c' \in C'$ als eine Permutation eines $c = (c_0, \ldots, c_{n-1}) \in C$ schreiben lässt: $c' = (c'_0, \ldots, c'_{n-1}) = (\pi(c_0), \ldots, \pi(c_{n-1}))$. Schreibweise: $C \sim_D C'$.

Dass es sich nach einer solchen Permutation immer noch um den gleichen Typ von Code handelt, sichert das folgende kleine

Lemma 1.2.1. *Äquivalente Codes sind vom gleichen Typ.*

BEWEIS. Es sei $C \in \mathbb{F}^n$ ein Code vom Typ (n, k, d), also der Länge n, der Dimension k und vom Minimalabstand d. Ist nun $C' \in \mathbb{F}^n$ ein zu C äquivalenter Code vom Typ (n', k', d'), so gibt es eine Permutation $\pi \in \mathfrak{S}_n$, also eine bijektive Abbildung von C nach C', insbesondere auch der Basisvektoren von C auf eine Basis von C'. Demnach gilt neben $n = n'$ (nach Voraussetzung) also auch $k = k'$. Ist nun $c = (c_0, \ldots, c_{n-1})$ mit minimalem Hamming-Gewicht (siehe Gleichung (1.1)), also mit nur d von Null verschiedenen Komponenten, so hat auch $c' = (\pi(c_0), \ldots, \pi(c_{n-1}))$ nur d Komponenten ungleich Null. Gäbe es ein Codewort $\hat{c} \in C'$ mit Hamming-Gewicht kleiner d, so würde die zu π inverse Permutation ein Codewort in C mit ebenfalls kleinerem Hamming-Gewicht als d liefern und wir erhielten einen Widerspruch. Folglich ist $(n, k, d) = (n', k', d')$. □

Kommen wir auf unser letztes Beispiel 1.2.1 zurück, so lässt sich die Generatormatrix G mit einer Permutation $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 5 & 6 & 7 & 1 & 2 & 4 \end{pmatrix} \in \mathfrak{S}_7$ so umsortieren, dass die Matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \overset{\pi}{\mapsto} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} := G' = (I_4 \; A)$$

entsteht. Mit der zu G äquivalenten Generatormatrix G' codierte Information ist in ihren ersten vier Komponenten gerade die Information selbst. Dies liegt daran, dass die ersten vier Spalten von G' eine Einheitsmatrix I_4 sind. Die Redundanz A wird in die hinteren Stellen verschoben. Natürlich muss auch die Kontrollmatrix H entsprechend verändert und angepasst werden. Wird ein Code von einer Matrix der Form $G = (I_k \; A)$ erzeugt, so wird er auch *systematischer Code* genannt, eine zugehörige Generatormatrix heißt *Standardgeneratormatrix*.

1.3 Zyklische Codes

Bisher benötigen wir zur Beschreibung eines linearen Codes eine k-elementige Menge von Basisvektoren der Länge n, die den (n, k)-Code C (in Form einer

Generatormatrix) erzeugen. In diesem Abschnitt werden wir sehen, wie sich zyklische Codes vollständig durch Angabe eines Polynoms mit Koeffizienten aus dem Körper, über dem der Code definiert werden soll, beschreiben lassen. Die Aussagen sind aus der Vorlesung [Hol99] entnommen.

Definition 1.3.1. Ein Code C der Länge n heißt *zyklisch*, wenn für jedes Wort $(c_0, c_1, \ldots, c_{n-1})$ auch die zyklische Vertauschung $(c_{n-1}, c_0, \ldots, c_{n-2})$ in C enthalten ist.

Um das Wissen über Ringe, Körper, Polynome, etc. für die Codierungstheorie nutzbar zu machen identifizieren wir nun mittels der Abbildungen

$$\begin{array}{ccccc} \mathbb{F}^n & \to & \mathbb{F}[X]_{\deg<n} & \to & \mathbb{F}[x] := \mathbb{F}[X]/(X^n-1) \\ (c_0,\ldots,c_{n-1}) & \mapsto & c_0 + c_1X + \ldots + c_{n-1}X^{n-1} & \mapsto & c_0 + c_1x + \ldots + c_{n-1}x^{n-1} \end{array}$$

die Wörter der Länge n mit den Restklassen des Polynomrings über dem Körper $\mathbb{F}$ modulo (X^n-1). Die Abbildungen sind offensichtlich Vektorraumisomorphismen. Die erste bildet die kanonische Basis des $\mathbb{F}^n$ auf die kanonische Basis $\{1, X, X^2, \ldots, X^{n-1}\}$ ab und die zweite bildet diese mittels $x := X \bmod (X^n-1)$ auf die Basis $\{1, x, x^2, \ldots, x^{n-1}\}$ von $\mathbb{F}[x]$ ab. Damit ist auch die Verknüpfung ein Vektorraumisomorphismus und schließlich $\mathbb{F}^n \cong \mathbb{F}[x]$.
Wie stellen sich die Codewörter in dieser polynomialen Betrachtung dar?
Nach Definition 1.3.1 gilt $(c_0, c_1, \ldots, c_{n-1}) \in C \Rightarrow (c_{n-1}, c_0, \ldots, c_{n-2}) \in C$ was sich nunmehr überträgt auf $c_0 + c_1x + \ldots + c_{n-1}x^{n-1} \in \mathbb{F}[x] \Rightarrow c_{n-1} + c_0x + \ldots + c_{n-2}x^{n-1} = x \cdot (c_0 + c_1x + \ldots + c_{n-1}x^{n-1}) \in \mathbb{F}[x]$, denn $x^n = 1$ in $\mathbb{F}[x]$. Dies lässt sich für beliebige Potenzen x^i mit $0 \leq i \leq n-1$ fortführen. Deshalb ist mit $c(x)$ auch stets $t(x) \cdot c(x)$ in C enthalten, wenn $t(x)$ eine beliebige Linearkombination der Basisvektoren $1, x, x^2, \ldots, x^{n-1}$, also ein beliebiges Element aus $\mathbb{F}[x]$ ist.

Bemerkung 1.3.1. Die zyklischen Codes entsprechen offenbar gerade den Idealen des Rings $\mathbb{F}[x]$ und jedes Ideal in $\mathbb{F}[x]$ ist Hauptideal, da $\mathbb{F}[x]$ ein Hauptidealring ist.

Beweis. Es bleibt nur noch der zweite Teil zu zeigen. Dazu sei $\mathfrak{c}$ ein beliebiges Ideal aus $\mathbb{F}[x]$. Die Abbildung ρ bildet umkehrbar eindeutig $\mathbb{F}[X]_{\deg<n}$ auf $\mathbb{F}[x]$ ab. Folglich gibt es auch zu $\mathfrak{c}$ ein Ideal $\rho^{-1}(\mathfrak{c}) \subset \mathbb{F}[X]$, das sogar Hauptideal ist, da $\mathbb{F}[X]$ ein Hauptidealring ist, d. h. $\rho^{-1}(\mathfrak{c}) = (g(X))$ für ein $g(X) \in \mathbb{F}[X]_{\deg<n}$. Es ist dann $\mathfrak{c} = \rho(\rho^{-1}(\mathfrak{c})) = \rho((g(X))) = (g(x))$. Daraus folgt nun unmittelbar, dass $\mathbb{F}[x]$ Hauptidealring ist, also die Behauptung. □

Definition 1.3.2. Es sei C ein zyklischer Code der Länge n. Ein normiertes Polynom kleinsten Grades in $C \setminus \{0\} \subseteq \mathbb{F}[X]$ heißt *Generatorpolynom* von C.

Satz 1.3.1. *Es sei $g(X)$ Generatorpolynom und $c(X)$ ein beliebiges Codewort eines zyklischen (n,k)-Codes, dann gilt:*

1. $g(X) \mid c(X)$,
2. $g(X) \mid X^n - 1$ *und*
3. $g(X)$ *ist eindeutig bestimmt.*

BEWEIS. 1. Es sei $\deg g(X) = r < n$ und $c(X) \neq g(X)$ ein Codewort des von $g(X)$ erzeugten Codes C. Dann gibt es zwei eindeutig bestimmte Polynome $t(X)$ und $r(X)$, so dass gilt $c(X) = t(X)g(X) + r(X)$ mit $\deg r(X) < r$. Umgestellt nach $r(X)$ ergibt sich $r(X) = c(X) - t(X)g(X)$. Wir betrachten nun die Reste: Da $c(x)$ und $g(x)$ Codewörter sind, muss auch $r(x)$ im Ideal $(g(x))$ liegen, also ein Codewort sein. Da aber $g(X)$ nach Definition kleinsten Grades in C ist, folgt mit der Gradformel $r(X) = 0$, da andernfalls auch $r(x) \neq 0$, also die Behauptung $c(X) = t(X)g(X)$.

2. Mit der gleichen Argumentation und $X^n - 1 \equiv 0$ anstatt $c(X)$ erhält man $\deg r(X) \geq \deg t(X) + \deg g(X) > \deg g(X)$ und folglich wiederum $r(X) = 0$, also $X^n - 1 = t(X)g(X)$.

3. Es sei nun $g'(X)$ ein weiteres Generatorpolynom, dann gilt auch $g'(X) \mid g(X)$ und $g(X) \mid g'(X)$. Da Generatorpolynome nach Definition normiert sind, folgt sofort $g(X) = g'(X)$. □

Um eine Generatormatrix zu einem gegebenem zyklischen Code $C \subseteq \mathbb{F}[X]$ aufstellen zu können, benötigen wir zunächst das Generatorpolynom $g(X) = g_0 + g_1X + \ldots + g_{n-1}X^{n-1}$. Die Dimension k des Codes erhalten wir dann aus der maximal linear unabhängigen Anzahl von Basisvektoren von C in $\mathbb{F}[x]$. Als Basis bieten sich damit die Vektoren $g(x), xg(x), x^2g(x), \ldots, x^{k-1}g(x)$ an. Sei nun o.B.d.A. $g(X) = g_0 + g_1X + \ldots + g_{n-k}X^{n-k}$ dann ist

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \vdots \\ \vdots & \ddots & & & & \ddots & 0 \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}.$$

eine Generatormatrix von C.
Die Codierung einer Information $a = (a_0, \ldots, a_{k-1})$ ist nun als Multiplikation von Polynomen darstellbar:

$$a \cdot G = c \cong c(x) = ((a_0 + a_1X + \ldots + a_{k-1}X^{k-1}) \cdot g(X)) \bmod (X^n - 1).$$

Eine technische Umsetzung ist das so genannte Schieberegister (shift register), wie sich bei [Ham87], [Hei95] oder auch [van99] nachlesen lässt. Um Codewörter decodieren zu können benötigen wir eine Kontrollmatrix, bzw. genügt uns auch hier wieder ein Kontrollpolynom, wie wir gleich sehen werden. Da nach Satz 1.3.1 stets gilt $g(X) \mid X^n - 1$, gibt es also ein Polynom $h(X)$, so dass $g(X)h(X) = X^n - 1 \equiv 0$. Ist, wie oben, $g(X) = g_0 + g_1X + \ldots + g_{n-k}X^{n-k}$ das Generatorpolynom, so hat das zugehörige $h(X)$ die Form $h(X) = h_0 + h_1X + \ldots + h_kX^k$.

Definition 1.3.3. Ist C ein von $g(X) \in \mathbb{F}[X]$ erzeugter (n, k)-Code, dann heißt das normierte Polynom kleinsten Grades $h(X) \neq 0$, für das gilt $g(X)h(X) = X^n - 1 \equiv 0$ in $\mathbb{F}[x]$ das *Kontrollpolynom* des Codes C.

Damit für alle Codewörter c, d.h. für jede Linearkombination aus Zeilen von G gilt $H \cdot c^T = 0$, bietet sich als zugehörige Kontrollmatrix die Form

$$H_{zykl} := \begin{pmatrix} 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ \vdots & 0 & h_k & \cdots & h_1 & h_0 & 0 \\ 0 & & & & & & \vdots \\ h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \end{pmatrix}$$

an. Multiplizieren wir nämlich die erste Zeile von G skalar mit der ersten Zeile von H, so ist das Produkt $g_{n-k-1}h_k + g_{n-k}h_{k-1}$. Ein Vergleich der Koeffizienten von $g(x) \cdot h(x)$ mit dem Nullpolynom liefert

$$\begin{array}{rcrcl} x^0 = 1 & : & g_0h_0 + g_{n-k}h_k & = & 0 \\ x^1 & : & g_1h_0 + g_0h_1 + \ldots & = & 0 \\ & & \cdots & & \\ x^{n-1} & : & g_{n-k-1}h_k + g_{n-k}h_{k-1} & = & 0 \end{array}$$

Der Koeffizient von x^{n-1} ist genau das Skalarprodukt von oben. Beim Multiplizieren der ersten Zeile von G mit den übrigen $n - k - 1$ Zeilen von H entstehen analog die Koeffizienten von x^{n-2} bis x^k. Zusammenfassend erhalten wir das

Korollar 1.3.1. *Es sei C ein zyklischer (n, k)-Code mit dem Kontrollpolynom $h(X) \in \mathbb{F}[X]$. Ein Element $v(x) \in \mathbb{F}[x]$ ist genau dann ein Codewort aus C, wenn in $\mathbb{F}[x]$ gilt $h(x)v(x) = 0$.*

Beweis. Nach Definition ist ein Polynom $c(X)$ genau dann ein Codewort, wenn es ein Polynom $t(X) \in \mathbb{F}[X]$ gibt mit $c(X) = t(X)g(X) = t(X)\frac{X^n-1}{h(X)}$ und das ist genau dann der Fall, wenn $h(X)c(X) = t(X) \cdot (X^n - 1) \equiv 0 \bmod (X^n - 1)$ gilt. □

Anmerkung 1.3.1. Wir wollen im folgenden fordern, dass für den über $\mathbb{F}_q$ definierten zyklischen Code der Länge n gilt $ggT(n,q) = 1$. Dann gilt: ist $X^n - 1 = f_1(X)f_2(X)\cdots f_t(X)$ eine Zerlegung in (über $\mathbb{F}_q$) irreduzible Faktoren, so kommen alle $f_i(X)$ nur einfach vor, d.h. $X^n - 1$ ist separabel (vgl. [Bos01], S. 182).

Wir haben damit die Möglichkeit eine von 2^t verschiedenen Kombinationen der irreduziblen Polynome auszuwählen und daraus das Generatorpolynom eines Codes zu bilden. Das Grundprinzip, das hinter allen zyklischen Codes steckt, besteht darin, dass man einer Information bei der Multiplikation mit dem Generatorpolynom bestimmte (einfache) Nullstellen hinzufügt und mit dem Kontrollpolynom nachsieht, ob in dem empfangenen Wort noch alle zugefügten Nullstellen enthalten sind.

1.3.1 Reed-Solomon-Codes

Definition 1.3.4. Es sei $\mathbb{F}_q$ wieder ein Körper mit q Elementen, $n, d \in \mathbb{N}$ mit $n \mid (q-1)$ und $2 \leq d \leq n$ sowie ζ eine primitive n-te Einheitswurzel. Ist nun

$$g(X) = (X - \zeta)(X - \zeta^2) \cdot \ldots \cdot (X - \zeta^{d-1}) \tag{1.7}$$

das Generatorpolynom eines zyklischen Codes der Länge n, so heißt C *Reed-Solomon-Code*[2]. Gilt sogar $n = q - 1$, so heißt C *primitiver Reed-Solomon-Code*.

Reed-Solomon-Codes finden in vielen Bereichen Anwendung, wie wir in einigen Beispielen (Bsp. 1.3.1) weiter unten sehen werden. Dies liegt neben schnellen Codier- und Decodieralgorithmen vor allem an der Aussage des folgenden Satzes.

Satz 1.3.2. *Reed-Solomon-Codes sind MDS-Codes (optimale Codes).*

Beweis. Es sei C ein Reed-Solomon-Code vom Typ (n, k, d) und

$$H := \begin{pmatrix} 1 & \cdots & 1 \\ \zeta & \cdots & \zeta^{d-1} \\ \vdots & & \vdots \\ \zeta^{n-1} & \cdots & \zeta^{(d-1)(n-1)} \end{pmatrix} \tag{1.8}$$

eine $n \times (d-1)$-Matrix. Wir zeigen zunächst, dass H Kontrollmatrix von C ist, dass also für die durch H bzgl. der kanonischen Basen beschriebene lineare

[2] benannt nach I. S. Reed und G. Solomon (1960)

Abbildung $\alpha : \mathbb{F}_q^n \to \mathbb{F}_q^{d-1}$ gilt $\ker \alpha = C$. Es sei dazu $a = (a_0, \ldots, a_{n-1})$ ein beliebiges Wort aus $\mathbb{F}_q^n$ und C, g wie in Definition 1.3.4. Dann sind folgende Beziehungen offensichtlich äquivalent:

$$a \in C \Leftrightarrow a(x) \in (g(x)) \Leftrightarrow \exists t(x) \in \mathbb{F}_q[x] : \; a(x) = t(x)g(x)$$
$$\Leftrightarrow a(\zeta^i) = 0 \; \forall i = 1, \ldots, d-1 \Leftrightarrow a \in \ker \alpha.$$

Da a beliebig gewählt war gilt nunmehr $C = \ker \alpha$. Die Kontrollmatrix H hat die Gestalt einer Vandermondeschen Matrix. Da die ζ^i verschieden sind für $i = 0, \ldots, n-1$, hat H dank der bekannten Determinante den Spaltenrang $d-1$. Außerdem sind auch mindestens $d-1$ Zeilen linear unabhänging, die Dimension des Bildes ist somit $\leq d-1$. Mit dem Dimensionssatz folgt

$$n = \underbrace{\dim \ker \alpha}_{=\dim C = k} + \underbrace{\dim \operatorname{Im} \alpha}_{\leq d-1},$$

also

$$n - k + 1 \leq d.$$

Gleichzeitig gilt auch für jeden Reed-Solomon-Code die Singleton-Schranke (Satz 1.2.1) $d \leq n-k+1$ und damit insgesamt die Behauptung $d = n-k+1$.$\square$

Beispiel 1.3.1 (Audio Compact Discs). [3] Auf einer gewöhnlichen Audio CD befinden sich durchschnittlich etwa 100.000 Fehler. Dass diese bei der Wiedergabe nicht zu hören sind, ist im wesentlichen den Fähigkeiten eines Reed-Solomon-Codes zu verdanken. Man verwendet einen primitiven 2-fehlerkorrigierenden Code vom Typ $(255, 251, 5)$. Er ist definiert über dem Körper $\mathbb{F}_{2^8} = \mathbb{F}_{256} = \mathbb{F}_2^8$, wie er aus dem Grundkörper $\mathbb{F}_2$ durch Adjunktion einer primitiven 255-ten Einheitswurzel ζ entsteht. Der Grad der Körpererweiterung ist hier 8. Folglich ist ein Polynom achten Grades notwendig, um die gewünschte Erweiterung zu erzeugen. Das Polynom $X^{255}-1$ hat in $\mathbb{F}_2[X]$ das irreduzible Polynom $f(X) = X^8+X^7+X^2+X+1$ als Faktor. Mit diesem ist $\mathbb{F}_{2^8} = \mathbb{F}_2[T]/(f(T))$. Das Generatorpolynom des Codes ist

$$g(X) = (X-\zeta)(X-\zeta^2)(X-\zeta^3)(X-\zeta^4)$$

und entsprechend ist das Kontrollpolynom

$$h(X) = (X-\zeta^5) \cdot \ldots \cdot (X-\zeta^{255}) = \frac{X^{255}-1}{g(X)}.$$

[3] Quellen: http://www.stanford.edu/class/ee387/handouts/hw7.pdf, http://www.math.uni-duesseldorf.de/~klopsch/students/seminar/cd_rscode.pdf

In der Praxis werden zwei gekürzte[4] $(255, 251, 5)$-Codes, genauer ein $(32, 28, 5)$-Code und ein $(28, 24, 5)$-Code, ineinander geschachtelt (cross interleaving) und dadurch ein $32 \cdot 8 = 256$ stelliger Code realisiert. Dadurch wird insgesamt ein Satz von 24 Zeichen aus $\mathbb{F}_{2^8}$ in ein Codewort von 256 Bit aus $\mathbb{F}_2 = \{0, 1\}$ codiert. Die Informationsrate beträgt dann $\frac{28}{32} \cdot \frac{24}{28} = \frac{3}{4}$.

Beispiel 1.3.2. [5] Der NASA-Standard-Code für die Satellitenkommunikation ist ein 16-fehlerkorrigierender Reed-Solomon-Code des Typs (255,223,33), der mit einer Codeform verknüpft wird, die wir hier nicht behandeln wollen (Faltungscodes). Das Generatorpolynom des ebenfalls über $\mathbb{F}_{2^8}$ definierten Codes ist

$$g(X) = \prod_{j=112}^{143} (X - \zeta^{11j}).$$

Beispiel 1.3.3. [6] Beim digitalen Fernsehen wird für die Übertragung durch das Kabelnetz ein (204,188,17)-Reed-Solomon-Code verwendet. Für die Übertragung via Antenne oder Satellit wird ein zusätzlicher Code zur Verknüpfung notwendig, da diese Verbindungsarten störungsanfälliger sind.

Wir wollen an dieser Stelle noch nicht auf Decodier- und Fehlerkorrekturalgorithmen eingehen, sondern verweisen auf den folgenden Abschnitt über BCH-Codes.
Das Grundkonzept der Reed-Solomon-Codes lässt sich noch verallgemeinern.

Definition 1.3.5. Es seien $v = (v_0, \ldots, v_{n-1})$ und $a = (a_0, \ldots, a_{n-1})$ aus $\mathbb{F}_{q^m}^n$, wobei $v_i \neq 0$ und $a_i \neq a_j$ für $i, j = 0, \ldots, n-1$. Der *generalisierte Reed-Solomon-Code* $GRS_k(a, v)$ hat als Codewörter alle n-Tupel

$$(v_0 f(a_0), v_1 f(a_1), \ldots, v_{n-1} f(a_{n-1})). \tag{1.9}$$

Dabei durchläuft f die Menge der Polynome vom Grad kleiner als k, also $f \in \mathbb{F}_{q^m}[x]_{\deg<k}$.

Sind nun $v = (1, \ldots, 1)$ und $a = (\zeta, \zeta^2, \ldots, \zeta^n)$ ($n = q-1$ und ζ primitive n-te Einheitswurzel), so erkennen wir unseren Reed-Solomon-Code aus Definition 1.3.4 wieder. Der Code besteht nämlich in diesem Fall aus folgender Menge von Wörtern

$$C = \{(f(\zeta), \ldots, f(\zeta^n));\ f \in \mathbb{F}_q[X]_{\deg<k}\}.$$

[4] Kürzen bedeutet nichts anderes, als das die ersten 223 bzw. 227 Zeilen und Spalten der ursprünglichen 251×255-Matrix gestrichen werden. d bleibt dadurch unverändert.

[5] Quelle: http://www.cambr.uidaho.edu/chips/rs16.pdf

[6] Quelle: http://www.radynecomstream.com/pdf/reedsol.pdf

Für das Hamming-Gewicht eines solchen Wortes gilt

$$w(c) = n - \#\{i;\ f(\zeta^i) = 0\} \geq n - \deg f \geq n - k + 1.$$

Wegen der Singleton-Schranke ist $n - k + 1 \geq d$, also

$$w(c) \geq n - k + 1 = d = \min\{w(c);\ c \in C\}.$$

Die Optimalität folgt unmittelbar und klar aus der Diagonal-Ähnlichkeit zu Reed-Solomon-Codes in Definition 1.3.5, da Diagonal-Ähnlichkeit offensichtlich den Typ erhält.

1.3.2 BCH-Codes

Definition 1.3.6. Ein zyklischer Code der Länge n über dem Körper $\mathbb{F}_q$ heißt *BCH[7]-Code zum Entwurfsabstand* δ ($\delta \geq 1$), wenn sein Generatorpolynom das kleinste gemeinsame Vielfache der zu $\zeta^l, \zeta^{l+1}, \ldots, \zeta^{l+\delta-2}$ gehörigen Minimalpolynome für ein beliebiges $l \in \mathbb{Z}$ ist, wobei $\zeta \in \mathbb{F}_{q^s}$ ($s \in \mathbb{N}$) eine primitive n-te Einheitswurzel ist.
Ist $l = 1$, so heißt der Code *BCH-Code im engeren Sinne*.
Wenn $n = q^s - 1$, wenn also ζ ein primitives Element von $\mathbb{F}_{q^s}$ ist, dann wird der BCH-Code *primitiv* genannt.

Korollar 1.3.2. *Reed-Solomon-Codes sind primitive BCH-Codes.*

Im weiteren wollen wir nur BCH-Codes im engeren Sinne betrachten, d.h. $l = 1$. Die Bezeichnung „... zum Entwurfsabstand δ“ begründet der folgende

Satz 1.3.3 (BCH-Schranke). *Der Minimalabstand d eines BCH-Codes zum Entwurfsabstand δ ist mindestens δ.*

Beweis. [Kai00] Es sei also C ein BCH-Code zum Entwurfsabstand δ der Länge n. Zunächst suchen wir wieder die Kontrollmatrix H (siehe Beweis von Satz 1.3.2). Die $(\delta - 1) \times n$-Matrix mit Einträgen aus $\mathbb{F}_{q^s}$

$$H := \begin{pmatrix} 1 & \zeta & \cdots & \zeta^{(n-1)} \\ 1 & \zeta^2 & \cdots & \zeta^{2(n-1)} \\ \vdots & & & \vdots \\ 1 & \zeta^{\delta-1} & \cdots & \zeta^{(\delta-1)(n-1)} \end{pmatrix}$$

[7] benannt nach R.C. Bose und D.K. Ray-Chaudhuri (1960) sowie A. Hocquenghem (1959)

erfüllt die Bedingung $Hc^T = 0$ genau dann, wenn c Codewort aus C ist mit der gleichen Begründung wie in Satz 1.3.2 (c^T liegt im Kern von H). Wir interpretieren die Körpererweiterung $\mathbb{F}_{q^s}$ wieder als s-dimensionalen Vektorraum über $\mathbb{F}_q$. Dazu wählen wir eine Basis $\mathfrak{B}$ von $\mathbb{F}_{q^s}$ über $\mathbb{F}_q$ aus. Ersetzen wir in der Kontrollmatrix H jeden Eintrag aus $\mathbb{F}_{q^s}$ durch den durch $\mathfrak{B}$ bestimmten entsprechenden Spaltenvektor aus $\mathbb{F}_q^s$, so erhalten wir eine $s(\delta-1) \times n$-Matrix $\hat{H}$. Dann ist der Code $C \subseteq \mathbb{F}_q^n$ genau der Kern der Matrix $\hat{H}$. Insbesondere ist die Minimaldistanz von C gleich der minimalen Anzahl von über $\mathbb{F}_q$ linear abhängigen Spalten von $\hat{H}$. Nach Konstruktion von $\hat{H}$ ist eine Menge von Spalten von $\hat{H}$ genau dann linear abhängig, wenn die entsprechende Menge von Spalten von H linear abhängig ist. Folglich genügt es, zu zeigen, dass die Determinante jeder $(\delta-1)\times(\delta-1)$-Untermatrix von H ungleich Null ist. Die Determinanten, die auf diese Weise entstehen sind wiederum allesamt Vandermondesche Determinanten. Seien etwa $j_1, j_2, \ldots, j_{\delta-1} \in \{0, 1, \ldots, n-1\}$ Spaltenindizes einer solchen Untermatrix

$$B := \begin{pmatrix} \zeta^{j_1} & \zeta^{j_2} & \ldots & \zeta^{j_{\delta-1}} \\ (\zeta^2)^{j_1} & (\zeta^2)^{j_2} & \ldots & (\zeta^2)^{j_{\delta-1}} \\ \vdots & \vdots & & \vdots \\ (\zeta^{\delta-1})^{j_1} & (\zeta^{\delta-1})^{j_2} & \ldots & (\zeta^{\delta-1})^{j_{\delta-1}} \end{pmatrix}.$$

Die Exponenten können vertauscht werden und wir erhalten eine Vandermondesche Matrix

$$B = \begin{pmatrix} \zeta^{j_1} & \zeta^{j_2} & \ldots & \zeta^{j_{\delta-1}} \\ (\zeta^{j_1})^2 & (\zeta^{j_2})^2 & \ldots & (\zeta^{j_{\delta-1}})^2 \\ \vdots & \vdots & & \vdots \\ (\zeta^{j_1})^{\delta-1} & (\zeta^{j_2})^{\delta-1} & \ldots & (\zeta^{j_{\delta-1}})^{\delta-1} \end{pmatrix}$$

für deren Determinante gilt

$$\det(B) = \zeta^{j_1+j_2+\ldots+j_{\delta-1}} \prod_{a<b} (\zeta^{j_b} - \zeta^{j_a}) \neq 0.$$

Dies gilt nur, da $1, \zeta, \zeta^2, \ldots, \zeta^{n-1}$ paarweise verschieden sind (ζ war ja primitive n-te Einheitswurzel) und die Spaltenindizes j_i ebenso. Also sind die Spalten beliebiger $(\delta-1)\times(\delta-1)$-Untermatrix von H linear unabhängig und damit hat ein beliebiges Codewort $c \neq 0$ einen Minimalabstand $\geq \delta$. □

Bevor wir darauf eingehen, wie BCH-Codes decodiert und mögliche Fehler erkannt und korrigiert werden können, wollen wir ein kurzes Beispiel für einen BCH-Code geben.

Beispiel 1.3.4. ([van99], S. 92) Wir wollen einen primitiven BCH-Code im engeren Sinne der Länge $n = 31 = 2^5 - 1$ ($q = 2, s = 5$) über dem Körper $\mathbb{F}_{2^5}$ zum Entwurfsabstand $\delta = 8$ konstruieren. Es sei ζ eine primitive 31-te Einheitswurzel aus $\mathbb{F}_{2^5}$. Dann ist das Generatorpolynom des primitiven BCH-Codes nach Definition 1.3.6 das kleinste gemeinsame Vielfache der Minimalpolynome von $\zeta, \zeta^2, \ldots, \zeta^7$. Die Minimalpolynome $m_i(X)$ von ζ^i für $i = 1, \ldots, 7$ sind

$$
\begin{aligned}
m_1(X) &= (X-\zeta)(X-\zeta^2)(X-\zeta^4)(X-\zeta^8)(X-\zeta^{16}) = m_2(X) = m_4(X),\\
m_3(X) &= (X-\zeta^3)(X-\zeta^6)(X-\zeta^{12})(X-\zeta^{24})(X-\zeta^{17}) = m_6(X),\\
m_5(X) &= (X-\zeta^5)(X-\zeta^{10})(X-\zeta^{20})(X-\zeta^{9})(X-\zeta^{18}) = m_{10}(X) = m_9(X),\\
m_7(X) &= (X-\zeta^7)(X-\zeta^{14})(X-\zeta^{28})(X-\zeta^{25})(X-\zeta^{19}).
\end{aligned}
$$

Das Generatorpolynom ist das Produkt $g(X) = m_1(X)m_3(X)m_5(X)m_7(X)$. Da aber $m_5(X) = m_{10}(X) = m_9(X)$ gilt, sind im Generatorpolynom auch ζ^9 und ζ^{10} Nullstelle und damit der tatsächliche Minimalabstand wenigstens 11.

Wir wollen nun den oben angekündigten Fehlerkorrekturalgorithmus vorstellen. Dieser kann z. B. bei [Hei95], [van99] oder [Kai00] nachgelesen werden.
Situation: Sei $C \subseteq \mathbb{F}_q^n$ ein BCH-Code zum Entwurfsabstand $\delta = 2t + 1$ und $\zeta \in \mathbb{F}_{q^s}$ wieder eine primitive n-te Einheitswurzel. Im weiteren gelten folgende Bezeichnungen:

- gesendetes Wort $C(x) = C_0 + C_1x + \ldots + C_{n-1}x^{n-1} \in C$,
- empfangenes Wort $R(x) = R_0 + R_1x + \ldots + R_{n-1}x^{n-1} \in \mathbb{F}_q^n$,
- Fehlerpolynom $E(x) := R(x) - C(x) = E_0 + E_1x + \ldots + E_{n-1}x^{n-1} \in \mathbb{F}_q^n$,
- Fehlerpositionen $M := \{i \in \{0, 1, \ldots, n-1\};\ E_i \neq 0\}$,
- Anzahl der Fehler $e := \#M$,
- Lokatorpolynom $\sigma(x) := \prod_{i \in M}(1 - \zeta^i x) = \sigma_0 + \sigma_1 x + \ldots + \sigma_m x^m$ und
- Auswertepolynom $\omega(x) := \sum_{i \in M} E_i \zeta^i x \prod_{j \in M \setminus \{i\}}(1 - \zeta^j x)$.

Zunächst stellen wir fest, dass auftretende Fehler korrigierbar sind, wenn wir in der Lage sind $\sigma(x)$ und $\omega(x)$ auszurechnen. Tritt nämlich ein Fehler etwa an der i-ten Position auf, dann ist $\sigma(\zeta^{-i}) = 0$. Der aufgetretene Fehler berechnet sich mit Hilfe des Auswertepolynoms zu

$$E_i = \frac{-\omega(\zeta^{-i})\zeta^i}{\sigma'(\zeta^{-i})}. \tag{1.10}$$

Kennen wir alle E_i, so können wir mittels $C(x) = R(x) - E(x)$ auf das korrekte Codewort schließen. Wir unterscheiden zwei Fälle:

1. $e > t$: Allgemein ist hier eine Fehlerkorrektur nicht möglich, da die Anzahl der Fehler den Fehlerkorrekturindex übersteigt. Im speziellen sind dennoch Korrekturen abschätzbar.

2. $e \leq t$: In diesem Fall ist eine Fehlerkorrektur grundsätzlich denkbar. Zunächst sehen wir, dass mit $(*)$ $\zeta^i x = \sum_{l=1}^{\infty} (\zeta^i x)^l - \sum_{l=2}^{\infty} (\zeta^i x)^l$ gilt

$$\begin{aligned} \frac{\omega(x)}{\sigma(x)} &= \sum_{i \in M} \frac{E_i \zeta^i x}{1 - \zeta^i x} \overset{(*)}{=} \sum_{i \in M} E_i \sum_{l=1}^{\infty} (\zeta^i x)^l = \sum_{l=1}^{\infty} x^l \sum_{i \in M} E_i \zeta^{il} \\ &= \sum_{l=1}^{\infty} x^l E(\zeta^l). \end{aligned} \tag{1.11}$$

   Wir wissen weiter, dass für das Syndrom

$$S_l := E(\zeta^l) = R(\zeta^l) - C(\zeta^l) = R(\zeta^l) \tag{1.12}$$

   mit $1 \leq l \leq 2t$ ist, da C nach Voraussetzung ein BCH-Code zum Entwurfsabstand $2t + 1$ ist. Demnach sind die ersten $2t$ Summanden obiger Darstellung von $\omega(x)/\sigma(x)$ bekannt; die übrigen streichen wir. Fassen wir nun die Gleichungen (1.11) und (1.12) zusammen, dann erhalten wir

$$\begin{aligned} \omega(x) = \sigma(x) \sum_{l=1}^{2t} S_l x^l &= \left(\sum_{i=0}^{e} \sigma_i x^i \right) \cdot \left(\sum_{l=1}^{2t} S_l x^l \right) \\ &= \sum_{k=1}^{2t} x^k \left(\sum_{i+l=k} S_l \sigma_i \right). \end{aligned} \tag{1.13}$$

   Da nach Konstruktion $\deg \omega(x) \leq e$, ist $\sum_{k=i+l} S_l \sigma_i = 0$ für $e < k \leq 2t$. Wir haben damit ein homogenes lineares Gleichungssystem mit $2t - e$ Gleichungen in den $e + 1$ Unbekannten $\sigma_0, \sigma_1, \ldots, \sigma_e$ gefunden und wir wissen bereits, dass $\sigma_0 = 1$. Auf jeden Fall ist $\sigma(x)$ Lösung des Systems. Sei $\tilde{\sigma}(x) = \sum_{i=0}^{e} \tilde{\sigma}_i x^i$ die Lösung kleinsten Grades. Dann haben wir für $e < k \leq 2t$

$$0 = \sum_{l} S_{k-l} \tilde{\sigma}_l = \sum_{i \in M} \sum_{l} E_i \zeta^{(k-l)i} \tilde{\sigma}_l = \sum_{i \in M} E_i \zeta^{ik} \tilde{\sigma}(\zeta^{-i}), \tag{1.14}$$

also wieder ein lineares Gleichungssystem mit Koeffizienten ζ^{ik}. Ist $\{i_1, \ldots, i_e\} = M$ die Menge der Fehlerpositionen, dann hat die Koeffizientenmatrix die Form

$$\begin{pmatrix} \zeta^{i_1(2t-e+1)} & \ldots & \zeta^{i_e 2t} \\ \vdots & \ddots & \vdots \\ \zeta^{i_e(2t-e+1)} & \ldots & \zeta^{i_e 2t} \end{pmatrix} = \begin{pmatrix} \zeta^{(2t-e+1)^{i_1}} & \ldots & \zeta^{(2t)^{i_1}} \\ \vdots & \ddots & \vdots \\ \zeta^{(2t-e+1)^{i_e}} & \ldots & \zeta^{(2t)^{i_e}} \end{pmatrix}.$$

Dies ist wieder eine Vandermondesche Matrix, deren Determinante ungleich Null ist, also ist $E_i\tilde{\sigma}(\zeta^{-i}) = 0$ für $i \in M$ die triviale Lösung. Da $E_i \neq 0$ für alle $i \in M$, muss $\tilde{\sigma}(\zeta^{-i}) = 0$ für alle $i \in M$. Damit sind alle $(1 - \zeta^i x)$ und damit auch das Produkt $\prod_{i \in M}(1 - \zeta^i x)$ Teiler von $\tilde{\sigma}(x)$ also gilt $\sigma(x) \mid \tilde{\sigma}(x)$. Da $\tilde{\sigma}(x)$ minimalen Grades war, gilt aber auch $\tilde{\sigma}(x) \mid \sigma(x)$ und damit die Gleichheit.

Wir haben nunmehr einen Algorithmus gefunden, der es uns ermöglicht bis zu t Fehler eines mit einem BCH-Code zum Entwurfsabstand $\delta = 2t + 1$ codierten Wortes zu korrigieren. In der Technik erfreuen sich weniger theoretische, dafür schnellere Algorithmen, wie etwa der von Berlekamp/Massey großer Beliebtheit. Diese beruhen auf der Lösung eines Kongruenzsystems

$$S(x)\sigma(x) \equiv \omega(x) \bmod x^{2t+1}.$$

Die Arbeitsweise unseres Algorithmus wollen wir in einem Beispiel testen.

Beispiel 1.3.5. (siehe auch Anhang A.1) Gegeben ist ein 2-fehlerkorrigierender BCH-Code vom Typ $(15, 7, 5)$ über $\mathbb{F}_{2^4} = \mathbb{F}_2[T]/(T^4 + T + 1)$ mit dem Generatorpolynom $g(X) = X^8 + X^7 + X^6 + X^4 + 1$. Das empfangene Wort ist $r = (111110111100101)$. Dies entspricht $R(x) = x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^6 + x^5 + x^2 + 1$.

1. Gehört r zum Code, so teilt $g(x)$ das empfangene Wort $R(x)$.
   Ergebnis: $g(x) \nmid R(x)$, also $R(x)$ kein Codewort.

2. Das Syndrom von $R(x)$ wird berechnet. Dazu wird überprüft, ob die ersten $d - 1 = 4$ Potenzen der primitiven 15-ten Einheitswurzel ζ, die von dem Minimalpolynom $m_\zeta(x) = x^4 + x + 1$ erzeugt wird, Nullstellen des Polynoms $R(x)$ sind.
   Ergebnis: $S_1 = \zeta^4, S_2 = \zeta^8, S_3 = \zeta, S_4 = \zeta$.

3. Nun wird das Lokatorpolynom ausgerechnet. Wie oben beschrieben erhalten wir ein 2×2-Gleichungssystem mit Koeffizienten $S_1, \ldots, S_4$

und Unbekannten σ_1, σ_2 ($\sigma_0 = 1$ ist bekannt):

$$S_1\sigma_2 + S_2\sigma_1 + S_3\sigma_0 = 0$$
$$S_2\sigma_2 + S_3\sigma_1 + S_4\sigma_0 = 0$$

Ergebnis: $(\sigma_1, \sigma_2) = (\zeta^4, \zeta^9)$ und damit $\sigma(x) = \zeta^9 x^2 + \zeta^4 x + 1$.

4. Nach Konstruktion sind die Exponenten der Nullstellen des Lokatorpolynoms gerade die Fehlerstellen des Wortes $R(x)$.
   Nullstellen: $x_1 = \zeta^{11}, x_2 = \zeta^{13}$.
   Es ist nur noch das Fehlerpolynom zu berechnen und der Fehler mittels $C(x) = R(x) - E(x)$ zu korrigieren. Das Fehlerpolynom lautet also $E(x) = x^{13} + x^{11}$.
   Ergebnis: Das korrigierte Wort ist dann $C(x) = x^{14} + x^{12} + x^{10} + x^8 + x^7 + x^6 + x^5 + x^2 + 1$. Der zur Probe berechnete Syndromvektor des korrigierten Wortes ist tatsächlich der Nullvektor.

Der Algorithmus lässt sich dank Korollar 1.3.2 problemlos auch auf die in Abschnitt 1.3.1 beschriebenen Reed-Solomon-Codes anwenden. Das Beispiel 1.3.5 ist ein 2-fehlerkorrigierender Code mit Informationsrate $\frac{7}{15}$. Legen wir den Körper $\mathbb{F}_{2^6}$ zugrunde, kann man mit Generatorpolynomen vom Grad ≤ 12 folgende weitere BCH-Codes vom Typ $(63, k)$ konstruieren:

Anzahl Informationsstellen: k	Anzahl korrigierbare Fehler: t	$d \geq \delta$
30	6	13
24	7	15
18	10	21
16	11	23
10	13	27
7	15	31

Im Satellitensystem INMARSAT[8] wird (neben anderen) ein $(63, 39)$-BCH-Code zur Datenübertragung eingesetzt.

[8] INMARSAT betreibt ein Netzwerk von derzeit neun geostationärer Satelliten, die mobile Kommunikation via Satellit ermöglichen.

Kapitel 2

Goppa-Codes

Wir beschäftigen uns im zweiten Kapitel mit einer Klasse von Codes, die noch relativ neu ist. Der russische Mathematiker V.D. Goppa kam um 1970 bei seinen Untersuchungen der BCH-Codes auf die Idee, diese als lineare Konstrukte auf der projektiven Geraden über einem endlichen Körper zu betrachten. Das brachte ihn im weiteren dazu, neuartige fehlerkorrigierende Codes auf linearen Räumen über algebraischen Kurven zu entwickeln. Von zentraler Bedeutung für die Bestimmung des Typs der Goppa-Codes ist der Satz von Riemann-Roch, der im Abschnitt 2.2.3 besprochen wird. Am Ende des Kapitels werden wir sehen, wie uns der ursprünglich aus der Funktionentheorie stammende Residuensatz eine Kontrolle der rational-geometrischen Goppa-Codes ermöglicht.

2.1 Klassische- und affin-lineare Goppa-Codes

Wir betrachten wieder einen endlichen Körper $\mathbb{F}_q$ mit einer Primzahlpotenz $q = p^m$.

Definition 2.1.1. Es sei $g(x) = g_0 + g_1 x + \ldots + g_t x^t$ ein Polynom vom Grad t aus $\mathbb{F}_{q^s}[x]$ ($s \in \mathbb{N}$). Sei $D = (p_0, \ldots, p_{n-1}) \in \mathbb{F}_{q^s}^n$, so dass $g(p_i) \neq 0$ für alle $0 \leq i \leq n-1$. Wir definieren den *klassischen Goppa-Code* $C := \Gamma(D, g)$ als die Menge von Codewörtern $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$, für die gilt

$$\sum_{i=0}^{n-1} \frac{c_i}{x - p_i} \equiv 0 \bmod g(x). \tag{2.1}$$

Anmerkung 2.1.1. Äquivalent dazu können klassische Goppa-Codes auch mittels einer Kontrollmatrix definiert werden (siehe [Sti93], S. 54).

Unter den gleichen Voraussetzungen, wie oben wird dabei zunächst ganz allgemein ein Code $C(D, g(x))$ über $\mathbb{F}_{q^s}$ definiert durch die Generatormatrix

$$H := \begin{pmatrix} g(p_0)^{-1} & g(p_1)^{-1} & \cdots & g(p_{n-1})^{-1} \\ p_0 g(p_0)^{-1} & p_1 g(p_1)^{-1} & \cdots & p_{n-1} g(p_{n-1})^{-1} \\ \vdots & \vdots & & \vdots \\ p_0^{t-1} g(p_0)^{-1} & p_1^{t-1} g(p_1)^{-1} & \cdots & p_{n-1}^{t-1} g(p_{n-1})^{-1} \end{pmatrix}. \tag{2.2}$$

Die Einschränkung des zu diesem Code dualen Codes auf $\mathbb{F}_q^n$

$$\Gamma(D, g(x)) := C(D, g(x))^{\perp} \cap \mathbb{F}_q^n, \text{ also}$$

$$\Gamma(D, g(x)) = \left\{ c \in \mathbb{F}_q^n;\ H \cdot c^T = 0 \right\}$$

heißt *klassischer Goppa-Code* mit dem *Goppa-Polynom* $g(x)$.
In dieser Definition ist der Bezug zu den generalisierten Reed-Solomon-Codes (Definition 1.3.5) deutlich sichtbar. Wählen wir nämlich $v_i := g(p_i)^{-1}$ und $a_i = p_i$ für $i = 0, \ldots, n-1$, sowie $k = t$, dann ist H Generatormatrix von $GRS_k(a, v)$, denn x^j für $j = 0, \ldots, k-1$ ist Basis von $\mathbb{F}[x]_{\deg<k}$. Damit ist $\Gamma(D, g(x))^{\perp} = GRS_k(a, v)$, der klassische Goppa-Code $\Gamma(D, g(x))$ also offenbar dual zu einem $GRS_k(a, v)$-Code.

Beispiel 2.1.1. Es sei ζ wieder eine n-te primitive Einheitswurzel in $\mathbb{F}_{q^s}$, $g(x) = x^{\delta-1}$ das Goppa-Polynom und $D = (1, \zeta^{-1}, \ldots, \zeta^{-(n-1)})$ das n-Tupel. Der daraus konstruierte klassische Goppa-Code $\Gamma(D, g(x))$ ist ein BCH-Code zum Entwurfsabstand δ im engeren Sinne. Die $(\delta - 1 \times n)$-Kontrollmatrix bilden wir aus den gegebenen Parametern $g(x)$ und D, wie oben zu

$$\begin{aligned} H &= \begin{pmatrix} 1 & \zeta^{(\delta-1)} & \cdots & \zeta^{(\delta-1)(n-1)} \\ 1 & \left(\zeta^{-1}\right)^1 \cdot \zeta^{(\delta-1)} & \cdots & \left(\zeta^{-(n-1)}\right)^1 \cdot \zeta^{(n-1)(\delta-1)} \\ \vdots & \vdots & & \vdots \\ 1 & \left(\zeta^{-1}\right)^{(\delta-2)} \cdot \zeta^{(\delta-1)} & \cdots & \left(\zeta^{-(n-1)}\right)^{(\delta-2)} \cdot \zeta^{(n-1)(\delta-1)} \end{pmatrix} \\ &= \begin{pmatrix} 1 & \zeta^{\delta-1} & \cdots & \zeta^{(\delta-1)(n-1)} \\ 1 & \zeta^{\delta-2} & \cdots & \zeta^{(\delta-2)(n-1)} \\ \vdots & \vdots & & \vdots \\ 1 & \zeta^1 & \cdots & \zeta^{1(n-1)} \end{pmatrix} \end{aligned}$$

und erkennen die Kontrollmatrix aus dem Beweis von Satz 1.3.3 wieder, wenn wir die Reihenfolge der Zeilen entsprechend vertauschen.

Bevor wir im Abschnitt 2.3, den Ideen Goppas folgend, Goppa-Codes auf der projektiven Geraden betrachten werden, wollen wir zunächst eine affine Interpretation geben.

Definition 2.1.2. Es seien $\mathbb{A}^n(\mathbb{F}) = \mathbb{F}^n$ der n-dimensionale affine Raum über dem Körper $\mathbb{F}$ und $D := (p_0, \ldots, p_{n-1}) \in (\mathbb{A}^1)^n$ ein System von Punkten der affinen Gerade $\mathbb{A}^1(\mathbb{F})$, wobei $p_i \neq p_j$ für alle $i \neq j$.

1. Die Abbildung

$$\begin{aligned} w_D : \mathbb{F}[X] &\to \mathbb{F}^n \\ f(X) &\mapsto f(D) := (f(p_0), \ldots, f(p_{n-1})) \end{aligned}$$

   heißt *Werte-Abbildung von D.*

2. Es sei $L \subseteq \mathbb{F}[X]$ ein bzgl. $\mathbb{F}$ linearer Unterraum. Das Bild von L unter der Werte-Abbildung w_D

$$C(D, L) := w_D(L) \subseteq \mathbb{F}^n$$

   heißt *affin-linearer Goppa-Code* mit den Parametern D und L auf der affinen Geraden $\mathbb{A}^1(\mathbb{F})$.

Wir stellen fest, dass generalisierte Reed-Solomon-Codes $GRS_k(a, v)$ mit $a = D$ und $v = (1, \ldots, 1)$ affin-lineare Goppa-Codes sind, denn es ist $\mathbb{F}[X]_{\deg<k}$ ein $\mathbb{F}$ linearer Unterraum von $\mathbb{F}[X]$ mit Basis $1, X, \ldots, X^{k-1}$. Insbesondere sind damit auch alle Reed-Solomon-Codes affin-lineare Goppa-Codes.

2.2 Grundbegriffe II

Bevor wir im nächsten Abschnitt zu geometrischen Goppa-Codes kommen, müssen wir noch einige wichtige Begriffe und Zusammenhänge erklären, auch wenn hier nicht immer sofort der Bezug zu den Codes sichtbar ist.

2.2.1 Geometrische Überlegungen - vom affinen zum projektiven Raum

Bisher haben wir Goppa-Codes über der *affinen Geraden* $\mathbb{A}^1(\mathbb{F})$ betrachtet. Den *affinen Raum* $\mathbb{A}^n$ über einem Körper $\mathbb{F}$ können wir uns allgemein als *Punktraum* vorstellen. Seine Elemente sind Punkte, die durch jeweils n Koordinaten aus dem Grundkörper $\mathbb{F}$ gegeben sind. Der affine Raum $\mathbb{A}^n(\mathbb{F})$ selbst ist kein Vektorraum: in ihm sind die Addition zweier Elemente, sowie

die Multiplikation eines Punktes mit einem Element aus dem Grundkörper (skalare Multiplikation) nicht definiert.
Betrachten wir nun den $n+1$-dimensionalen affinen Raum über $\mathbb{F}$ ohne den Nullpunkt $(0, 0, \ldots, 0)$ und definieren darauf eine Äquivalenzrelation

$$(a_0, a_1, \ldots, a_n) \sim (\lambda a_0, \lambda a_1, \ldots, \lambda a_n),$$

wobei $\lambda \in \mathbb{F} \setminus \{0\}$. Dann ist jede Äquivalenzklasse eindeutig durch ein *System homogener Koordinaten* $(a_0, a_1, \ldots, a_n)$ bestimmt. Die Klasse wird mit $(a_0 : a_1 : \ldots : a_n)$ bezeichnet. Durch diese Äquivalenzrelation wird der n-dimensionale *projektive Raum* über dem Körper $\mathbb{F}$ definiert als

$$\mathbb{P}^n(\mathbb{F}) = \{(a_0 : a_1 : \ldots : a_n);\ (a_0, a_1, \ldots, a_n) \in \mathbb{A}^{n+1}(\mathbb{F}) \setminus \{(0, 0, \ldots, 0)\}\}.$$

$P := (a_0 : a_1 : \ldots : a_n)$ heißt *projektiver Punkt.*
Die *projektive Gerade* ist der 1-dimensionale projektive Raum

$$\mathbb{P}^1(\mathbb{F}) = \{(a_0 : a_1);\ (a_0, a_1) \in \mathbb{A}^2(\mathbb{F}) \setminus \{(0, 0)\}\}.$$

Da $\mathbb{F}$ ein Körper ist, existiert ein multiplikatives Inverses zu $a_0 \neq 0$, so dass sich mit $a := \frac{a_1}{a_0}$ die projektive Gerade schreiben lässt als

$$\mathbb{P}^1(\mathbb{F}) = \{(1 : a);\ a \in \mathbb{F}\} \cup \{(0 : 1)\}.$$

Die Menge der projektiven Punkte, aus denen die projektive Gerade besteht, kann demnach als Menge der Steigungen $a \in \mathbb{F}$ affiner Ursprungsgeraden in $\mathbb{A}^2$, bzw. als affine Gerade $\{(1 : a);\ a \in \mathbb{F}\}$ interpretiert werden. Da wir oben $a_0 = 0$ nicht berücksichtigt haben, dieser Fall für $a_1 \neq 0$ jedoch eintreten kann, müssen wir noch eine Gerade mit Steigung ∞ hinzufügen und definieren $\infty := (0 : 1)$ als den *unendlich fernen Punkt.*
Wir fassen zusammen:

- $\mathbb{P}^1(\mathbb{F}) = \mathbb{A}^1(\mathbb{F}) \cup \{\infty\}$ ist die projektive Gerade.
- Die Punkte P der projektiven Geraden (bis auf $(0 : 1)$) sind eindeutig durch einen Parameter $a \in \mathbb{F}$ bestimmt, d.h. $P = P(a)$.
- Der Punkt $(0 : 1)$ wird als ∞ definiert und heißt der unendlich ferne Punkt.

Im folgenden ist $\mathbb{P}^1 = \mathbb{P}^1(\mathbb{F})$.

2.2.2 Funktionenkörper, Stellen, Divisoren, Funktionenräume und Differentialformen

Der gesamte Abschnitt orientiert sich weitestgehend an der Vorlesung [Hol99]. An einigen besonders gekennzeichneten Stellen haben wir Ergänzungen vorgenommen.

Funktionenkörper

Der über unserem Körper $\mathbb{F}$ definierte Polynomring $\mathbb{F}[X]$ ist nullteilerfrei und besitzt ein Einselement, ist also ein Integritätsbereich. Wir sind daher in der Lage einen Quotientenkörper über $\mathbb{F}[X]$ zu definieren. Dabei orientieren wir uns an der Konstruktion der rationalen Zahlen als Brüche ganzer Zahlen (vgl. [Bos01]). Durch Äquivalenzklassenbildung entsteht dann

$$F := \mathbb{F}(X) = Q(\mathbb{F}[X]) = \left\{ \frac{f(X)}{g(X)};\ f(X), g(X) \in \mathbb{F}[X], g(X) \neq 0 \right\}$$

der *Körper der rationalen Funktionen* einer Variable X mit Koeffizienten aus $\mathbb{F}$ oder einfach der *rationale Funktionenkörper*. Von zentraler Bedeutung für die geometrischen Goppa-Codes ist das Verhalten der rationalen Funktionen in den Punkten der projektiven Geraden $\mathbb{P}^1$. Insbesondere das Auftreten von Null- oder Polstellen von rationalen Funktionen in diesen Punkten und ihre Vielfachheiten werden uns im folgenden beschäftigen. Wir wollen hier bereits folgende 1 : 1-Zuordnung zwischen den Punkten $P = P(a)$ und ∞ von $\mathbb{P}^1(\mathbb{F})$ und dem Funktionenkörper $\mathbb{F}(X)$ treffen:

$$P = P(a) \longleftrightarrow (X - a) \text{ und } \infty \longleftrightarrow X' := \tfrac{1}{X}.$$

Stellen

Definition 2.2.1. Es sei $\varphi(X) = \frac{f(X)}{g(X)}$ mit $g(X) \neq 0$ eine beliebige rationale Funktion aus $\mathbb{F}(X) \setminus \{0\}$. Da $\mathbb{F}[X]$ ein ZPE-Ring ist, existiert zu vorgegebenem $p(X)$ eine eindeutige Darstellung $\varphi(X) = \frac{f(X)}{g(X)} = p(X)^k \frac{f_1(X)}{g_1(X)}$, wobei $p(X)$ ein Primpolynom ist, $f_1(X)$, $g_1(X)$ teilerfremd zu $p(X)$ sind und $g_1(X) \neq 0$ gilt. Die Abbildung

$$v_{p(X)}: \quad \mathbb{F}(X) \to \mathbb{Z}, \quad \varphi(X) = p(X)^k \frac{f_1(X)}{g_1(X)} \mapsto v_{p(X)}(\varphi(X)) = k \tag{2.3}$$

heißt eine *Bewertung von φ bzgl. eines Primpolynoms $p(X)$*. Um alle rationalen Funktionen bewerten zu können, setzen wir noch $v_{p(X)}(0) := \infty_{\mathbb{Z}}$.
Mit $X' := \frac{1}{X}$ definieren wir weiter

$$v_{X'}(\varphi) := -\deg f(X) + \deg g(X) \tag{2.4}$$

und analog $v_{X'}(0) := \infty_{\mathbb{Z}}$.

Die Bewertung der rationalen Funktionen des Funktionenkörpers $\mathbb{F}(X)$ bezüglich der Primpolynome $p(X) \in \mathbb{F}(X)$ und X' ist ein Gruppenhomomorphismus von $(\mathbb{F}(X)^*, \cdot) \to (\mathbb{Z}, +) \cup \{\infty\}$ mit folgenden wichtigen Eigenschaften (ohne Beweis; $\varphi, \psi \in \mathbb{F}(X)$):

1. $v_{p(X)}(\varphi) = \infty_{\mathbb{Z}}$ genau dann, wenn $p(X) = 0$,
2. $v_{p(X)}(\varphi \cdot \psi) = v_{p(X)}(\varphi) + v_{p(X)}(\psi)$,
3. $v_{p(X)}(\varphi + \psi) \geq \min\{v_{p(X)}(\varphi), v_{p(X)}(\psi)\}$ bzw.
   $v_{p(X)}(\varphi + \psi) = \min\{v_{p(X)}(\varphi), v_{p(X)}(\psi)\}$, falls $v_{p(X)}(\varphi) \neq v_{p(X)}(\psi)$
4. $v_{p(X)}(p(X)) = 1$ und
5. $v_{p(X)}(c) = 0$ für alle $c \in \mathbb{F}^*$.

Da diese fünf Eigenschaften erfüllt sind, sprechen wir allgemeiner auch von einer *diskreten Bewertung* $v_{p(X)} =: v$ von $F/\mathbb{F}$.

Definition 2.2.2. Ein Ring $\mathcal{O}$, der die beiden Eigenschaften

1. $\mathbb{F} \subsetneq \mathcal{O} \subsetneq F$ und
2. für alle $\varphi \in F$ gilt $\varphi \in \mathcal{O}$ oder $\varphi^{-1} \in \mathcal{O}$

erfüllt, heißt *Bewertungsring* oder *Stellenring* des rationalen Funktionenkörpers $F = \mathbb{F}(X)$.

Die Ringe

$$\begin{aligned} \mathcal{O}_{p(X)} &:= \{\varphi \in F;\ v_{p(X)} \geq 0\} \\ &= \left\{\varphi = \frac{f(X)}{g(X)} \in F;\ f(X), g(X) \in \mathbb{F}[X],\ p(X) \nmid g(X)\right\}, \\ \mathcal{O}_{X'} &:= \{\varphi \in F;\ v_{X'} \geq 0\} \\ &= \left\{\varphi = \frac{f(X)}{g(X)} \in F;\ f(X), g(X) \in \mathbb{F}[X],\ \deg f \leq \deg g\right\}, \end{aligned} \tag{2.5}$$

wobei $p(X)$ ein normiertes Primpolynom aus $\mathbb{F}[X]$ ist, sind demnach Stellenringe, denn es ist jeweils $\mathbb{F}$ als Menge der konstanten Funktionen ($v_{p(X)}(c) = 0 \quad \forall c \in \mathbb{F}$) echt enthalten und die Ringe selbst sind echte Unterringe von F (denn für $\varphi \in F$ mit $v_{p(X)} < 0$ ist $\varphi \notin \mathcal{O}_{p(X)}$, aber $\mathcal{O}_{p(X)} \subset F$). Außerdem ist jedes $\varphi \in F$ eindeutig (bis auf die Reihenfolge der Faktoren) in Primpolynome zerlegbar und damit entweder $\varphi \in \mathcal{O}_{p(X)}$ oder $\varphi^{-1} \in \mathcal{O}_{p(X)}$, und

damit sind beide Forderungen aus Definition 2.2.1 erfüllt. Die zugehörigen Einheitengruppen sind

$$\begin{aligned} \mathcal{O}^*_{p(X)} &:= \{\varphi \in F;\ v_{p(X)}(\varphi) = 0\}, \\ \mathcal{O}^*_{X'} &:= \{\varphi \in F;\ v_{X'}(\varphi) = 0\}. \end{aligned} \tag{2.6}$$

Einige Eigenschaften von Stellenringen fasst der folgende Satz zusammen.

Satz 2.2.1. *Sei $\mathcal{O}$ ein Stellenring des Funktionenkörpers F. Dann gilt:*

1. *$\mathcal{O}$ ist ein kommutativer Ring,*
2. *$\mathcal{O}$ ist ein lokaler Ring, d.h. $\mathcal{O}$ besitzt ein eindeutig bestimmtes Maximalideal, nämlich $P := \mathcal{O} \setminus \mathcal{O}^*$,*
3. *$\mathcal{O}$ ist ein Hauptidealring, in dem P ein Hauptideal ist, d.h. $P = (t) = t \cdot \mathcal{O}$ mit einem erzeugenden Element t,*
4. *Wenn $P = t \cdot \mathcal{O}$ ist, dann ist jedes Element $0 \neq \varphi \in F$ eindeutig darstellbar in der Form $\varphi = t^n \cdot u$, wobei $n \in \mathbb{Z}$ und $u \in \mathcal{O}^*$.*

Beweis. 1. Nach Definition ist $\mathbb{F} \subsetneq \mathcal{O} \subsetneq F$, wobei F und $\mathbb{F}$ Körper, also insbesondere auch kommutative Ringe sind. Dann ist auch der Unterring $\mathcal{O} \subsetneq F$ kommutativ.

2. *Existenz:* Wir zeigen, dass $P := \mathcal{O} \setminus \mathcal{O}^*$ ein Ideal in $\mathcal{O}$ und dass P maximal ist. Es müssen also die Idealeigenschaften geprüft werden. Dazu sei $a \in P$ und $o \in \mathcal{O}$. Dann ist auch $a \cdot o \in \mathcal{O} \setminus \mathcal{O}^* = P$, denn wäre $a \cdot o \in \mathcal{O}^*$, so wäre a eine Einheit. Des Weiteren gilt für zwei beliebige Elemente $a, a' \in P$ auch $a' - a \in P$, denn wir können o.B.d.A. annehmen, dass mit a, a' auch a/a' und $1 - a/a'$ in $\mathcal{O}$ liegen. Mit der ersten Idealeigenschaft ist dann auch $a'(1 - a/a') = a' - a \in \mathcal{O}$ und somit P ein Ideal in $\mathcal{O}$. Für die Maximalität von P müssen wir zeigen, dass für ein weiteres Ideal $I \subset \mathcal{O}$ mit $P \subset I \subset \mathcal{O}$ folgt $I = P$ oder $I = \mathcal{O}$. Wir nehmen an, es gäbe ein Element $i \in I$, das nicht in P enthalten ist, d.h. $i \in \mathcal{O}^*$. Dann enthält I sowohl P (denn $iP \subseteq I$), als auch die ganze Einheitengruppe $\mathcal{O}^*$. Es folgt $I = \mathcal{O}^* \cup P = \mathcal{O}$ und damit P maximal. *Eindeutigkeit:* Angenommen, es gäbe ein zweites Maximalideal $P' \in \mathcal{O}$, $P' \neq P$. Wir unterscheiden zwei Fälle: a) P' enthält eine Einheit $e \in \mathcal{O}^*$. Dann ist $P' = \mathcal{O}$. Dies steht jedoch im Widerspruch zur Maximalität von P. b) P' enthält keine Einheit aus $\mathcal{O}$. Nun ist offenbar $P' \subseteq P = \mathcal{O} \setminus \mathcal{O}^*$ und damit $P = P'$ wegen der Maximalität von P und P'.

3. Wir beweisen die Behauptung hier nur für den Spezialfall $\mathcal{O} = \mathcal{O}_{p(X)}$. Es sei dazu $\mathfrak{a}$ ein beliebiges Ideal aus $\mathcal{O}_{p(X)}$ und $a := \min\{v_{p(X)}(\alpha);\, \alpha \in \mathfrak{a}\} \geq 0$, sowie $\mu \in \mathfrak{a}$ mit $v_{p(X)}(\mu) = a$. Da $v_{p(X)}(0) = \infty_{\mathbb{Z}}$ ist, folgt $\mu \neq 0$. Ist nun $\alpha \in \mathfrak{a}$ beliebig, dann ist $v_{p(X)}(\frac{\alpha}{\mu}) = v_{p(X)}(\alpha) - v_{p(X)}(\mu) \geq 0$. Folglich ist $\frac{\alpha}{\mu} \in \mathcal{O}_{p(X)}$ und somit $\alpha \in \mu \cdot \mathcal{O}_{p(X)} = (\mu)$. Da $\alpha \in \mathfrak{a}$ beliebig gewählt war, gilt $\mathfrak{a} \subseteq (\mu)$. Andererseits ist aber auch $(\mu) \subseteq \mathfrak{a}$ und damit $(\mu) = \mathfrak{a}$. $\mathfrak{a}$ ist also von einem μ erzeugt und damit Hauptideal. Da auch $\mathfrak{a}$ beliebig aus $\mathcal{O}_{p(X)}$ war, ist jedes Ideal Hauptideal und $\mathcal{O}_{p(X)}$ Hauptidealring. Damit ist dann auch das Maximalideal P Hauptideal und wird entsprechend von einem Element t erzeugt, so dass $P = t \cdot \mathcal{O}$.

4. *Existenz:* Nach Definition ist für jedes $\varphi \in F$ entweder $\varphi \in \mathcal{O}$ oder $\varphi^{-1} \in \mathcal{O}$. Wenn $\varphi \in \mathcal{O}^*$, dann ist $\varphi = t^0\varphi$. Ist $\varphi \notin \mathcal{O}^*$, dann bleibt der Fall $\varphi \in P$ zu betrachten. Da P ein Hauptideal ist, besitzt es ein erzeugendes Element t und jedes Element des Hauptideals ist darstellbar als Produkt einer Einheit und einer Potenz des Erzeugenden. Es gibt also ein (maximales) $m \geq 1$, mit $\varphi = t^m u$ und $u \in \mathcal{O}^*$.
   Eindeutigkeit: trivial. □

Definition 2.2.3. Ist $\mathcal{O}$ ein Stellenring mit dem eindeutig bestimmten Maximalideal P, dann heißt P eine *Stelle* des Funktionenkörpers F. Ein erzeugendes Element t des Maximalideals $P = (t) = t \cdot \mathcal{O}$ heißt *lokaler Parameter* in P.

$$\mathbb{P}_F := \{P;\ P \text{ ist Stelle von } F\}.$$

$\mathcal{O}_P := \mathcal{O}$ heißt *Bewertungs-* oder *Stellenring an der Stelle* P.

Beispiel 2.2.1. Im Stellenring $\mathcal{O}_{p(X)}$ ist $p(X)$ ein lokaler Parameter und in $\mathcal{O}_{X'}$ ist dies $X' = 1/X$ (siehe Gleichungen (2.5) und (2.6)).

Wir stellen darüber hinaus fest, dass $K_P := \mathcal{O}_P/P$ ein (Restklassen-) Körper ist, da $\mathcal{O}_P$ ein Ring und P Maximalideal ist. Insbesondere ist jedes Maximalideal auch Primideal.

Lemma 2.2.1. *Sei $p(X) \in \mathbb{F}[X]$ ein lokaler Parameter in der Stelle P. Der Restklassenkörper $K_P := \mathcal{O}_P/P$ ist eine endliche Körpererweiterung vom Grad $[K_P : \mathbb{F}] = \deg p(X)$, falls $P \neq \infty$. Für $P = \infty$ ist $[K_P : \mathbb{F}] = 1$. Insbesondere ist K_P für alle Stellen P ein endlicher Körper mit $\#K_P = (\#\mathbb{F})^{\deg p(X)} = q^{\deg p(X)}$.*

BEWEIS. Zum Beweis des Lemmas betrachten wir hier verkürzend nur das folgende Diagramm:

$$\begin{array}{ccccc} P & \hookrightarrow & \mathcal{O}_P & \overset{\rho_P}{\twoheadrightarrow} & K_P = \mathcal{O}_P/P \\ \uparrow & & & & \uparrow \\ (p(X)) & \hookrightarrow & \mathbb{F}[X] & \twoheadrightarrow & \mathbb{F}[X]/(p(X)) \end{array}.$$

Die vertikalen Abbildungen sind injektiv. Also ist insbesondere $(p(X)) \subseteq P$. Die Abbildung $\mathbb{F}[X]/(p(X)) \to K_P$ ist zusätzlich surjektiv, also insgesamt bijektiv und damit isomorph, da sie linear ist. Daraus folgt die Gleichheit $\mathbb{F}[X]/(p(X)) = K_P$. □

Von besonderer Bedeutung ist die Abbildung

$$\rho_P : \mathcal{O}_P \to K_P,\ \varphi \mapsto \varphi \bmod P.$$

Definition 2.2.4. Es sei ρ_P die Restklassenabbildung, wie oben. Das Bild einer rationalen Funktion φ

$$\varphi(P) := \rho_P(\varphi)$$

in K_P heißt *Funktionswert von φ in der Stelle P.*

Der Funktionswert einer rationalen Funktion in einer Stelle P ist demnach der Rest, den die Funktion bei Division durch das P erzeugende normierte Primpolynom lässt.

Definition 2.2.5. Sei $\varphi(X) = \frac{f(X)}{g(X)} \in F$ weitestgehend gekürzt.

1. Der Grad $[K_P : \mathbb{F}]$ der Körpererweiterung $K_P/\mathbb{F}$ heißt *Grad der Stelle P*, in Zeichen $[K_P : \mathbb{F}] =: \deg P$.

2. Die Stelle P heißt *Nullstelle* von $\varphi(X) = \frac{f(X)}{g(X)}$ genau dann, wenn $\varphi(P) = 0$, wenn also gilt $p(X) \mid f(X)$. $v_P(\varphi(X))$ heißt dann *Nullstellenordnung* von φ in P.

3. Die Stelle P heißt *Polstelle* oder *Pol* von $\varphi(X)$ genau dann, wenn $\frac{1}{\varphi}(P) = 0$, wenn also gilt $p(X) \mid g(X)$. $v_P(\frac{1}{\varphi(X)})$ heißt dann *Polordnung* von φ in P.

Wir wollen nun die Punkte der projektiven Geraden über $\mathbb{F}$ mit den Stellen des rationalen Funktionenkörpers F in Beziehung setzen. Dabei helfen uns

Satz 2.2.2. *Es gibt keine weiteren Stellen in dem rationalen Funktionenkörper F, außer $P_{p(X)} := \mathcal{O}_{p(X)} \setminus \mathcal{O}^*_{p(X)}$ und $P_\infty := \mathcal{O}_\infty \setminus \mathcal{O}^*_\infty$.*

und das

Korollar 2.2.1. *Es gibt eine $1:1$-Zuordnung zwischen $\mathbb{P}_F$ und der Menge der normierten Primpolynome aus $\mathbb{F}[X]$ vereinigt mit $\{X'\} = \{1/X\}$.*

Beweis. (von Satz und Korollar) siehe [Sti93], S. 10. □

Das Korollar lässt uns die Punkte $P = P(a)$ und ∞ der projektiven Geraden $\mathbb{P}^1(\mathbb{F})$ mit den Stellen vom Grad 1 des Funktionenkörpers F identifizieren, also mit den von normierten linearen Primpolynomen erzeugten Maximalidealen in $\mathbb{F}(X)$:

$$\mathbb{P}^1(\mathbb{F}) \ni a \leftrightarrow (X - a) \in \mathbb{F}[X],$$
$$\mathbb{P}^1(\mathbb{F}) \ni \infty \leftrightarrow (X') = (1/X) \in \mathbb{F}[X'].$$

Divisoren

Definition 2.2.6 (und Satz). Wir betrachten wieder den rationalen Funktionenkörper $F = F/\mathbb{F} = \mathbb{F}(X)$ über dem endlichen Körper $\mathbb{F} = \mathbb{F}_q$ und setzen $m_{P_i} := v_{P_i}(\varphi) := v_{p_i(X)}(\varphi)$ für $\varphi \in F$.

1. Ein *Divisor* ist eine formale Summe $D := \sum_{P \in \mathbb{P}_F} m_P P$, mit $m_P \in \mathbb{Z}$ und $m_p = 0$ für fast alle P.

2. Die freie abelsche Gruppe $(Div(F), +)$ (mit koeffizientenweiser Addition der formalen Summen) heißt die *Divisorgruppe* des Funktionenkörpers F.

3. Der *Träger eines Divisors D* ist die Menge der Stellen $P_i \in \mathbb{P}_F$ mit $m_{P_i} \neq 0$, in Zeichen $supp(D) := \{P_i \in \mathbb{P}_F;\ m_{P_i} \neq 0\}$.

4. Ein Divisor D heißt *positiver* oder *effektiver Divisor*, wenn $m_{P_i} \geq 0$ für alle $P_i \in \mathbb{P}_F$.

5. Wir definieren eine Ordnungsrelation (eine Halbordnung) auf $Div(F)$ durch $D \geq 0 :\Leftrightarrow m_{P_i} \geq 0$ für alle $P_i \in \mathbb{P}_F$. Und es ist $D \geq D' \Leftrightarrow D - D' \geq 0$, da $Div(F)$ additive Gruppe.

6. Der Divisor einer rationalen Funktion $\varphi \in F$, definiert durch $(\varphi) := \sum_{P_i \in \mathbb{P}_F} v_{p_i(X)}(\varphi) P_i$ heißt *Hauptdivisor* von $\varphi \in F$. Die Menge der Hauptdivisoren von F ist eine Untergruppe der Divisorgruppe $Div(F)$ und wird mit $Princ(F)$ bezeichnet.

7. Der *Grad eines Divisors* wird durch $\deg D := \sum_{P \in \mathbb{P}_F} (m_P \deg P)$ definiert. Hauptdivisoren haben Grad Null.

BEWEIS. (von 5.,6. und 7.) *5.:* Die Ordnungsrelation ist reflexiv, transitiv und antisymmetrisch. Es ist $D \geq D$ und für $D \geq E$ und $E \geq F$ ist $D \geq F$, sowie $D \geq D' \Rightarrow D' \ngeq D$ falls $D \neq D'$. Also ist $\geq$ eine Halbordnung auf $Div(F)$.
6.: Es genügt, zu zeigen, dass Summe und Differenz zweier beliebiger Hauptdivisoren $\varphi, \psi \in Princ(F)$ in $Div(F)$ enthalten sind. Dies sehen wir mit den Definition 2.2.1 folgenden Eigenschaften der Bewertung sofort ein.
7.: Sei $\varphi = \frac{f(X)}{g(X)} = \frac{p_1^{m_1} \cdot \ldots \cdot p_r^{m_r}}{q_1^{n_1} \cdot \ldots \cdot q_s^{n_s}}$ eine rationale Funktion aus F, wobei $m_i, n_j \in \mathbb{N}$ und $p_i(X), q_j(X) \in \mathbb{F}[X]$ paarweise verschiedene Primpolynome sind. Seien P_i, Q_j die den Primpolynomen zugehörigen Stellen, so ist (φ) von der Form $(\varphi) = \sum_{i=1}^{r} m_i P_i - \sum_{j=1}^{s} n_j Q_j$. Für den Grad gilt damit

$$\begin{aligned}
\deg(\varphi) &= \sum_{i=1}^{r} m_i \deg P_i - \sum_{j=1}^{s} n_j \deg Q_j + v_\infty(\varphi(X)) \\
&= \sum_{i=1}^{r} m_i \deg p_i(X) - \sum_{j=1}^{s} n_j \deg q_j(X) + \deg g(X) - \deg f(X) \\
&= \deg f(X) - \deg g(X) + \deg g(X) - \deg f(X) = 0
\end{aligned}$$

und damit die Behauptung. □

In [van99] lesen wir, dass der Divisor einer rationalen Funktion φ die Rolle eines Buchhalters spielt, der Auskunft über die Nullstellen und Pole von φ mit ihren jeweiligen Ordnungen gibt. Allgemein können wir jeden Divisor in einen Nullstellen- und einen Polstellenanteil zerlegen ($m_{P_i} \in \mathbb{N}$, $P_i \in \mathbb{P}_F$, $P_i \neq P_j$ für $i \neq j$):

$$D = \underbrace{\sum_{i=1}^{r} m_{P_i} P_i}_{Nullstellenanteil} - \underbrace{\sum_{i=r+1}^{n} m_{P_i} P_i}_{Polanteil}.$$

Definition 2.2.7. Zwei Divisoren $E, E' \in Div(F)$ heißen *linear äquivalent* (geschrieben: $E \equiv E'$), wenn es einen Hauptdivisor $(\epsilon) \in Princ(F)$ gibt, so dass gilt

$$E' = E + (\epsilon).$$

Funktionenräume

Definition 2.2.8. Es sei D ein Divisor. Die Menge der rationalen Funktionen

$$\mathcal{L}(D) := \{\varphi \in F;\ (\varphi) + D \geq 0\} \cup \{0\}$$

heißt der durch D bestimmte *Funktionenraum*.

Ein so definierter Funktionenraum besitzt einige wichtige Eigenschaften, die wir in einem Lemma zusammenfassen.

Lemma 2.2.2. *Es sei $\mathcal{L}(D)$ ein durch D bestimmter Funktionenraum.*

1. *$\mathcal{L}(D)$ ist ein Vektorraum über dem Körper $\mathbb{F}$.*
2. *Für seine Elemente $\varphi \in \mathcal{L}(D)$ gilt für ein $r \in \mathbb{N}_0$:*
   (a) *φ hat Polstellen in $P_1, \ldots, P_r$ der Ordnung $\leq m_{P_i}$ für $1 \leq i \leq r$,*
   (b) *φ hat Nullstellen in $P_{r+1}, \ldots, P_n$ der Ordnung $\geq m_{P_i}$ für $r+1 \leq i \leq n$.*

Beweis. 1. Mit $a \in \mathbb{F}^*$ und $\varphi, \psi \in \mathcal{L}(D)$ folgt die Abgeschlossenheit bzgl. der Multiplikation mit einem Skalar:

$$\mathcal{L}(D) \ni (a \cdot \varphi) = (a) + (\varphi) = 0 + (\varphi) = (\varphi) \in \mathcal{L}(D).$$

Die Abgeschlossenheit bzgl. der Addition zweier Elemente erhalten wir mit der Dreiecksungleichung für Bewertungen, da mit $(\varphi), (\psi) \geq -D$ auch $(\varphi + \psi) \geq -D$ ist.

2. folgt sofort mit einer Zerlegung des Divisors D, wie oben, in $D = \sum_{i=1}^{r} m_{P_i} P_i - \sum_{i=r+1}^{n} m_{P_i} P_i$ mit $m_{P_i} \in \mathbb{N}$ und der Definition von $(\varphi) \geq D$. □

Wir wollen hier nur ein konkretes Beispiel anführen, das an späterer Stelle interessant sein wird.

Beispiel 2.2.2. Es sei $G = (k-1) \cdot \infty$ ein Divisor. Der zugehörige Funktionenraum über F ist dann

$$\begin{aligned}
\mathcal{L}(G) &= \{\varphi \in F;\ (\varphi) + G \geq 0\} \\
&= \{\varphi \in F;\ v_P(\varphi) \geq 0 \wedge v_\infty(\varphi) \geq -(k-1)\ \forall P \neq \infty\} \\
&= \{\varphi = \frac{f(X)}{g(X)} \in F;\ -\deg f(X) + \deg g(X) \geq -(k-1)\} \\
&\overset{(*)}{=} \{f(X) \in \mathbb{F}[X];\ \deg f(X) < k\}
\end{aligned}$$

also die Menge der Funktionen mit Koeffizienten aus $\mathbb{F}$ vom Grad kleiner als k. Dabei gilt $(*)$, da $g(X) = const \in \mathbb{F}^*$, denn sonst hätte $g(X)$ einen Pol im endlichen und es wäre $v_P(1/g(X)) = -v_P(g(X)) < 0$ für eine Stelle P im Widerspruch zur Voraussetzung.

Lemma 2.2.3. *Linear äquivalente Divisoren liefern isomorphe Funktionenräume.*

BEWEIS. Es seien E, E' zwei Divisoren, für die gilt $E' = (\epsilon) + E$, $(\epsilon) \in Princ(F)$ und $\mathcal{L}(E)$, $\mathcal{L}(E')$ die zugehörigen Funktionenräume. Die Abbildung $\mathcal{L}(E) \to \mathcal{L}(E')$, $\varphi \mapsto \varphi\epsilon$ ist ein Isomorphismus, denn $\varphi \in \mathcal{L}(E') \Leftrightarrow (\varphi) \geq -E - (\epsilon) \Leftrightarrow (\varphi) + (\epsilon) \geq -E \Leftrightarrow (\varphi \cdot \epsilon) \geq -E \Leftrightarrow \varphi\epsilon \in \mathcal{L}(E)$. □

Wir wollen nun die Dimension des $\mathbb{F}$-Vektorraums $\mathcal{L}(D)$ untersuchen, die wir mit $l(D) := \dim_{\mathbb{F}} \mathcal{L}(D)$ bezeichnen.

Satz 2.2.3. *Es sei $\mathcal{L}(D)$ der Funktionenraum zum Divisor D, dann gilt*

1. *$l(D) < \infty$ für jeden Divisor $D \in Div(F)$,*
2. *$l(D) = 0$, falls $\deg D < 0$ und*
3. *$l(D) = 1 + \deg D$, falls $\deg D \geq 0$.*

BEWEIS.

1. Die Körpererweiterungen $K_P/\mathbb{F}$ sind allesamt endlichen Grades, also sind auch alle Stellen P von endlichem Grad. Damit ist auch der Grad eines jeden Divisors (als endliche Summe) endlich. Die Behauptung folgt dann aus 3. und 2.

2. Wir können zeigen, dass $D \equiv (\deg D) \cdot \infty$. Dann ist auch $\mathcal{L}(D) \cong \mathcal{L}((\deg D) \cdot \infty)$ und folglich $l(D) = l((\deg D) \cdot \infty)$. Es gilt außerdem $\mathcal{L}((r-1) \cdot \infty) = \mathbb{F}[X]_{\deg < r}$, also $l((r-1) \cdot \infty) = r$. Ist nun $\deg D < 0$, so ist $l(D) = l((\deg D) \cdot \infty) = 0$.

3. Für $\deg D \geq 0$ erhalten wir mit den gleichen Überlegungen, wie in 2. schließlich $l(D) = l((\deg D) \cdot \infty) = 1 + \deg D$. □

Differentialformen

Wir betrachten wieder nur den Fall des rationalen Funktionenkörpers F.

Definition 2.2.9. Es sei F der rationale Funktionenkörper über $\mathbb{F}$.

1. Die *formale Ableitung eines Polynoms* $f(X) = \sum_{i=0}^{n} a_i X^i \in \mathbb{F}[X]$ ist definiert durch $f'(X) = \sum_{i=1}^{n} i a_i X^{i-1}$.

2. Der Raum $\Omega := \Omega_F := F \cdot dX$ heißt *Raum der rationalen Differentialformen*. Dabei ist dX zunächst nur ein Symbol, so dass Ω_F ein eindimensionaler Vektorraum über F mit Vektorraumbasis dX ist (der frei erzeugte F-Modul, denn jeder Vektorraum über einem Körper F ist ein F-Modul).

3. Die Abbildung
$$d: F \to \Omega,\ \varphi \mapsto d\varphi := \varphi' \cdot dX$$
heißt *Differentialabbildung*, wobei für $\varphi = \frac{f(X)}{g(X)} \in F$ mit $g(X) \neq 0$ definiert ist $\varphi' := \frac{f'(X)g(X)-f(X)g'(X)}{g(X)^2}$.

Für die Differentialabbildung gelten die üblichen Ableitungsregeln (Produktregel, $\mathbb{F}$-Linearität). Für zwei Differentialformen $\omega := f_1 \cdot d\varphi$ und $\eta := g_1 \cdot d\varphi$ wird eine Fortsetzung von $\frac{df}{dg} := \frac{f'}{g'}$ definiert durch $\frac{\omega}{\eta} := \frac{f_1}{g_1}$, so dass auch die Kettenregel $\frac{df}{dg} \cdot \frac{dg}{dt} = \frac{df}{dt}$ gilt.
Für die Bewertung einer Differentialform dX definieren wir

$$v_P(dX) := \begin{cases} 0, & P \neq \infty \\ -2, & P = \infty \end{cases}, \tag{2.7}$$

wegen $dX_\infty = X'_\infty dX = -1/X^2 dX = -X^{-2} dX$.
Damit können wir nun beliebige Differentialformen bewerten, was wir im folgenden Beispiel zeigen wollen.

Beispiel 2.2.3. Es seien $f, g \in F$ und $\omega = g \cdot dt \in \Omega_F$.

- $v_P(g \cdot dt) = v_P(gt' \cdot dX) = v_P(g) + v_P(t') + v_P(dX) = v_P(g) + v_P(t' \cdot dX) = v_P(g) + v_P(dt)$
- $v_P(f \cdot \omega) = v_P(fg \cdot dt) = v_P(f) + v_P(g) + v_P(dt) = v_P(f) + v_P(g \cdot dt) = v_P(f) + v_P(\omega)$

Definition 2.2.10. Es sei $\omega \in \Omega$ eine Differentialform.

1. Gilt $v_P(\omega) \geq 0$, so heißt ω *regulär in der Stelle* P.
2. Ist $\omega \neq 0$, dann heißt der Divisor $(\omega) := \sum_{P \in \mathbb{P}_F} v_P(\omega) \cdot P$ *kanonischer Divisor* von ω.

Ist ω regulär in der Stelle P und ist $v_P(\omega) > 0$, dann hat ω in P eine Nullstelle der Ordnung $v_P(\omega)$. Ist $v_P(\omega) < 0$, ω also nicht regulär, so hat ω in P eine Polstelle mit Ordnung $-v_P(\omega)$. Dies wird hier analog zu Definition 2.2.5 definiert.

Es ist unmittelbar einsichtig, dass kanonische Divisoren Divisoren im Sinne von Definition 2.2.6 sind. Denn ist etwa $\omega = g \cdot dX$, $g \in F$, eine beliebige Differentialform aus Ω_F mit zugehörigem kanonischen Divisor $(\omega) = (g) + (dX)$, dann ist $v_P(\omega) = v_P(g) + v_P(dX) = 0$ für fast alle Stellen $P \in \mathbb{P}_F$, gemäß Gleichung (2.7).
Es ist weiterhin offensichtlich

$$\begin{aligned} \deg(\omega) &= \deg((g) + (dX)) = \deg(g) + \deg(dX) \\ &= 0 + (0 + \ldots + 0 + v_\infty(dX)) \\ &= -2. \end{aligned} \tag{2.8}$$

Bemerkung 2.2.1. Sind $\omega, \eta \neq 0$ zwei Differentialformen aus Ω_F, dann finden wir eine rationale Funktion $f \in F$, so dass $\omega = f \cdot \eta$. Folglich sind die zugehörigen kanonischen Divisoren linear äquivalent gemäß Definition 2.2.3 und bilden eine Äquivalenzklasse, die wir mit $W := (\omega)$ bezeichnen wollen.

Beweis. Nach Definition 2.2.9 ist $\Omega_F = F \cdot dX = F \cdot \eta = F \cdot f \cdot \eta = F \cdot \omega$ für ein $f \in F$ und es gilt $(\omega) = (f \cdot \eta) = (f) + (\eta) \Rightarrow (\omega) \equiv (\eta)$, denn f ist rational. □

Definition 2.2.11. Es sei D ein Divisor auf $\mathbb{P}_F$. Wir definieren einen Raum

$$\Omega(D) := \{\omega \in \Omega_F;\ (\omega) - D \geq 0\}$$

und nennen seine Dimension $\dim_{\mathbb{F}} \Omega(D) =: \delta(D)$ *index of speciality.*

Satz 2.2.4. *Es gilt* $\delta(D) = l(W - D)$.

Beweis. Es genügt, zu zeigen, dass $\Omega(D)$ isomorph ist zu $\mathcal{L}(W - D)$, denn isomorphe Räume haben die gleiche Dimension. Es sei $W = (\omega)$. Wir betrachten die lineare Abbildung

$$\phi :\ \Omega(D) \to \mathcal{L}(W - D),\ f \mapsto \phi(f) := f \cdot \omega \tag{2.9}$$

und sehen sofort ein, dass diese Abbildung bijektiv ist, also ist ϕ ein Isomorphismus und die Behauptung gezeigt. □

Daraus folgt unmittelbar, dass $\delta(D) < \infty$, denn $l(G)$ ist für alle Divisoren G endlich nach Satz 2.2.3.

Definition 2.2.12. Unter einer *logarithmischen Differentialform* verstehen wir eine Abbildung

$$\lambda :\ F^* \to \Omega, \quad f \mapsto \lambda(f) := \frac{1}{f} \cdot df.$$

Sie hat folgende wichtige Eigenschaften:

- $\lambda(f) = \frac{1}{f} \cdot df = \frac{1}{f} \cdot \frac{df}{dt} \cdot dt = \frac{f'}{f} \cdot dt$. Daher erklärt sich auch der Name „logarithmisch“, denn $(\ln(f))' = f'/f$.
- $\lambda(f \cdot g) = \frac{1}{fg} \cdot d(fg) = \frac{1}{fg} \cdot (g \cdot df + f \cdot dg) = \frac{1}{f} \cdot df + \frac{1}{g} \cdot dg = \lambda(f) + \lambda(g)$. Man sagt, die logarithmische Differentialform sei additiv.

2.2.3 Der Satz von Riemann-Roch

Wir benötigen diesen Satz, um später Aussagen über den Typ residueller Goppa-Codes machen zu können.

Satz 2.2.5 (Riemann-Roch für das Geschlecht $g = 0$). *Ist D ein Divisor aus $Div(F)$ und W ein kanonischer Divisor, dann gilt*

$$l(D) - l(W - D) = \deg D + 1.$$

Beweis. Wir unterscheiden drei Fälle:

1. $\deg D \geq 0$: Es genügt, zu zeigen, dass $l(W - D) = 0$. Dazu zeigen wir, dass $\deg(W - D) < 0$. Es ist

$$\deg(W - D) = \deg W - \deg D \overset{(2.8)}{=} -2 - \deg D < 0. \tag{2.10}$$

   Die Behauptung folgt dann mit Satz 2.2.3.

2. $\deg D = -1$: Nach Satz 2.2.3 ist in diesem Fall $l(D) = 0$. Mit Gleichung (2.10) erhalten wir $\deg(W-D) = -1$, also auch $l(W-D) = 0$. Folglich gilt die Behauptung auch in diesem Fall.

3. $\deg D \leq -2$: Es ist wieder $l(D) = 0$. Mit Gleichung (2.10) ist $l(W - D) = (-2 - \deg D) + 1$. Insgesamt haben wir damit

$$l(D) - l(W - D) = 0 - (-\deg D - 2 + 1) = \deg D + 1,$$

   was zu beweisen war. □

2.2.4 Der Residuensatz

Der Residuensatz wird uns im nächsten Abschnitt helfen, eine Kontrollmöglichkeit der dort definierten rational-geometrischen Goppa-Codes zu finden. Bevor wir jedoch zum Satz kommen, müssen wir erst noch erklären, was wir unter einem Residuum verstehen wollen.

Definition 2.2.13 (und Satz). Es seien F der rationale Funktionenkörper über $\mathbb{F}$ und $P \in \mathbb{P}_F$ eine Stelle vom Grad 1 mit lokalem Parameter $t = t(P)$.

1. Es seien $f \in F$, $c_i \in \mathbb{F}$, $m \in \mathbb{Z}$ und $n \in \mathbb{N}$, dann gibt es eine eindeutige Zerlegung

$$f = c_m t^m + c_{m+1} t^{m+1} + \ldots + c_{n-1} t^{n-1} + f_n, \tag{2.11}$$

   wobei gilt $t^n \mid f_n$. Diese Zerlegung heißt *Laurentreihenentwicklung von f in P bzgl. des lokalen Parameters t im Abschnitt $[m, n)$.*

2. Der (-1)-te Koeffizient einer solchen Entwicklung heißt das *Residuum der rationalen Funktion f in der Stelle P* und wird bezeichnet mit

$$res_t(f) := c_{-1}.$$

3. Für eine Differentialform $\Omega \ni \omega = f \cdot dt$ definieren wir das Residuum von ω in P durch

$$res_P(\omega) := res_t(f).$$

   Es ist unabhängig von der Wahl des lokalen Parameters der Reihenentwicklung in P.

4. Die Menge $\mathbb{F}((t)) := \{\sum_{i=m}^{\infty} c_m t^m;\ m \in \mathbb{Z},\ c_i \in \mathbb{F}\}$ ist ein Körper, der *Körper der formalen Laurentreihen.*

Satz 2.2.6 (Residuensatz). *Es sei ω eine beliebige Differentialform aus $\Omega_F = F \cdot dX$, dann gilt*

$$\sum_{P \in \mathbb{P}_F} res_P(\omega) = 0.$$

Wir wollen diesen wichtigen Satz, wie auch den vorigen, hier nicht beweisen, sondern verweisen auf die entsprechende Literatur zur Funktionentheorie.

Lemma 2.2.4. *Es seien P eine Stelle vom Grad 1 aus $\mathbb{P}_F$, f eine rationale Funktion in $\mathcal{O}_P$ und $\omega \in \Omega_F$ eine Differentialform mit $v_P(\omega) \geq -1$, dann gilt*

$$res_P(f \cdot \omega) = f(P) \cdot res_P(\omega).$$

BEWEIS. Wir betrachten die Laurententwicklung von ω in der Stelle P bzgl. eines zu P gehörigen lokalen Parameters t. Da nach Voraussetzung $v_P(\omega) \geq -1$ verschwinden alle Koeffizienten c_i der Entwicklung für $i < -1$. Die Entwicklung von ω ist also von der Form

$$\omega = (c_{-1}t^{-1} + c_0t^0 + c_1t + \ldots)dt. \tag{2.12}$$

Entwickeln wir nun auch f in einer Laurentreihe in P bzgl. t, so erhalten wir

$$f = a_0t^0 + a_1t + a_2t^2 + \ldots, \tag{2.13}$$

denn nach Voraussetzung ist $f \in \mathcal{O}_P$, also $v_P(f) \geq 0$ und damit alle Koeffizienten $a_j = 0$ für $j < 0$. Mit den beiden Gleichungen (2.12) und (2.13) erhalten wir

$$\begin{aligned} res_P(f \cdot \omega) &= res_t(a_0c_{-1}t^{-1} + (a_0c_0 + a_1c_{-1})t^0 + \ldots) \\ &= a_0c_{-1} \\ &= (f \bmod P) \cdot res_t(c_{-1}t^{-1} + c_0t^0 + c_1t + \ldots) \\ &= f(P) \cdot res_P(\omega), \end{aligned}$$

was zu beweisen war. □

Wir benötigen später noch ein weiteres

Lemma 2.2.5. *Es gibt eine logarithmische Differentialform $\eta \in \Omega_F$, so dass für die Stellen P_i, $i = 1, \ldots, n$ aus $supp(D)$ gilt:*

$$res_{P_i}(\eta) = 1 \text{ und } v_{P_i}(\eta) = -1.$$

BEWEIS. Es seien $X - a_j$ die mit P_j identifizierten normierten Primpolynome vom Grad 1, für $j = 1, \ldots, n$ und es sei $h(X) := \prod_{j=1}^{n}(X - a_j)$ das Produkt dieser linearen Polynome. Dann gilt für das Residuum der logarithmischen

Differentialform $\eta := \lambda(h)$ in der Stelle P_i

$$\begin{aligned} res_{P_i}(\eta) = res_{P_i}(\lambda(h)) &= res_{P_i}(\lambda(\prod_{j=1}^{n}(X-a_j))) \\ &= res_{P_i}(\sum_{j=1}^{n}(\lambda(X-a_j))) = res_{P_i}(\sum_{j=1}^{n}(\frac{1}{X-a_j}\cdot d(X-a_j))) \\ &= res_{P_i}(\sum_{j=1}^{n} 1\cdot(X-a_j)^{-1}) = \delta_{ij} = 1 \end{aligned}$$

und damit die erste Behauptung. Außerdem gilt auch

$$\begin{aligned} v_{P_i}(\eta) = v_{P_i}(\lambda(h)) &= v_{P_i}(\lambda(\prod_{j=1}^{n}(X-a_j))) \\ &= v_{P_i}(\sum_{j=1}^{n}(\lambda(X-a_j))) = v_{P_i}(\sum_{j=1}^{n}(\frac{1}{X-a_j}\cdot d(X-a_j))) \\ &= -1\cdot\delta_{ij} + 0 = -1, \end{aligned}$$

also der zweite Teil der Behauptung. □

2.3 Geometrische Goppa-Codes

In diesem Abschnitt wollen wir zeigen, wie die vorbereiteten Erkenntnisse über Stellen, Divisoren, Funktionenräume und Differentialformen für die Codierungstheorie nutzbar sind.
Im weiteren gelten folgende Bezeichnungen und Bedingungen:

- F ist der rationale Funktionenkörper über dem endlichen Körper $\mathbb{F} = \mathbb{F}_q$,
- $P_1, \ldots, P_n$ sind paarweise verschiedene Stellen aus $\mathbb{P}_F$ vom Grad 1,
- $D := P_1 + \ldots + P_n$ ist ein Divisor aus $Div(F)$,
- G ist ein Divisor aus $Div(F)$, so dass $supp(D) \cap supp(G) = \emptyset$.

2.3.1 Rational-geometrische Goppa-Codes

Definition 2.3.1. Der zu den beiden Divisoren D, G gehörige Code

$$C_{\mathcal{L}}(D,G) := ev_D(\mathcal{L}(G)) := \{\underbrace{(\varphi(P_1),\ldots,\varphi(P_n))}_{:=ev_D(\varphi)};\ \varphi \in \mathcal{L}(G)\} \subseteq \mathbb{F}^n$$

heißt *rational-geometrischer Goppa-Code.*

Wir werden nun einige Aussagen über den Typ eines solchen Codes machen.

Satz 2.3.1. *Es sei $C_{\mathcal{L}}(D,G)$ ein rational-geometrischer Goppa-Code des Typs (n,k,d).*

1. *Die Länge n des Codes ist durch den Divisor D bestimmt.*

2. *Für die Dimension k des Codes $C_{\mathcal{L}}(D,G)$ gilt allgemein:*

$$k = l(G) - l(G-D)$$

   und im speziellen unter der zusätzlichen Voraussetzung, dass G ein effektiver Divisor ist

$$k = \begin{cases} 1+\deg G, & \text{für } \deg G < \deg D; \\ \deg D, & \text{für } \deg G \geq \deg D. \end{cases}$$

3. *Für den Minimalabstand d gilt $d \geq \delta := n - \deg G$. Falls $\deg G < \deg D$ ist, gilt sogar $d = n-k+1$, also $C_{\mathcal{L}}(D,G)$ optimal.*

Beweis. Zum Beweis des Satzes stellen wir zunächst fest, dass alle Funktionen φ aus $\mathcal{L}(G)$ keinen Pol in den Stellen $P_1,\ldots,P_n$ haben, denn wegen $supp(D) \cap supp(G) = \emptyset$ ist $v_{P_i}(\varphi) \geq -v_{P_i}(G) = 0$. (Der Divisor G wird daher auch als *Polschrankendivisor* bezeichnet.)

1. Damit ist offenbar $\varphi \in \mathcal{O}_{P_i}$ und $\varphi(P_i) \in \mathbb{F}$, denn P_i ist Stelle vom Grad 1. Es folgt nun sofort, dass der aus den Tupeln $(\varphi(P_1),\ldots,\varphi(P_n))$ bestehende Code die durch den Divisor D bestimmte Länge n hat.

2. Wir betrachten zum Beweis der allgemeinen Gleichung für k die Abbildung ev_D von $\mathcal{L}(G)$ in $C_{\mathcal{L}}(D,G)$. Der Kern dieser Abbildung ist der Funktionenraum $\mathcal{L}(G-D) = \{f \in F;\ (f) \geq -G+D = -(G-D)\}$, denn für alle $f \in \mathcal{L}(G-D)$ ist $ev_D(f) = f \bmod P_i = 0$ für die $P_i \in supp(D)$, da $v_{P_i}(f) \geq 1$ in $\mathcal{L}(G-D)$. Den Rest erledigt die Dimensionsformel:

$$\begin{aligned} k = \dim_{\mathbb{F}} C_{\mathcal{L}}(D,G) &= \dim \mathcal{L}(G) - \dim \ker(ev_D(\mathcal{L}(G))) \\ &= l(G) - l(G-D). \end{aligned}$$

   Für die Fallunterscheidung im speziellen erinnern wir uns an Satz 2.2.3. Mit der allgemeinen Gleichung von oben und $G \geq 0$ erhalten wir die Behauptung durch einfaches Ausrechnen.

3. Es sei $c = ev_D(\varphi)$ ein Codewort aus $C_{\mathcal{L}}(D,G)$ mit Hamming-Gewicht d. Dann gibt es nach Definition gerade $n-d$ Stellen vom Grad 1, etwa $P_{i_1},\ldots,P_{i_{n-d}}$, so dass $\varphi(P_i)=0$ für diese Stellen und für alle übrigen aus $supp(D)$ ungleich Null ist. Es ist φ also Element eines Funktionenraums $\mathcal{L}(G-E)\neq\{0\}$, wobei $E := P_{i_1}+\ldots+P_{i_{n-d}}$. Es ist $l(G-E)\geq 1$, woraus mit Satz 2.2.3 folgt: $0\leq \deg(G-E) = \deg G - \deg E = \deg G-(n-d)$ und somit

$$d \geq n - \deg G = \delta.$$

Die Optimalität von $C_{\mathcal{L}}(D,G)$ im Fall $\deg G < \deg D$ folgt mit der Singleton-Schranke (Gleichung (1.6)) und $d\geq n-\deg G = n-(k-1) = n-k+1$. □

Wir wollen im weiteren nur noch den optimalen Fall $0\leq \deg G < \deg D$ betrachten, d.h. es ist $k = 1+\deg G$.
Wie sieht dann eine Generatormatrix von $C_{\mathcal{L}}(D,G)$ aus? Wir erinnern uns, dass die Zeilen einer Generatormatrix gerade die Basisvektoren des Codes sind. Ist $\{f_0,f_1,\ldots,f_{\deg G}\}$ ein linear unabhängiges Erzeugendensystem von $1+\deg G$ Funktionen aus $\mathcal{L}(G)$, dann ist die Matrix

$$G_{C_{\mathcal{L}}(D,G)} = \begin{pmatrix} f_0(P_1) & f_0(P_2) & \cdots & f_0(P_n) \\ f_1(P_1) & f_1(P_2) & \cdots & f_1(P_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_{\deg G}(P_1) & f_{\deg G}(P_2) & \cdots & f_{\deg G}(P_n) \end{pmatrix}$$

eine Generatormatrix unseres rational-geometrischen Goppa-Codes.

2.3.2 Goppa-Residuen-Codes

Definition 2.3.2. Es seien D,G Divisoren, die die oben angegebenen Bedingungen erfüllen. Der Code

$$C_\Omega(D,G) := \{\underbrace{(res_{P_1}(\omega),\ldots,res_{P_n}(\omega))}_{=:Res_D(\omega)};\ \omega\in\Omega(G-D)\}\subseteq\mathbb{F}^n$$

heißt *Goppa-Residuen-Code* oder *residueller Goppa-Code*.

Korollar 2.3.1. *Es seien D,G zwei Divisoren, wie oben. Der residuelle Goppa-Code $C_\Omega(D,G)$ und der rational-geometrische Goppa-Code $C_{\mathcal{L}}(D,G)$ sind dual zueinander:*

$$C_\Omega(D,G)^\perp = C_{\mathcal{L}}(D,G).$$

BEWEIS. Es ist zu zeigen, dass jedes Codewort $\varphi = \varphi(D) := ev_D(\varphi) \in C_{\mathcal{L}}(D,G)$ senkrecht zu den Codewörtern von $C_\Omega(D,G)$ ist. Mit Definition 1.2.2 soll also gelten

$$\langle \varphi(D), Res_D(\omega) \rangle = 0.$$

Wir berechnen das Produkt:

$$\begin{aligned}
\langle \varphi(D), Res_D(\omega) \rangle &= \langle (\varphi(P_1), \ldots, \varphi(P_n)), (res_{P_1}(\omega), \ldots, res_{P_n}(\omega)) \rangle \\
&= \sum_{i=1}^{n} \varphi(P_i) \cdot res_{P_i}(\omega) \\
&= \sum_{i=1}^{n} res_{P_i}(\varphi \cdot \omega) \qquad \text{(wegen Lemma 2.2.4)} \\
&\overset{(*)}{=} 0 \qquad \text{(wegen des Residuensatzes 2.2.6, siehe unten).}
\end{aligned}$$

Wir zeigen, dass $(*)$ gilt. Die Summe der Residuen der Stellen vom Grad 1 zerfällt in zwei Summen:

$$0 = \sum_{P \in \mathbb{P}_F^1} res(\varphi \cdot \omega) = \sum_{P \in supp(D)} res(\varphi \cdot \omega) + \sum_{P \in \mathbb{P}_F^1 \setminus supp(D)} res(\varphi \cdot \omega)$$

Es genügt zu zeigen, dass sie zweite Summe verschwindet, denn dann muss wegen des Residuensatzes auch die erste Summe verschwinden. Nach Voraussetzung war $\varphi \in \mathcal{L}(G)$, also $(\varphi) \geq -G$. Für ω galt $\omega \in \Omega(G - D)$, also $(\omega) \geq G - D$. Daraus folgt nun, dass $(\varphi \cdot \omega) \geq -G + G - D = -D$ ist. Damit ist

$$v_P(\varphi \cdot \omega) \begin{cases} \geq -1 & \text{falls } P \in supp(D), \\ \geq 0 & \text{sonst.} \end{cases}$$

Folglich sind die Residuen im Fall $P \in \mathbb{P}_F^1 \setminus supp(D)$ allesamt 0 und damit verschwindet die ganze Summe. Also muss die erste Summe 0 sein und die Codes sind tatsächlich dual. □

Folglich ist eine Generatormatrix von $C_\Omega(D,G)$ gleichzeitig auch Kontrollmatrix von $C_{\mathcal{L}}(D,G)$. Wir wollen wieder den Typ der eben definierten Goppa-Residuen-Codes bestimmen.

Satz 2.3.2. *Es sei $C_\Omega(D,G)$ ein durch die Divisoren D, G, für die gelten soll $-2 < \deg G < \deg D$ bestimmter Goppa-Residuen-Code. Dann gilt für seine Parameter (n, k^*, d^*):*

1. *n ist wieder durch D bestimmt,*
2. *$k^* = n - \deg G - 1$*

3. $d^* \geq \deg G + 2$*, also ist* $d^* \geq n - k^* + 1$ *und* $C_\Omega(D, G)$ *optimal.*

BEWEIS. Wir stellen zunächst fest, dass die Definition sinnvoll ist, da die Residuen der Laurentreihenentwicklungen jeweils aus $\mathbb{F}$ sind.

1. Folglich ist die Länge des Codes n durch die Anzahl der Stellen in $supp(D)$ festgelegt und es ist schließlich $n = \deg D$.

2. Nach Satz 2.2.4 gilt

$$k^* = \dim_{\mathbb{F}} \Omega(G - D) = \delta(G - D) = l(W - (G - D))$$

für einen kanonischen Divisor W. Mit dem Satz von Riemann-Roch ist

$$l(W - (G - D)) = l(G - D) - \deg(G - D) - 1.$$

Da aber $\deg(G - D) < 0$ nach Voraussetzung, folgt $l(G - D) = 0$ und mit $\deg(G - D) = \deg G - \deg D$ haben wir die Behauptung bewiesen. Es ist dann

$$\delta(G - D) = n - \deg G - 1.$$

3. Für die Abschätzung des Minimalabstands d^* verfahren wir analog zum Beweis der Abschätzung von d für rational-geometrische Goppa-Codes. Daher stellen wir hier abkürzend nur noch fest: Es gibt ein $\omega \in \Omega(G - D + E) \neq \{0\}$ mit E wie oben. Nach Satz 2.2.4 ist in diesem Fall $l(W - G + D - E) \geq 1$ und mit Satz 2.2.3 und Gleichung (2.8) gilt nun

$$\begin{aligned} 0 &\leq \deg(W - G + D - E) \\ &= \deg W - \deg G + \deg D - \deg E \\ &= -2 - \deg G + n - (n - d^*), \end{aligned}$$

also $d^* \geq \deg G + 2$, wie behauptet. Die Optimalität folgt ebenfalls mit der Singleton-Schranke. □

Wie es nicht anders zu erwarten ist, zeigt sich, dass für die Summe der Dimensionen der beiden zueinander dualen Codes $C_\Omega(D, G)$ und $C_\mathcal{L}(D, G)$ mit $\deg G < \deg D$ gilt

$$k + k^* = (1 + \deg G) + (n - \deg G - 1) = n.$$

Wir wollen nun überlegen, wie wir eine Generatormatrix für einen solchen Residuen-Code $C_\Omega(D, G)$ finden können. Dabei hilft uns die in Korollar 2.3.1

festgestellte Orthogonalität von $C_{\mathcal{L}}(D,G)$ und $C_{\Omega}(D,G)$. Wir kennen bereits die Generatormatrix von $C_{\mathcal{L}}(D,G)$ aus dem vorigen Abschnitt. Diese ist gleichzeitig Kontrollmatrix von $C_{\Omega}(D,G)$. Auf der anderen Seite ist aber auch jede Kontrollmatrix von $C_{\mathcal{L}}(D,G)$ Generatormatrix von $C_{\Omega}(D,G)$. Das Problem reduziert sich daher darauf, eine Kontrollmatrix von $C_{\mathcal{L}}(D,G)$ zu finden.

Bemerkung 2.3.1. Es sei $C_{\mathcal{L}}(D,G)$ ein rational-geometrischer Goppa-Code des Typs (n,k,d) und $\mathcal{G}$ eine Generatormatrix in Standardform, so dass sich durch Spaltenpermutationen mittels einer $n \times n$-Matrix $\mathcal{P}$ eine Form $\mathcal{G}' = \mathcal{P} \cdot \mathcal{G} = (I_k\, A)$ erzeugen lässt, wobei wir mit I_k die $k \times k$-Einheitsmatrix bezeichnen wollen. Dann ist die Matrix

$$\mathcal{H}' := \mathcal{H} \cdot \mathcal{P}^T := \begin{pmatrix} -A \\ I_{n-k} \end{pmatrix}$$

eine Permutation der Kontrollmatrix $\mathcal{H}$ des Codes $C_{\mathcal{L}}(D,G)$.

Beweis. $\mathcal{G}$ kann mit dem Gauss-Algorithmus soweit transformiert werden, dass mit Spaltenvertauschungen eine Standardform hergestellt werden kann. Die entstandene Matrix ist Generatormatrix eines zu $C_{\mathcal{L}}(D,G)$ äquivalenten Codes gleichen Typs. Wir haben lediglich die Information von der Redundanz getrennt. Die Permutationsmatrizen $\mathcal{P}$ und $\mathcal{P}^T$ sind zueinander invers; ihr Produkt ist eine Einheitsmatrix. Es gilt daher

$$\mathcal{H}' \cdot \mathcal{G}' = \mathcal{H} \cdot \mathcal{P}^T \cdot \mathcal{P} \cdot \mathcal{G} = \mathcal{H} \cdot \mathcal{G} = -A + A = 0$$

und damit die geforderte Orthogonalität. □

Die transponierte Matrix $\mathcal{H}'^T$ ist also eine Generatormatrix des zu $C_{\mathcal{L}}(D,G)$ dualen Codes $C_{\Omega}(D,G)$.

Auf den ersten Blick scheinen Goppa-Residuen-Codes sehr verschieden von rational-geometrischen Goppa-Codes. Der nächste Satz lässt sie uns einfach nur als residuelle Darstellung rational-geometrischer Goppa-Codes verstehen.

Satz 2.3.3. *([Sti93], S. 48) Es seien $D = P_1 + \ldots + P_n$ und G zwei Divisoren mit $supp(D) \cap supp(G) = \emptyset$ und $C_{\Omega}(D,G)$, $C_{\mathcal{L}}(D,G)$ die zugehörigen Goppa-Codes. Ist nun $\eta \in \Omega$ eine logarithmische Differentialform, für die gilt $v_{P_i}(\eta) = -1$ und $res_{P_i}(\eta) = 1$ für $i = 1, \ldots, n$, dann ist*

$$C_{\mathcal{L}}(D,G)^{\perp} = C_{\Omega}(D,G) = C_{\mathcal{L}}(D,H),$$

wobei $H = D - G + (\eta)$.

BEWEIS. Zunächst stellen wir fest, dass wegen der Voraussetzungen $v_{P_i}(\eta) = -1$ für $i = 1, \ldots, n$ und $supp(D) \cap supp(G) = \emptyset$ für den Schnitt $supp(D) \cap supp(D-G+(\eta)) = \emptyset$ gilt. Die Differentialform η existiert nach Lemma 2.2.5. Der Funktionenraum $\mathcal{L}(H) = \mathcal{L}(D-G+(\eta))$ ist somit sinnvoll definiert. Mit Satz 2.2.4 wissen wir bereits, dass es einen Isomorphismus ϕ gibt, der $\mathcal{L}(H)$ mittels $\phi(f) = f\eta$ auf $\Omega(G-D)$ abbildet, da (η) ein kanonischer Divisor ist. Verwenden wir die Voraussetzung, dass $res_{P_i}(\eta) = 1$ für $i = 1, \ldots, n$ ist, dann erhalten wir

$$f(P_i) = f(P_i) \cdot 1 = f(P_i) \cdot res_{P_i}(\eta) \overset{\text{(L. 2.2.4)}}{=} res_{P_i}(f\eta) \tag{2.14}$$

für alle $i = 1, \ldots, n$. Damit folgt unmittelbar die Gleichheit $C_\Omega(D, G) = C_\mathcal{L}(D, H)$, da die Funktionenräume isomorph sind und sich die Darstellung mit der Gleichung (2.14) beliebig überführen lässt. □

Proposition 2.3.1. *([Sti93], S. 205) Es seien y und t rationale Funktion aus F mit $v_{P_i}(t) = 1$, $v_{P_i}(y) = 0$ und $y(P_i) = 1$ für alle $i = 1, \ldots, n$. Für das Differential $\eta := y \cdot \lambda(t) = y \cdot \frac{dt}{t}$ gilt dann $v_{P_i}(\eta) = -1$ und $res_{P_i}(\eta) = 1$.*

BEWEIS. Nach Voraussetzung ist $v_{P_i}(t) = 1$, t also lokaler Parameter der Stelle P_i. Entwickeln wir y in eine Laurentreihe in der Stelle P_i bzgl. des lokalen Parameters t, so ist η von der Form

$$\begin{aligned} \eta = y \cdot \lambda(t) &= y \cdot \frac{dt}{t} \\ &= (1 + c_1 \cdot t + \ldots) \cdot \frac{dt}{t} \\ &= (1 \cdot t^{-1} + c_1 + \ldots)dt. \end{aligned}$$

Das Residuum ist dann tatsächlich 1 und es gilt $v_{P_i}(\eta) = -1$, da der erste von Null verschiedene Koeffizient in der Reihe der von t^{-1} ist. □

Als Folgerung erhalten wir für den Divisor H aus Satz 2.3.3

$$H = D - G + (\eta) = D - G + (y) + (dt) - (t),$$

wobei für $y, t \in F$ gelten muss $v_{P_i}(t) = 1$, $v_{P_i}(y) = 0$ und $y(P_i) = 1$ für alle $i = 1, \ldots, n$.

2.3.3 Decodieralgorithmus

Wir wollen nun zur Praxis zurückkehren und einen Algorithmus[1] zur Decodierung mit Fehlerkorrektur für geometrische Goppa-Codes vorstellen. Dieser

[1] Anmerkung: Der Algorithmus geht zurück auf Arbeiten von A.N. Skorobogatov und S.G. Vladut.

kann in ([Sti93], Kap. VII.5) nachgelesen werden. Zur Herleitung betrachten wir einen beliebigen residuellen Goppa-Code $C_\Omega(D,G)$ vom Typ (n,k^*,d^*) mit Divisoren D,G. Dieser korrigiert bis zu $t \leq \left[\frac{d^*-1}{2}\right]$ Fehler in einem Codewort der Länge n. Wir verwenden im weiteren folgende Bezeichnungen:

- empfangenes Wort: $a = (a_1, \ldots, a_n)$
- richtiges Codewort: $c = (c_1, \ldots, c_n)$
- Fehlervektor: $e = (e_1, \ldots, e_n) := a - c$
- Fehlerpositionen: $M := \{i;\ e_i \neq 0,\ 1 \leq i \leq n\}$
- Syndrom: $s(b) := [b,f] := \langle b, ev_D(f)\rangle = \sum_{j=1}^{n} b_j \cdot f(P_j)$, wobei $b = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$ und $f \in \mathcal{L}(G)$

Da $C_\Omega(D,G)^\perp = C_\mathcal{L}(D,G)$ gilt, erhalten wir mit der Abbildung $[\cdot,\cdot]$ [2] eine neue Beschreibung von $C_\Omega(D,G)$:

$$C_\Omega(D,G) = \{b \in \mathbb{F}_q^n;\ [b,f] = 0\ \forall f \in \mathcal{L}(G)\}. \tag{2.15}$$

Es sei von nun an $0 \leq w(e) = \#M \leq t$. Wir definieren einen weiteren Divisor $G_1 \in Div(F)$ mit folgenden drei Eigenschaften:

$$\begin{aligned} &supp(G_1) \cap supp(D) = \emptyset, \\ &\deg G_1 < \deg G + 2 - t, \\ &l(G_1) > t. \end{aligned} \tag{2.16}$$

Unser erster Schritt wird sein, eine Fehlerlokatorfunktion zu konstruieren, d.h. eine rationale Funktion f aus $\mathcal{L}(G_1)$, für die gilt $f(P_i) = 0$ für alle $i \in M$. In einem zweiten Schritt wollen wir dann die einzelnen Fehlerwerte e_i und damit den Fehlervektor e bestimmen. Dazu haben wir einige lineare Gleichungssysteme zu lösen. Wir fixieren zunächst Basen der Funktionenräume

$$\begin{gathered} \{f_1, \ldots, f_l\} \text{ für } \mathcal{L}(G_1), \\ \{g_1, \ldots, g_k\} \text{ für } \mathcal{L}(G - G_1), \\ \{h_1, \ldots, h_m\} \text{ für } \mathcal{L}(G). \end{gathered}$$

Diese hängen nicht vom empfangenen Wort a ab. Es ist offensichtlich $f_\lambda g_\rho$ wieder ein Element aus $\mathcal{L}(G)$. Wir betrachten das Gleichungssystem

$$\sum_{\lambda=1}^{l} [a, f_\lambda g_\rho] \cdot x_\lambda = 0 \text{ mit } \rho = 1, \ldots, k. \tag{2.17}$$

[2] $[\cdot,\cdot] :\ \mathbb{F}_q^n \times \mathcal{L}(G) \to F/\mathbb{F}_q,\ (b,f) \mapsto \sum_{j=1}^{n} b_j \cdot f(P_j)$ offensichtlich bilinear

Lemma 2.3.1. *Mit obigen Vereinbarungen und Bezeichnungen hat das Gleichungssystem* (2.17) *eine nicht-triviale Lösung. Ist* $(\alpha_1, \ldots, \alpha_l)$ *eine solche nicht-triviale Lösung, so setzen wir*

$$f := \sum_{\lambda=1}^{l} \alpha_\lambda f_\lambda \in \mathcal{L}(G_1). \tag{2.18}$$

Dann ist $f(P_i) = 0$ *für alle Fehlerstellen* $i \in M$*,* f *ist also eine Fehlerlokatorfunktion.*

Beweis. Der Funktionenraum $\mathcal{L}(G_1 - \sum_{i\in M} P_i)$ hat Dimension > 0, da $0 \leq \#M \leq t < l(G_1)$ nach Voraussetzung (2.16). Stellen wir nun eine Funktion $z \neq 0$ aus $\mathcal{L}(G_1 - \sum_{i\in M} P_i)$ als Linearkombination der fixierten Basisvektoren $\{f_1, \ldots, f_l\}$ dar

$$z = \sum_{\lambda=1}^{l} \gamma_\lambda f_\lambda,$$

wobei $\gamma_\lambda \in \mathbb{F}_q$, dann ist auch zg_ρ enthalten in $\mathcal{L}(G_1)$ für alle $\rho = 1, \ldots, k$ und wir erhalten

$$[a, zg_\rho] = \sum_{\lambda=1}^{l} [a, f_\lambda g_\rho] \cdot \gamma_\lambda. \tag{2.19}$$

Auf der anderen Seite wissen wir, dass für das Codewort $c \in C_\Omega(D, G)$ und ein beliebiges Wort aus $\mathcal{L}(G)$, also insbesondere auch für $zg_\rho \in \mathcal{L}(G_1)$ gilt $[c, zg_\rho] = 0$, da $C_\Omega(D, G)^\perp = C_\mathcal{L}(D, G)$. Wir fassen unsere bisherigen Ergebnisse zusammen und erhalten

$$\begin{aligned} [a, zg_\rho] &= [c + e, zg_\rho] = [c, zg_\rho] + [e, zg_\rho] \\ &= [e, zg_\rho] = \sum_{i=1}^{n} e_i \cdot z(P_i) \cdot g_\rho(P_i) = 0. \end{aligned} \tag{2.20}$$

Die Summe verschwindet, da alle Summanden verschwinden. Es ist nämlich $e_i = 0$, wenn $i \notin M$ und $z(P_i) = 0$, wenn $i \in M$, denn z ist nach Konstruktion aus $\mathcal{L}(G_1 - \sum_{i\in M} P_i)$ und damit ist $v_{P_i}(z) \geq 1$ für die $i \in M$.
Daraus folgt, dass $\gamma = (\gamma_1, \ldots, \gamma_l)$ eine nicht-triviale Lösung des Gleichungssystems (2.17) ist, denn $z \neq 0$ und die linke Seite von (2.19) verschwindet wegen (2.20).
Den zweiten Teil des Lemmas beweisen wir indirekt. Wir wählen eine beliebige nicht-triviale Lösung $(\alpha_1, \ldots, \alpha_l)$ aus und setzen $f := \sum_{\lambda=1}^{l} \alpha_\lambda f_\lambda$. Nehmen wir also an, es gäbe ein $i_0 \in M$, so dass $f(P_{i_0}) \neq 0$. Nach unseren allgemeinen

Voraussetzungen (2.16) ist

$$\deg(G - G_1 - \sum_{i \in M} P_i) \geq \deg G - \underbrace{\deg G_1}_{<\deg G+2-t} - t > -2.$$

Dann gilt mit dem Satz von Riemann-Roch auch

$$\mathcal{L}(G - G_1 - \sum_{i \in M} P_i) \subsetneqq \mathcal{L}(G - G_1 - \sum_{i \in M \setminus \{i_0\}} P_i).$$

Folglich ist es möglich, eine Funktion h in $\mathcal{L}(G - G_1)$ zu finden, so dass $h(P_{i_0}) \neq 0$ und $h(P_i) = 0$ für die übrigen $i \in M \setminus \{i_0\}$ ist. Wir berechnen wieder

$$\begin{aligned} [a, fh] = [e, fh] &= \sum_{i=1}^{n} e_i \cdot f(P_i) \cdot h(P_i) \\ &= e_{i_0} \cdot f(P_{i_0}) \cdot h(P_{i_0}) \neq 0. \end{aligned} \tag{2.21}$$

Da sich h als Linearkombination von $\{g_1, \ldots, g_k\}$ darstellen lässt und wir bereits aus den Gleichungen (2.17) und (2.19) wissen, dass für $\rho = 1, \ldots, k$

$$[a, fg_\rho] = \sum_{\lambda=1}^{l} [a, f_\lambda g_\rho] \cdot \alpha_\lambda = 0$$

gilt, haben wir einen Widerspruch. Also muss $f(P_i) = 0$ sein für alle $i \in M$.$_\square$

Bezeichnen wir mit $N_f := \{i;\ 1 \leq i \leq n,\ f(P_i) = 0\}$, so stellen wir fest, dass stets gilt

$$\deg G_1 \geq \# N_f, \tag{2.22}$$

denn es war $f \in \mathcal{L}(G_1 - \sum_{i \in N_f} P_i)$ nach Konstruktion. Wir können nun zum zweiten Schritt übergehen und den Fehlervektor e bestimmen. Wir betrachten dazu ein weiteres lineares Gleichungssystem

$$\sum_{i \in N_f} h_\mu(P_i) \cdot z_i = [a, h_\mu] \tag{2.23}$$

mit $\mu = 1, \ldots, m$ und $\{h_1, \ldots, h_m\}$ Basis von $\mathcal{L}(G)$.

Lemma 2.3.2. *Unter obigen Voraussetzungen besitzt das Gleichungssystem* (2.23) *genau eine Lösung. Die eindeutig bestimmte Lösung entspricht den Komponenten e_i des Fehlervektors, für die $f(P_i) = 0$ gilt.*

BEWEIS. Wir zeigen zunächst die Existenz einer Lösung. Da $e_i = 0$ für alle $i \notin M$, ist $(e_i)_{i \in N_f}$ eine Lösung des Systems (2.23), denn

$$\begin{aligned}[a, h_\mu] &= [c + e, h_\mu] = [e, h_\mu] \\ &= \sum_{i=1}^{n} e_i \cdot h_\mu(P_i) = \sum_{i \in N_f} e_i \cdot h_\mu(P_i).\end{aligned}$$

Zum Beweis der Eindeutigkeit nehmen wir an, es gäbe eine zweite Lösung des Systems, etwa $(e'_i)_{i \in N_f}$. Wir können den Vektor $e' := (e'_1, \ldots, e'_n)$ bilden, indem wir $e'_i = 0$ setzen für die Positionen $i = 1, \ldots, n$, für die $f(P_i) \neq 0$ ist. Damit erhalten wir

$$\begin{aligned}[e', h_\mu] = \sum_{i \in N_f} h_\mu(P_i) \cdot e'_i = [a, h_\mu] = [e, h_\mu] \\ \Rightarrow [e', h_\mu] - [e, h_\mu] = [e' - e, h_\mu] = 0\end{aligned}$$

für $\mu = 1, \ldots, n$. Da $\{h_1, \ldots, h_m\}$ Basisvektoren von $\mathcal{L}(G)$ sind, folgt, dass $e' - e$ Element von $C_\Omega(D, G)$ ist, wegen Gleichung (2.15). Wir schätzen den Abstand von e zu e', also das Hamming-Gewicht von $e' - e$ ab:

$$\begin{aligned}d(e', e) = w(e' - e) \leq w(e') + w(e) \\ \leq \#N_f + t \overset{(2.22)}{\leq} \deg G_1 + t \overset{(2.16)}{<} \deg G + 2 \overset{\text{S. 2.3.2}}{\leq} d^*.\end{aligned}$$

Da aber der Minimalabstand von $C_\Omega(D, G)$ gerade d^* ist, bleibt nur noch $e' = e$ um die Abschätzung zu erfüllen. □

Zusammengefasst erhalten wir folgenden Decodieralgorithmus:

1. Berechne das Syndrom $s(a)$. Ist $s(a) = 0$, dann ist kein Fehler enthalten/erkennbar; andernfalls nächster Schritt.

2. Bestimme eine nicht-triviale Lösung $(\alpha_1, \ldots, \alpha_l)$ des Systems (2.17) und bilde damit die Fehlerlokatorfunktion $f = \sum_{\lambda=1}^{l} \alpha_\lambda f_\lambda$.

3. Berechne die *möglichen* Fehlerpositionen $1 \leq i \leq n$ über $f(P_i) = 0$ für diese Stellen.

4. Der Fehlervektor ist die Lösung des Systems (2.23), wobei $e_i = 0$ gesetzt wird, falls $f(P_i) \neq 0$ für eine Position.

5. Rekonstruiere nun das korrekte Codewort c aus dem empfangenen Wort a mittels $c = a - e$. Probe durch Syndromberechnung von c.

Wir wollen diesen Algorithmus in einem Beispiel testen. Gleichzeitig wollen wir auch zeigen, wie ein Goppa-Code konkret aussehen kann und wie Codewörter erzeugt werden.

Beispiel 2.3.1. (siehe auch Anhang A.2) Unser Code $C_\Omega(D, G)$ soll die Länge $n = 15$ und einen Minimalabstand von wenigstens $d^* = 7$ haben. Wir betrachten die projektive Gerade über dem Körper $\mathbb{F} = \mathbb{F}_{16}$. Ihre Punkte identifizieren wir wieder mit den Stellen vom Grad 1 aus $\mathbb{P}_F$ in der üblichen Weise.

Stellen vom Grad 1 in $\mathbb{P}_F$	normierte Primpolynome 1-ten Grades	Punkte der projektiven Geraden $\mathbb{P}^1(\mathbb{F})$
P_0	X	$(1:0)$
P_1	$X-\zeta$	$(1:\zeta)$
$\vdots$	$\vdots$	$\vdots$
P_{15}	$X-\zeta^{15}$	$(1:\zeta^{15})$
∞	$X'=1/X$	$\infty=(0:1)$

Für den Divisor D definieren wir dann $D := P_1 + \ldots + P_{15}$, womit wir auch die Vorgabe $n = 15$ einhalten. Die beiden übrigen Parameter k^*, d^* werden durch den Divisor G bestimmt. Da $supp(D) \cap supp(G) = \emptyset$ gelten soll, bieten sich für G lineare Kombinationen aus P_0 und ∞ an. Ferner gilt nach Satz 2.3.1, dass $k^* = n - \deg G - 1$, falls $\deg G < \deg D$ und $d^* = n - k^* + 1$, wegen der Optimalität der Goppa-Codes. Mit $d^* = 7$ folgt $k^* = 9$. Wir können also $G := 5 \cdot \infty$ als zweiten Divisor wählen. Daraus folgt, dass $k = n - k^* = 6$. Der Funktionenraum $\mathcal{L}(G)$ ist dann gerade die Menge der Funktionen mit Koeffizienten aus $\mathbb{F}$ vom Grad < 6, also $\mathbb{F}[X]_{\deg<6}$. Eine zu $C_\mathcal{L}(D, G)$ gehörige Generatormatrix ist dann wegen der Basis $h = \{1, X, \ldots, X^5\}$ von $\mathcal{L}(G)$ die Matrix

$$H_{C_\Omega(D,5\cdot\infty)} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \zeta & \zeta^2 & \cdots & \zeta^{15} \\ \vdots & \vdots & & \vdots \\ \zeta^5 & (\zeta^2)^5 & \cdots & (\zeta^{15})^5 \end{pmatrix} = G_{C_\mathcal{L}(D,5\cdot\infty)}.$$

Sie ist die Kontrollmatrix von $C_\Omega(D, G)$[3]. Wie wir aus Bemerkung 2.3.1 wissen, lässt sich daraus eine Generatormatrix des Residuen-Codes bilden. Diese

[3]Die Ähnlichkeit zur Kontrollmatrix eines BCH-Codes ist kein Zufall, wie wir im nächsten Kapitel sehen werden.

geben wir hier nur als Resultat an.

$$G_{C_\Omega(D,5\cdot\infty)} = \begin{pmatrix} 1 & \zeta^6 & \zeta^3 & \zeta^{14} & \zeta^{13} & \zeta^{11} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \zeta^{11} & \zeta^9 & \zeta^{12} & \zeta & \zeta^4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \zeta & \zeta^{11} & \zeta^6 & \zeta^2 & \zeta^2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \zeta^2 & \zeta^6 & \zeta^4 & \zeta^6 & \zeta^8 & \zeta & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \zeta & \zeta^{12} & \zeta^{13} & \zeta^3 & \zeta^{12} & \zeta^{11} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \zeta^{11} & \zeta^{13} & \zeta^6 & \zeta^{14} & \zeta^{11} & \zeta^2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \zeta^2 & \zeta^{12} & \zeta^{11} & \zeta^{11} & \zeta^{11} & \zeta^5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \zeta^5 & \zeta^4 & \zeta^{11} & \zeta^2 & \zeta^9 & \zeta^6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \zeta^6 & \zeta^3 & \zeta^{14} & \zeta^{13} & \zeta^{11} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Wir wollen eine Information von 9 Bit, z.B. $(\zeta^7, \zeta^{15}, \zeta^{12}, \zeta^4, \zeta^6, \zeta, \zeta^4, \zeta^5, \zeta^{14})$ codieren. Dazu lassen wir die Generatormatrix $G_{C_\Omega(D,5\cdot\infty)}$ von rechts darauf wirken und erhalten als Codewort

$$c = (\zeta^{14}, \zeta^3, \zeta, \zeta^8, 1, \zeta^{13}, \zeta^7, 1, \zeta^{12}, \zeta^4, \zeta^6, \zeta, \zeta^4, \zeta^5, \zeta^{14}).$$

Während der Übertragung sollen drei Fehler auftreten, so dass das empfangene Wort lautet

$$a = (\zeta^{14}, \zeta^3, \zeta, \zeta^8, \zeta^9, \zeta^{13}, \zeta^7, \zeta^{11}, \zeta^{12}, \zeta^4, \zeta^6, \zeta, \zeta^4, \zeta^2, \zeta^{14}).$$

Dies entspricht der Addition $a = c + e$ eines Fehlervektors

$$e = (0, 0, 0, 0, \zeta^2, 0, 0, \zeta^{14}, 0, 0, 0, 0, 0, \zeta^6, 0).$$

zum Vektor c. Auf der Empfängerseite wird zunächst das Syndrom berechnet. Dazu wählen wir o.B.d.A. $f(X) = X$ aus $\mathcal{L}(G)$. Damit ergibt sich

$$s(a) = [a, f(X)] = \sum_{i=1}^{n} a_i f(P_i) = \zeta^5 \neq 0$$

und wir stellen fest, dass a fehlerbehaftet ist. Nun benötigen wir den Divisor G_1 mit den Eigenschaften (2.16). Sei $G_1 = 3 \cdot \infty$ und $\{f_1, \ldots, f_4\} = \{1, X, \ldots, X^3\}$ Basis von $\mathcal{L}(G_1)$. Der Funktionenraum $\mathcal{L}(G - G_1) = \mathcal{L}(2 \cdot \infty)$ habe $\{g_1, g_2, g_3\} = \{1, X, X^2\}$ als Basis. Wir suchen jetzt eine Fehlerlokatorfunktion. Dazu brauchen wir eine nicht-triviale Lösung des homogenen Gleichungssystems

$$\sum_{\lambda=1}^{4} [a, f_\lambda g_\rho] \cdot x_\lambda = 0 \text{ mit } \rho = 1, \ldots, 3.$$

Wir setzen ein und erhalten das folgende System in Matrixschreibweise

$$\begin{pmatrix} \zeta^5 & \zeta^7 & \zeta^7 & \zeta \\ \zeta^7 & \zeta^7 & \zeta & 1 \\ \zeta^7 & \zeta & 1 & \zeta^2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$$

Als eine nicht-triviale Lösung finden wir $(\zeta^{12}, \zeta^3, \zeta^4, 1)$. Die Fehlerlokatorfunktion ist dann

$$f(X) = \zeta^{12} + \zeta^3 \cdot X + \zeta^4 \cdot X^2 + X^3$$

und wir ermitteln damit die potentiellen Fehlerstellen, indem wir für $1 \leq i \leq n$ die Werte von $f(P_i)$ berechnen. Wir sehen dabei, dass $f(P_i)$ für $i \in \{5, 8, 14\} =: N_f$ verschwindet. Damit haben wir die möglichen Fehlerpositionen gefunden. Nun wollen wir die Fehlerwerte mit Hilfe des Gleichungssystems

$$\begin{pmatrix} 1 & 1 & 1 \\ \zeta^5 & \zeta^8 & \zeta^{14} \\ \zeta^{10} & \zeta & \zeta^{13} \\ \zeta^{15} & \zeta^9 & \zeta^{12} \\ \zeta^5 & \zeta^2 & \zeta^{11} \\ \zeta^{10} & \zeta^{10} & \zeta^{10} \end{pmatrix} \cdot \begin{pmatrix} e_5 \\ e_8 \\ e_{14} \end{pmatrix} = \begin{pmatrix} \zeta^7 \\ \zeta^5 \\ \zeta^5 \\ \zeta \\ 1 \\ \zeta^2 \end{pmatrix} \tag{2.24}$$

berechnen. Der Computer liefert als Fehlerwerte $e_5 = \zeta^2, e_8 = \zeta^{14}, e_{14} = \zeta^6$. Mit der Beziehung $c = a - e$ lässt sich nun leicht das richtige Codewort herstellen.

Kapitel 3

Elementare Codes als Goppa-Codes

Ziel dieses Kapitels soll es sein, zu untersuchen, wie sich die im ersten Kapitel beschriebenen elementaren Codes in der Theorie der Goppa-Codes aus dem zweiten Kapitel darstellen. Wir werden sehen, dass den generalisierten Reed-Solomon-Codes eine besondere Bedeutung bei diesen Untersuchungen zukommt. Wir orientieren uns dabei an [Sti93], Kapitel II, reduzieren jedoch wieder auf die projektive Gerade ($g = 0$).
Grundsätzlich sollen auch in diesem Kapitel wieder folgende Bezeichnungen und Bedingungen gelten:

- $F = \mathbb{F}(X)$ ist der rationale Funktionenkörper über dem endlichen Körper $\mathbb{F} = \mathbb{F}_q$,
- $P_1, \ldots, P_n$ sind paarweise verschiedene Stellen aus $\mathbb{P}_F$ vom Grad 1,
- $D := P_1 + \ldots + P_n$ ist ein Divisor aus $Div(F)$,
- G ist ein Divisor aus $Div(F)$, so dass $supp(D) \cap supp(G) = \emptyset$.

Wenn wir von rationalen Goppa-Codes reden, dann meinen wir damit von nun an rational-geometrische Goppa-Codes.

3.1 Reed-Solomon-Codes

Im ersten Kapitel haben wir gesehen, dass wir Reed-Solomon-Codes zu generalisierten Reed-Solomon-Codes verallgemeinern können. Insbesondere konnten wir dort auch alle Reed-Solomon-Codes als generalisierte Reed-Solomon-Codes mit $a = (\zeta, \zeta^2, \ldots, \zeta^n)$ und $v = (1, \ldots, 1)$ schreiben, wobei ζ eine primitive n-te Einheitswurzel war und $n = q - 1$ galt. Abschnitt 2.1

hat gezeigt, dass generalisierte Reed-Solomon-Codes $GRS_k(a, v)$ affin-lineare Goppa-Codes sind, wenn der Vektor a mit dem Punktesystem D übereinstimmt und $v = (1, \ldots, 1)$ ist. Um das Verhältnis von generalisierten Reed-Solomon-Codes und rationalen Goppa-Codes zu klären hilft uns der folgende

Satz 3.1.1. *Es sei $C = C_{\mathcal{L}}(D, G)$ ein rationaler Goppa-Code über dem endlichen Körper $\mathbb{F} = \mathbb{F}_q$ vom Typ (n, k, d).*

1. *Wenn $n \leq q$ ist, dann existieren Elemente $a_1, \ldots, a_n$ in $\mathbb{F}_q$ mit $a_i \neq a_j$ für $i \neq j$ und $v_1, \ldots, v_n$ in $\mathbb{F}_q^*$, so dass*

$$C = \{(v_1 f(a_1), v_2 f(a_2), \ldots, v_n f(a_n));\ f \in \mathbb{F}_q[X]_{\deg < k}\}.$$

   Die Matrix

$$G_C = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \\ a_1 v_1 & a_2 v_2 & \cdots & a_n v_n \\ a_1^2 v_1 & a_2^2 v_2 & \cdots & a_n^2 v_n \\ \vdots & \vdots & & \vdots \\ a_1^{k-1} v_1 & a_2^{k-1} v_2 & \cdots & a_n^{k-1} v_n \end{pmatrix} \tag{3.1}$$

   ist eine Generatormatrix von C.

2. *Ist $n = q + 1$, dann hat C die Matrix*

$$G_C = \begin{pmatrix} v_1 & v_2 & \cdots & v_{n-1} & 0 \\ a_1 v_1 & a_2 v_2 & \cdots & a_{n-1} v_{n-1} & 0 \\ a_1^2 v_1 & a_2^2 v_2 & \cdots & a_{n-1}^2 v_{n-1} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1^{k-2} v_1 & a_2^{k-2} v_2 & \cdots & a_{n-1}^{k-2} v_{n-1} & 0 \\ a_1^{k-1} v_1 & a_2^{k-1} v_2 & \cdots & a_{n-1}^{k-1} v_{n-1} & 1 \end{pmatrix} \tag{3.2}$$

   als Generatormatrix, wobei $\mathbb{F}_q = \{a_1, \ldots, a_{n-1}\}$ und $v_1, \ldots, v_{n-1}$ aus $\mathbb{F}_q^*$.

Beweis. Es sei $C = C_{\mathcal{L}}(D, G)$ ein beliebiger rationaler Goppa-Code über $\mathbb{F}_q$ vom Typ (n, k, d) mit $D = P_1 + \ldots + P_n$, P_i Stelle vom Grad 1 aus $\mathbb{P}_F$.

1. Fall $n \leq q$: Nach Voraussetzung enthält $supp(D)$ nicht alle möglichen Stellen vom Grad 1, denn $D = P_1 + \ldots + P_n$, aber $\mathbb{F}_q$ hat $q \geq n$ Elemente. Es gibt also eine Stelle vom Grad 1 aus $\mathbb{P}_F$, die nicht in $supp(D)$ enthalten ist; es sei P eine solche Stelle.
   Wählen wir nun eine weitere Stelle vom Grad 1 aus $\mathbb{P}_F$, die von P verschieden ist, etwa $Q = P_1 \neq P$, dann gilt $l(Q - P) = 1$, wegen Satz

2.2.3, denn es ist $\deg(Q-P) = \deg Q - \deg P = 1-1 = 0$.
Daraus folgt weiter, dass $Q - P =: (\varphi)$ ein Hauptdivisor ist, denn nur für Hauptdivisoren ist der Grad Null.
Dann ist $\varphi = \frac{f(X)}{g(X)}$ eine rationale Funktion aus F mit $f(X), g(X) \in \mathbb{F}[X]$, $g(X) \neq 0$.
Wegen

$$\begin{aligned} Q - P = (\varphi) &= \left(\frac{f(X)}{g(X)}\right) = (f(X)) - (g(X)) \\ &= \sum_{\infty \neq P \in \mathbb{P}_F^1} v_{P_i}(f(X)) P_i - \deg f(X) \cdot \infty \\ &- \sum_{\infty \neq P \in \mathbb{P}_F^1} v_{P_i}(g(X)) P_i + \deg g(X) \cdot \infty \end{aligned}$$

folgt $P = \infty$ und $g(X) = c \in \mathbb{F}^*$. Sei o.B.d.A. $c = 1$. Die Funktion $\varphi = f(X)$ ist ein erzeugendes Element des rationalen Funktionenkörpers $F = \mathbb{F}_q(X)/\mathbb{F}_q$.
Nach Voraussetzung ist C vom Typ (n, k, d), wobei $k = \deg G + 1$ ist. Betrachten wir nun den Divisor $(k-1) \cdot \infty - G =: (u)$, $0 \neq u \in F$, so hat dieser den Grad Null und ist demnach ebenfalls ein Hauptdivisor. Nach Konstruktion ist $u \cdot \varphi^j$ für $j = 0, 1, \ldots, k-1$ enthalten in $\mathcal{L}(G)$, denn $(u \cdot \varphi^j) = (u) + (\varphi^j) = (u) + j(Q - \infty) \geq (u) - (k-1) \cdot \infty = -G$ und Q hatten wir o.B.d.A. gleich $P_1 \in \mathit{supp}(D)$ gewählt. Die k Elemente $u, u \cdot \varphi, u \cdot \varphi^2, \ldots, u \cdot \varphi^{k-1}$ sind außerdem linear unabhängig über $\mathbb{F}_q$ und da die Dimension von $\mathcal{L}(G)$ gerade k ist, bilden sie eine Basis von $\mathcal{L}(G)$. Es ist also

$$\mathcal{L}(G) = \{u \cdot \psi(\varphi);\ \psi \in \mathbb{F}_q[X]_{\deg < k}\}.$$

Setzen wir nun noch $a_i := \varphi(P_i)$ und $v_i := u(P_i)$, so erhalten wir

$$u \cdot \psi(\varphi)(P_i) = u(P_i) \cdot \psi(\varphi(P_i)) = v_i \cdot \psi(a_i)$$

für $i = 1, \ldots, n$ und damit

$$C = C_{\mathcal{L}}(D, G) = \{(v_1 \cdot \psi(a_1), v_2 \cdot \psi(a_2), \ldots, v_n \cdot \psi(a_n));\ \psi \in \mathbb{F}_q[X]_{\deg < k}\}.$$

Die Generatormatrix ergibt sich in der gewünschten Form, wenn wir in die k Zeilen jeweils das zu $u \cdot \varphi^j$ gehörige Codewort $(v_1 a_1^j, \ldots, v_n a_n^j)$, $j = 0, 1, \ldots, k-1$ eintragen und damit die Basis $1, X, \ldots, X^{k-1}$ von $\mathbb{F}_q[X]_{\deg < k}$ ausnutzen

2. Fall $n = q+1$: Nun haben wir für den Divisor $D = P_1+\ldots+P_{n-1}+P_n = P_1+\ldots+P_q+\infty$. Wir wählen die Funktion $\varphi \in F$, wie im ersten Fall, so, dass sie erzeugend ist und dass $P = P_n = \infty$. Nun betrachten wir wieder den Divisor $(u) := (k-1)\cdot\infty - G$, $0 \neq u \in F$. Analog ist dann auch $u, u\cdot\varphi, u\cdot\varphi^2, \ldots, u\cdot\varphi^{k-1}$ wieder Basis von $\mathcal{L}(G)$. Die Elemente $a_i := \varphi(P_i)$ sind nun jedoch nur für $i = 1,\ldots,n-1 = q$ paarweise verschieden, da $\mathbb{F}_q = \{a_1,\ldots,a_{n-1} = a_q\}$. Mit den gleichen Ersetzungen $v_i := u(P_i) \in \mathbb{F}_q^*$, wie oben, erhalten wir für die ersten $k-1$ Basisvektoren ($j = 0,\ldots,k-2$)

$$((u\varphi^j)(P_1),\ldots,u\varphi^j(P_n)) = (a_1^j v_1,\ldots,a_{n-1}^j v_{n-1},0).$$

Den k-ten Basisvektor ($j = k-1$) erzeugen wir, indem wir die n-te Komponente durch ein $\gamma \neq 0$ aus $\mathbb{F}_q$ ersetzen:

$$((u\varphi^{k-1})(P_1),\ldots,u\varphi^{k-1}(P_n)) = (a_1^{k-1}v_1,\ldots,a_{n-1}^{k-1}v_{n-1},\gamma).$$

Dadurch sichern wir die lineare Unabhängigkeit zu den übrigen Basisvektoren. Die Generatormatrix erhalten wir mit $\gamma = 1$. □

Korollar 3.1.1. *Jeder generalisierte Reed-Solomon-Code lässt sich als rationaler Goppa-Code darstellen und jeder rationale Goppa-Code ist darstellbar als generalisierter Reed-Solomon-Code.*

Beweis. Sei $GRS_k(a,v)$ ein generalisierter Reed-Solomon-Code der Länge n mit $a = (a_1,\ldots,a_n)$, $a_i \in \mathbb{F}_q$ und $v = (v_1,\ldots,v_n)$, $v_i \in \mathbb{F}_q^*$. Wir betrachten wieder den rationalen Funktionenkörper $F = \mathbb{F}(X)$ und identifizieren die Stellen P_i mit $(X-a_i)$ für $i = 1,\ldots,n$ und ∞ mit $X' = 1/X$ (siehe Korollar 2.2.1). Wir wählen nun $u(X) \in F$ gerade so, dass $u(P_i) = v_i$ für $i = 1,\ldots,n$. Die Divisoren D, G definieren wir in der folgenden Weise:

$$D := P_1+\ldots+P_n, \tag{3.3}$$
$$G := (k-1)\cdot\infty - (u). \tag{3.4}$$

Damit haben wir eine Situation hergestellt, wie sie im Beweis des Satzes auftaucht. Folglich ist der Code $C_{\mathcal{L}}(D,G)$ ein rationaler Goppa-Code, der durch die Divisoren D, G bestimmt ist. Die Umkehrung folgt ebenfalls unmittelbar aus dem Beweis des Satzes. □

Da wir bereits eingesehen haben, dass jeder Reed-Solomon-Code auch ein generalisierter Reed-Solomon-Code ist, können wir somit Reed-Solomon-Codes, wie etwa die Brenncodes für CD und DVD, als rationale Goppa-Codes darstellen. Dies wollen wir im folgenden Beispiel praktisch durchführen.

Beispiel 3.1.1. Wir betrachten einen $(255, 251, 5)$-Reed-Solomon-Code, wie er bei der Sicherung von Audiodaten auf CD verwendet wird. Wie wir bereits in Beispiel 1.3.1 gesehen haben, wird dieser Code vom Generatorpolynom

$$g(X) = (X - \zeta)(X - \zeta^2)(X - \zeta^3)(X - \zeta^4)$$

erzeugt, wobei ζ eine primitive 255-te Einheitswurzel aus dem Körper $\mathbb{F}_{2^8} = \mathbb{F}_2[T]/(T^8 + T^7 + T^2 + T + 1)$ ist. Den Körper fassen wir als Vektorraum $\mathbb{F}_2^8$ mit Basis $\{1, \zeta, \zeta^2, \ldots, \zeta^7\}$ auf und finden damit eine Darstellung der 256 Elemente von $\mathbb{F}_{2^8}$ als Linearkombinationen aus diesen Basiselementen. Um die zugehörige Generatormatrix dieses zyklischen Codes zu finden, multiplizieren wir das Generatorpolynom $g(X)$ aus und konstruieren aus den Koeffizienten die 251×255-Generatormatrix mit Einträgen aus $\mathbb{F}_{2^8}$.

$$G_{CD} = \begin{pmatrix} \zeta^{10} & \zeta^{48} & \zeta^{52} & \zeta^{43} & 1 & 0 & & \cdots & 0 \\ 0 & \zeta^{10} & \zeta^{48} & \zeta^{52} & \zeta^{43} & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & & & \ddots & & \vdots \\ 0 & \cdots & 0 & \zeta^{10} & \zeta^{48} & \zeta^{52} & \zeta^{43} & 1 & 0 \\ 0 & \cdots & & 0 & \zeta^{10} & \zeta^{48} & \zeta^{52} & \zeta^{43} & 1 \end{pmatrix}$$

Mit dieser Matrix können wir nun Information mit einer Länge von 251 Stellen in Codewörter der Länge 255 codieren. Unser Reed-Solomon-Code ist 2-fehlerkorrigierend, da für den Fehlerkorrekturindex gilt

$$e = \left\lfloor \frac{d-1}{2} \right\rfloor = 2.$$

Der Decodier- und Fehlerkorrekturalgorithmus funktioniert wie in Abschnitt 1.3.2 und Beispiel 1.3.5 beschrieben.
Zur Darstellung als rationaler Goppa-Code benötigen wir als Zwischenschritt eine Darstellung unseres Codes als generalisierter RS-Code $GRS_{251}(a, v)$. Diese erhalten wir, wenn wir

$$a = (\zeta, \zeta^2, \ldots, \zeta^{255}) \tag{3.5}$$

$$v = (1, 1, \ldots, 1) \tag{3.6}$$

setzen. Dann sind die Codewörter von $GRS_{251}(a, v)$ gemäß Definition 1.3.5

$$C_{GRS_{251}(a,v)} = \{(f(\zeta), f(\zeta^2), \ldots, f(\zeta^{255}));\ f \in \mathbb{F}_{2^8}[X]_{\deg<251}\}$$

Wir wollen nun die beiden Divisoren D, G bestimmen. Dazu identifizieren wir zunächst wieder die Stellen vom Grad 1 mittels der normierten linearen Primpolynome mit den Punkten der projektiven Geraden über $\mathbb{F}_{2^8}$.

Stellen vom Grad 1 in $\mathbb{P}_F$	normierte Primpolynome 1-ten Grades	Punkte der projektiven Geraden $\mathbb{P}^1(\mathbb{F})$
P_0	X	$(1:0)$
P_1	$X-\zeta$	$(1:\zeta)$
$\vdots$	$\vdots$	$\vdots$
P_n	$X-\zeta^n$	$(1:\zeta^n)$
∞	$X'=1/X$	$\infty=(0:1)$

Den Divisor $D = P_1 + \ldots + P_{255}$ haben wir damit bereits gefunden. Dem Beweis von Korollar 3.1.1 folgend, konstruieren wir G, indem wir eine Funktion $u(X)$ finden, so dass gilt $u(P_i) = 1$ für $i = 1, \ldots, 255$. Die Funktion

$$u(X) = 1$$

erfüllt diese Forderung, denn $u(P_i) = u(X) \bmod P_i = 1$. Wir erhalten damit für den Divisor $G = 250 \cdot \infty - (1)$. Wir können nun den rationalen Goppa-Code $C_{\mathcal{L}}(D, G)$ angeben und eine Generatormatrix konstruieren. Im weiteren ist $F = \mathbb{F}_{2^8}(X)$ der Körper der rationalen Funktionen über $\mathbb{F}_{2^8}$.

$$C_{\mathcal{L}}(D,G) = \{(\psi(\varphi(P_1)), \ldots, \psi(\varphi(P_{255}))); \ \psi \in \mathbb{F}_{2^8}[X]_{\deg<251}\}$$

Wir haben eine ähnliche Rechnung bereits in Beispiel 2.2.2 durchgeführt. Es gilt also offenbar tatsächlich $C_{\mathcal{L}}(D,G) = C_{GRS_{251}(a,v)} = C_{CD}$. Als Generatormatrix verwenden wir Matrix (3.1), denn es ist ja $n = q - 1 \leq q$. Die Funktion $\varphi \in F$ musste lediglich erzeugendes Element des rationalen Funktionenkörpers F sein. Wir können daher $\varphi(X) = X$ wählen und erhalten mit Matrix (3.1) als Generatormatrix

$$G_{C_{\mathcal{L}}(D,G)} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \zeta & \zeta^2 & \cdots & \zeta^{255} \\ \vdots & \vdots & & \vdots \\ \zeta^{250} & \zeta^{(2\cdot 250)} & \cdots & \zeta^{(255\cdot 250)} \end{pmatrix}.$$

Mit Bemerkung 2.3.1 können wir leicht auch eine Kontrollmatrix des Codes $C_{\mathcal{L}}(D, G)$ finden. Wir verweisen an dieser Stelle auf den Anhang A.3. Dort ist die Generatormatrix in Standardform eines $(28, 24, 5)$-RS-Codes als rationaler Goppa-Code über der projektiven Geraden angegeben (inklusive eines vollständigen Satzes von Kontrollfunktionen, d.h. Funktionen, deren Residuen in den Stellen $P_1, \ldots, P_{28}$ die Einträge der Kontrollmatrix bilden).

3.2 BCH-Codes im engeren Sinne

Wir haben im Abschnitt 1.3.2 BCH-Codes ganz allgemein definiert und als eine Verallgemeinerung der Reed-Solomon-Codes erkannt. Da wir uns dort nur auf BCH-Codes im engeren Sinne beschränkt haben, wollen wir uns auch hier nur auf diesen Spezialfall konzentrieren.

Satz 3.2.1. *Es sei n Teiler von $q^s - 1$ sowie $F = \mathbb{F}_{q^s}(X)$ und es sei C ein BCH-Code im engeren Sinne vom Typ (n, k, d) zum Entwurfsabstand δ mit einer primitiven n-ten Einheitswurzel ζ. Es gelte folgende Zuordnung für $i = 1, \ldots, n$: $P_i \leftrightarrow X - \zeta^{i-1}$ und $P_0 \leftrightarrow X$. Den Divisor D definieren wir durch $D := D_\zeta := P_1 + \ldots + P_n$. Ist nun $a \in \mathbb{Z}$ eine ganze Zahl mit $0 < a < n$, dann ist:*

1. $$C_{\mathcal{L}}(D_\zeta, -P_0 + a \cdot \infty) = C^{\perp},$$

   wobei für den Entwurfsabstand $\delta = a + 1$ gilt.

2. *Der duale Code von $C_{\mathcal{L}}(D_\zeta, -P_0 + a \cdot \infty)$ ist gegeben durch*

   $$C_{\mathcal{L}}(D_\zeta, -P_0 + a \cdot \infty)^{\perp} = C_{\mathcal{L}}(D_\zeta, b \cdot \infty),$$

   mit $b = n - a - 1$.

Beweis. Wir gehen von einem rationalen Goppa-Code $C_{\mathcal{L}}(D_\zeta, -P_0 + a \cdot \infty)$ mit $0 < a < n$ aus. Die Elemente X^j mit $j = 1, \ldots, a$ bilden eine Basis des Funktionenraumes $\mathcal{L}(-P_0 + a \cdot \infty)$. Also ist die Matrix

$$H_{C_{\mathcal{L}}} := \begin{pmatrix} 1 & \zeta & \zeta^2 & \cdots & \zeta^{(n-1)} \\ 1 & \zeta^2 & (\zeta^2)^2 & \cdots & (\zeta^2)^{(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \zeta^a & (\zeta^a)^2 & \cdots & (\zeta^a)^{(n-1)} \end{pmatrix}$$

eine Generatormatrix von $C_{\mathcal{L}}(D_\zeta, -P_0 + a \cdot \infty)$. Ersetzen wir nun noch $\delta = a + 1$, so erkennen wir in $H_{C_{\mathcal{L}}}$ aber auch die Kontrollmatrix eines primitiven BCH-Codes zum Entwurfsabstand δ aus Satz 1.3.3 wieder:

$$H_{C_{\mathcal{L}}} = \begin{pmatrix} 1 & \zeta & \zeta^2 & \cdots & \zeta^{(n-1)} \\ 1 & \zeta^2 & (\zeta^2)^2 & \cdots & (\zeta^2)^{(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \zeta^{(\delta-1)} & (\zeta^{(\delta-1)})^2 & \cdots & (\zeta^{(\delta-1)})^{(n-1)} \end{pmatrix}.$$

Wir haben jetzt also einen Zusammenhang zwischen einem rationalen Goppa-Code und dem dualen Code eines BCH-Codes (beide mit den entsprechenden Parametern). Mit der zweiten Aussage des Satzes gelangen wir zu unserem Ziel, eine Darstellung von BCH-Codes als rationale Goppa-Codes zu finden. Zum Beweis erinnern wir uns an Satz 2.3.3. Demnach genügt es, eine (logarithmische) Differentialform η mit $v_{P_i}(\eta) = -1$ und $res_{P_i}(\eta) = 1$ zu finden, so dass gilt

$$b \cdot \infty = D_\zeta - G + (\eta) = D_\zeta + P_0 - a \cdot \infty + (\eta),$$

wobei $b = n - a - 1$ sein soll. Durch Umformen erhalten wir für (η) die Bedingung

$$(\eta) = (n-1) \cdot \infty - D_\zeta - P_0. \tag{3.7}$$

Mit Proposition 2.3.1 wissen wir, dass eine solche Differentialform η eine Darstellung der Form $\eta = y \cdot \lambda(t)$ besitzt, wobei für $y, t \in F$ gelten muss $v_{P_i}(t) = 1$, $v_{P_i}(y) = 0$ und $y(P_i) = 1$ für alle $i = 1, \ldots, n$. Wählen wir $t = h(X) = \prod_{i=1}^{n}(X - \zeta^i) = X^n - 1$ und $y = X^{-n}$ so erhalten wir für den kanonischen Divisor

$$\begin{aligned}(\eta) &= (y) + (h'(X)) - (h(X)) - 2 \cdot \infty \\ &= (X^{-n}) + (h'(X)) - D_\zeta + n \cdot \infty - 2 \cdot \infty \\ &= -n \cdot P_0 + n \cdot \infty + ((n-1) \cdot (P_0 - \infty)) - D_\zeta + (n-2) \cdot \infty \\ &= -P_0 - D_\zeta + (n-1) \cdot \infty\end{aligned} \tag{3.8}$$

und damit die geforderte Bedingung aus Gleichung (3.7). Wir wollen uns noch kurz überzeugen, dass $y(P_i) = 1$ gilt, alle übrigen Forderungen sind offensichtlich erfüllt. Dazu berechnen wir

$$\begin{aligned}&h(X) = X^n - 1 = 0 \bmod P_i \\ \Leftrightarrow &X^n = 1 \bmod P_i \\ \Leftrightarrow &1 = X^{-n} \bmod P_i = y(P_i).\end{aligned}$$

Die Bewertungen von η in den Stellen $P_i \in supp(D)$ sind alle -1 und die Residuen in diesen Stellen sind $+1$ nach Proposition 2.3.1. Somit ist η die gesuchte (logarithmische) Differentialform und $C_{\mathcal{L}}(D_\zeta, (n-a-1) \cdot \infty)$ eine Darstellung eines BCH-Codes im engeren Sinne als rationaler Goppa-Code, was zu zeigen war. □

Wir wollen auch hierfür ein konkretes Beispiel diskutieren.

Beispiel 3.2.1. Wir betrachten einen BCH-Code der Länge $n = 15$ und der Dimension $k = 9$ im engeren Sinne zum Entwurfsabstand $\delta = 7$ über dem

Körper $\mathbb{F}_{2^4} = \mathbb{F}_2[T]/(T^4+T^3+1)$, der vom Generatorpolynom $g(X) = X^{10}+X^9+X^8+X^6+X^5+X^2+1$ erzeugt wird. Nach Satz 3.2.1 ist der zugehörige rationale Goppa-Code dann $C_{\mathcal{L}}(D_\zeta, 8\cdot\infty)$. Der Funktionenraum $\mathcal{L}(8\cdot\infty)$ hat $1, X, \ldots, X^8$ als Basis; die Stellen P_i des Divisors D_ζ identifizieren wir mit $X-\zeta^{i-1}$. Die Matrix

$$G_{BCH(15,9,7)} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \zeta^1 & \cdots & \zeta^{14} \\ \vdots & \vdots & & \vdots \\ 1 & \zeta^{(1\cdot 8)} & \cdots & \zeta^{(14\cdot 8)} \end{pmatrix}$$

ist eine Generatormatrix des Codes als rationaler Goppa-Code und in Standardform

$$G'_{BCH(15,9,7)} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&\zeta^6&\zeta^{11}&\zeta^7&\zeta^2&\zeta^{15}&\zeta^{12} \\ 0&1&0&0&0&0&0&0&0&\zeta^3&1&\zeta^2&\zeta^5&\zeta^7&\zeta^2 \\ 0&0&1&0&0&0&0&0&0&\zeta^8&\zeta^8&\zeta^2&\zeta^{11}&\zeta^6&\zeta^5 \\ 0&0&0&1&0&0&0&0&0&\zeta^{11}&\zeta^{14}&\zeta^{11}&\zeta^{12}&\zeta^{13}&\zeta^5 \\ 0&0&0&0&1&0&0&0&0&\zeta^{11}&\zeta^6&\zeta^6&\zeta^{10}&\zeta^3&\zeta \\ 0&0&0&0&0&1&0&0&0&\zeta^7&\zeta^8&1&\zeta^7&\zeta^3&\zeta^8 \\ 0&0&0&0&0&0&1&0&0&\zeta^{14}&\zeta^8&\zeta^6&\zeta^5&\zeta^4&\zeta^{12} \\ 0&0&0&0&0&0&0&1&0&\zeta^3&\zeta&\zeta^7&\zeta^{12}&\zeta^3&\zeta^{14} \\ 0&0&0&0&0&0&0&0&1&\zeta^5&\zeta&\zeta^{11}&\zeta^9&\zeta^6&\zeta^9 \end{pmatrix}.$$

Die zugehörige transponierte Kontrollmatrix ist eine Generatormatrix für den dualen Residuen-Code. Wenden wir das in Bemerkung 2.3.1 beschriebene Verfahren an, so erhalten wir

$$H^T_{BCH(15,9,7)} = \begin{pmatrix} \zeta^6&\zeta^3&\zeta^8&\zeta^{11}&\zeta^{11}&\zeta^7&\zeta^{14}&\zeta^3&\zeta^5&1&0&0&0&0&0 \\ \zeta^{11}&1&\zeta^8&\zeta^{14}&\zeta^6&\zeta^8&\zeta^8&\zeta&\zeta&0&1&0&0&0&0 \\ \zeta^7&\zeta^2&\zeta^2&\zeta^{11}&\zeta^6&1&\zeta^6&\zeta^7&\zeta^{11}&0&0&1&0&0&0 \\ \zeta^2&\zeta^5&\zeta^{11}&\zeta^{12}&\zeta^{10}&\zeta^7&\zeta^5&\zeta^{12}&\zeta^9&0&0&0&1&0&0 \\ 1&\zeta^7&\zeta^6&\zeta^{13}&\zeta^3&\zeta^3&\zeta^4&\zeta^3&\zeta^6&0&0&0&0&1&0 \\ \zeta^{12}&\zeta^2&\zeta^5&\zeta^5&\zeta&\zeta^8&\zeta^{12}&\zeta^{14}&\zeta^9&0&0&0&0&0&1 \end{pmatrix}.$$

Aus dieser Matrix lassen sich nun unmittelbar die Kontrollfunktionen ablesen, d.h. die Funktionen, deren Residuen an den Stellen $P_1, \ldots, P_{15}$ die Zeilen der letzten Matrix bilden.

$$f_1 = \frac{\zeta^6}{X-1} + \frac{\zeta^3}{X-\zeta} + \frac{\zeta^8}{X-\zeta^2} + \frac{\zeta^{11}}{X-\zeta^3} + \frac{\zeta^{11}}{X-\zeta^4} + \frac{\zeta^7}{X-\zeta^5} + \frac{\zeta^{14}}{X-\zeta^6} + \frac{\zeta^3}{X-\zeta^7} + \frac{\zeta^5}{X-\zeta^8} + \frac{1}{X-\zeta^9}$$

$$f_2 = \frac{\zeta^{11}}{X-1} + \frac{\zeta^{15}}{X-\zeta} + \frac{\zeta^8}{X-\zeta^2} + \frac{\zeta^{14}}{X-\zeta^3} + \frac{\zeta^6}{X-\zeta^4} + \frac{\zeta^8}{X-\zeta^5} + \frac{\zeta^8}{X-\zeta^6} + \frac{\zeta}{X-\zeta^7} + \frac{\zeta}{X-\zeta^8} + \frac{1}{X-\zeta^{10}}$$

$$f_3 = \frac{\zeta^7}{X-1} + \frac{\zeta^2}{X-\zeta} + \frac{\zeta^2}{X-\zeta^2} + \frac{\zeta^{11}}{X-\zeta^3} + \frac{\zeta^6}{X-\zeta^4} + \frac{1}{X-\zeta^5} + \frac{\zeta^6}{X-\zeta^6} + \frac{\zeta^7}{X-\zeta^7} + \frac{\zeta^{11}}{X-\zeta^8} + \frac{1}{X-\zeta^{11}}$$
$$f_4 = \frac{\zeta^2}{X-1} + \frac{\zeta^5}{X-\zeta} + \frac{\zeta^{11}}{X-\zeta^2} + \frac{\zeta^{12}}{X-\zeta^3} + \frac{\zeta^{10}}{X-\zeta^4} + \frac{\zeta^7}{X-\zeta^5} + \frac{\zeta^5}{X-\zeta^6} + \frac{\zeta^{12}}{X-\zeta^7} + \frac{\zeta^9}{X-\zeta^8} + \frac{1}{X-\zeta^{12}}$$
$$f_5 = \frac{1}{X-1} + \frac{\zeta^7}{X-\zeta} + \frac{\zeta^6}{X-\zeta^2} + \frac{\zeta^{13}}{X-\zeta^3} + \frac{\zeta^3}{X-\zeta^4} + \frac{\zeta^3}{X-\zeta^5} + \frac{\zeta^4}{X-\zeta^6} + \frac{\zeta^3}{X-\zeta^7} + \frac{\zeta^6}{X-\zeta^8} + \frac{1}{X-\zeta^{13}}$$
$$f_6 = \frac{\zeta^{12}}{X-1} + \frac{\zeta^2}{X-\zeta} + \frac{\zeta^5}{X-\zeta^2} + \frac{\zeta^5}{X-\zeta^3} + \frac{\zeta}{X-\zeta^4} + \frac{\zeta^8}{X-\zeta^5} + \frac{\zeta^{12}}{X-\zeta^6} + \frac{\zeta^{14}}{X-\zeta^7} + \frac{\zeta^9}{X-\zeta^8} + \frac{1}{X-\zeta^{14}}$$

3.3 Vergleich und Diskussion

Um die Stärken und Schwächen unterschiedlicher Codes diskutieren zu können, müssen wir Vergleichskriterien festlegen. Von Bedeutung sind vor allem die Informationsrate $\frac{k}{n}$ und die Wahrscheinlichkeit des Auftretens eines nichtkorrigierbaren Fehlers bei gegebener Fehlerhäufigkeit des Übertragungskanals p_e. Wir wollen in einem konkreten Beispiel einen geometrischen Goppa-Code mit einem Reed-Solomon-Code vergleichen.

Beispiel 3.3.1. Wir fixieren als Alphabet bzw. Zeichenvorrat für beide Codes den Körper $\mathbb{F}_{16} = \mathbb{F}_2^4$. Wir betrachten einen 4-fehlerkorrigierenden Reed-Solomon-Code des Typs $(16, 8, 9)$. Er entsteht aus einem $(15, 8, 8)$-RS-Code durch Hinzufügen eines zusätzlichen Paritätsprüfbits. Die Informationsrate ist $\frac{k}{n} = \frac{8}{16} = \frac{1}{2}$. Die Wahrscheinlichkeit, dass höchstens $e = \left[\frac{d-1}{2}\right]$ Fehler auftreten beträgt

$$p(\#M \leq e) = \sum_{i=0}^{e} \binom{n}{i} \cdot p_e^i \cdot (1-p_e)^{n-i}.$$

Für $p_e = 0,02$ errechnet man daraus eine Wahrscheinlichkeit für mehr als 4 Fehler von ca. $1 \cdot 10^{-5}$. Im Vergleich dazu betrachten wir einen geometrischen Goppa-Code über der projektiven Geraden. Er habe die Länge $n = 16$ und den Minimalabstand $d = 9$. Für die Dimension des Codes gilt dann $k = 8$ und damit $\frac{k}{n} = \frac{8}{16} = \frac{1}{2}$, wie oben. Auch die Wahrscheinlichkeiten ändern sich nicht, da beide Codes in d und damit in e übereinstimmen. Somit ist bei dieser Betrachtungsweise kein Vor- oder Nachteil erkennbar.

Leider haben wir uns in unseren Betrachtungen nur auf geometrische Codes über der projektiven Geraden beschränkt, ihre volle Leistungsfähigkeit entfalten diese aber erst über anspruchsvolleren algebraischen Kurven (Geschlecht $g > 0$) mit vielen rationalen Punkten, etwa der Klein-Quartik (vgl. [Pre98], S. 69) oder über Hermite-Kurven (vgl. [van99], S. 163). Das Geschlecht der Kurven geht nämlich in die Parameter des Codes ein und kann dadurch einen Vorteil gegenüber den klassischen Codes produzieren.

Anhang A

Codierung und Decodierung mit Maple V

Das Computer-Algebra-System Maple V (Release 5) ermöglicht es Codierung und Decodierung, sowie Fehlererkennung und -korrektur praktisch umzusetzen. Wir haben einige ausgewählte Beispiele so programmiert, dass bestimmte Parameter (Fehlerposition, Fehlergröße, Information) verändert werden können und dadurch ein Einblick in die konkrete Arbeitsweise der zuvor theoretisch behandelten Algorithmen gewährt wird. Die Dateien sind auf der beiliegenden CD gespeichert.

A.1 Fehlerkorrektur beim BCH-Code

Auf den folgenden drei Seiten ist der Quelltext der Datei `BCH-Code.mws` angegeben.

A.2 SV-Fehlerkorrektur

Die nächsten fünf Seiten enthalten den Quelltext der Datei `Goppa-Code.mws`.

A.3 Generatormatrix des CD-Brenncodes als rationaler Goppa-Code und Kontrollfunktionen

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

$$
\begin{array}{cccccccccccccccccccccccccc}
\zeta^{21} & \zeta^{9} & \zeta^{209} & \zeta^{117} & \zeta^{137} & \zeta^{124} & \zeta^{198} & \zeta^{56} & \zeta^{81} & \zeta^{231} & \zeta^{25} & \zeta^{6} & \zeta^{238} & \zeta^{238} & \zeta^{234} & \zeta^{162} & \zeta^{244} & \zeta^{196} & \zeta^{60} & \zeta^{218} & \zeta^{208} & \zeta^{165} & \zeta^{234} & \zeta^{11} \\
\zeta^{32} & \zeta^{119} & \zeta^{233} & \zeta^{219} & \zeta^{104} & \zeta^{78} & \zeta^{39} & \zeta^{146} & \zeta^{122} & \zeta^{214} & \zeta^{159} & \zeta^{44} & \zeta^{49} & \zeta^{50} & \zeta^{141} & \zeta^{187} & \zeta^{182} & \zeta^{127} & \zeta^{112} & \zeta^{205} & \zeta^{62} & \zeta^{29} & \zeta^{27} & \zeta^{222} \\
\zeta^{243} & \zeta^{220} & \zeta^{178} & \zeta^{78} & \zeta^{41} & \zeta^{135} & \zeta^{83} & \zeta^{77} & \zeta^{47} & \zeta^{90} & \zeta^{232} & \zeta^{13} & \zeta^{177} & \zeta^{206} & \zeta^{43} & \zeta^{184} & \zeta^{42} & \zeta^{155} & \zeta^{133} & \zeta^{92} & \zeta^{139} & \zeta^{228} & \zeta^{236} & \zeta^{105} \\
\zeta^{126} & \zeta^{89} & \zeta^{192} & \zeta^{191} & \zeta^{68} & \zeta^{240} & \zeta^{53} & \zeta^{34} & \zeta^{146} & \zeta^{183} & \zeta^{21} & \zeta^{254} & \zeta^{59} & \zeta^{247} & \zeta^{112} & \zeta^{254} & \zeta^{207} & \zeta^{183} & \zeta^{74} & \zeta^{26} & \zeta^{194} & \zeta^{218} & \zeta^{93} & \zeta^{227}
\end{array}
$$

$$
\begin{aligned}
\omega_1 := {} & \frac{\zeta^{21}}{X-\zeta} + \frac{\zeta^{9}}{X-\zeta^{2}} + \frac{\zeta^{209}}{X-\zeta^{3}} + \frac{\zeta^{117}}{X-\zeta^{4}} + \frac{\zeta^{137}}{X-\zeta^{5}} + \frac{\zeta^{124}}{X-\zeta^{6}} \\
& + \frac{\zeta^{198}}{X-\zeta^{7}} + \frac{\zeta^{56}}{X-\zeta^{8}} + \frac{\zeta^{81}}{X-\zeta^{9}} + \frac{\zeta^{231}}{X-\zeta^{10}} + \frac{\zeta^{25}}{X-\zeta^{11}} + \frac{\zeta^{6}}{X-\zeta^{12}} \\
& + \frac{\zeta^{238}}{X-\zeta^{13}} + \frac{\zeta^{238}}{X-\zeta^{14}} + \frac{\zeta^{234}}{X-\zeta^{15}} + \frac{\zeta^{162}}{X-\zeta^{16}} + \frac{\zeta^{244}}{X-\zeta^{17}} + \frac{\zeta^{196}}{X-\zeta^{18}} \\
& + \frac{\zeta^{60}}{X-\zeta^{19}} + \frac{\zeta^{218}}{X-\zeta^{20}} + \frac{\zeta^{208}}{X-\zeta^{21}} + \frac{\zeta^{165}}{X-\zeta^{22}} + \frac{\zeta^{234}}{X-\zeta^{23}} + \frac{\zeta^{11}}{X-\zeta^{24}} + \frac{\zeta^{255}}{X-\zeta^{25}} \\
\omega_2 := {} & \frac{\zeta^{32}}{X-\zeta} + \frac{\zeta^{119}}{X-\zeta^{2}} + \frac{\zeta^{233}}{X-\zeta^{3}} + \frac{\zeta^{219}}{X-\zeta^{4}} + \frac{\zeta^{104}}{X-\zeta^{5}} + \frac{\zeta^{78}}{X-\zeta^{6}} \\
& + \frac{\zeta^{39}}{X-\zeta^{7}} + \frac{\zeta^{146}}{X-\zeta^{8}} + \frac{\zeta^{122}}{X-\zeta^{9}} + \frac{\zeta^{214}}{X-\zeta^{10}} + \frac{\zeta^{159}}{X-\zeta^{11}} + \frac{\zeta^{44}}{X-\zeta^{12}} \\
& + \frac{\zeta^{49}}{X-\zeta^{13}} + \frac{\zeta^{50}}{X-\zeta^{14}} + \frac{\zeta^{141}}{X-\zeta^{15}} + \frac{\zeta^{187}}{X-\zeta^{16}} + \frac{\zeta^{182}}{X-\zeta^{17}} + \frac{\zeta^{127}}{X-\zeta^{18}} \\
& + \frac{\zeta^{112}}{X-\zeta^{19}} + \frac{\zeta^{205}}{X-\zeta^{20}} + \frac{\zeta^{62}}{X-\zeta^{21}} + \frac{\zeta^{29}}{X-\zeta^{22}} + \frac{\zeta^{27}}{X-\zeta^{23}} + \frac{\zeta^{222}}{X-\zeta^{24}} + \frac{\zeta^{255}}{X-\zeta^{26}} \\
\omega_3 := {} & \frac{\zeta^{243}}{X-\zeta} + \frac{\zeta^{220}}{X-\zeta^{2}} + \frac{\zeta^{178}}{X-\zeta^{3}} + \frac{\zeta^{78}}{X-\zeta^{4}} + \frac{\zeta^{41}}{X-\zeta^{5}} + \frac{\zeta^{135}}{X-\zeta^{6}} \\
& + \frac{\zeta^{83}}{X-\zeta^{7}} + \frac{\zeta^{77}}{X-\zeta^{8}} + \frac{\zeta^{47}}{X-\zeta^{9}} + \frac{\zeta^{90}}{X-\zeta^{10}} + \frac{\zeta^{232}}{X-\zeta^{11}} + \frac{\zeta^{13}}{X-\zeta^{12}} \\
& + \frac{\zeta^{177}}{X-\zeta^{13}} + \frac{\zeta^{206}}{X-\zeta^{14}} + \frac{\zeta^{43}}{X-\zeta^{15}} + \frac{\zeta^{184}}{X-\zeta^{16}} + \frac{\zeta^{42}}{X-\zeta^{17}} + \frac{\zeta^{155}}{X-\zeta^{18}} \\
& + \frac{\zeta^{133}}{X-\zeta^{19}} + \frac{\zeta^{92}}{X-\zeta^{20}} + \frac{\zeta^{139}}{X-\zeta^{21}} + \frac{\zeta^{228}}{X-\zeta^{22}} + \frac{\zeta^{236}}{X-\zeta^{23}} + \frac{\zeta^{105}}{X-\zeta^{24}} + \frac{\zeta^{255}}{X-\zeta^{27}} \\
\omega_4 := {} & \frac{\zeta^{126}}{X-\zeta} + \frac{\zeta^{89}}{X-\zeta^{2}} + \frac{\zeta^{192}}{X-\zeta^{3}} + \frac{\zeta^{191}}{X-\zeta^{4}} + \frac{\zeta^{68}}{X-\zeta^{5}} + \frac{\zeta^{240}}{X-\zeta^{6}} \\
& + \frac{\zeta^{53}}{X-\zeta^{7}} + \frac{\zeta^{34}}{X-\zeta^{8}} + \frac{\zeta^{146}}{X-\zeta^{9}} + \frac{\zeta^{183}}{X-\zeta^{10}} + \frac{\zeta^{21}}{X-\zeta^{11}} + \frac{\zeta^{254}}{X-\zeta^{12}} \\
& + \frac{\zeta^{59}}{X-\zeta^{13}} + \frac{\zeta^{247}}{X-\zeta^{14}} + \frac{\zeta^{112}}{X-\zeta^{15}} + \frac{\zeta^{254}}{X-\zeta^{16}} + \frac{\zeta^{207}}{X-\zeta^{17}} + \frac{\zeta^{183}}{X-\zeta^{18}} \\
& + \frac{\zeta^{74}}{X-\zeta^{19}} + \frac{\zeta^{26}}{X-\zeta^{20}} + \frac{\zeta^{194}}{X-\zeta^{21}} + \frac{\zeta^{218}}{X-\zeta^{22}} + \frac{\zeta^{93}}{X-\zeta^{23}} + \frac{\zeta^{227}}{X-\zeta^{24}} + \frac{\zeta^{255}}{X-\zeta^{28}}
\end{aligned}
$$

Literaturverzeichnis

[Bos01] Bosch, S.: *Algebra*. Springer Verlag, 2001.

[Ham87] Hamming, R. W.: *Information und Codierung*. VCH Verlagsgesellschaft Weinheim, 1987.

[Hei95] Heise, W. und Quattrocchi, P.: *Informations- und Codierungstheorie*. Springer Verlag, 1995.

[Hol99] Holzapfel, R.-P.: *Einführung in die algebraische Codierungstheorie*. Vorlesung vom Sommersemester, Humboldt-Universität zu Berlin, 1999.

[Kai00] Kaibel, V.: *Kodierungstheorie*. Vorlesung vom Wintersemester, Technische Universität Berlin, 1999/2000.

[Pre98] Pretzel, O.: *Codes and Algebraic Curves*. Oxford University Press, 1998.

[Sti93] Stichtenoth, H.: *Algebraic function fields and codes*. Springer Verlag, 1993.

[van99] van Lint, J. H.: *Introduction to Coding Theory*. Springer Verlag, 1999.

Erklärung

Ich versichere, dass ich die vorliegende Hausarbeit über das Thema „Beschreibung klassischer Codes als Goppa-Codes“ in der gesetzten Frist selbständig verfasst und keine anderen Hilfsmittel als die angegebenen verwendet habe. Alle Stellen der Arbeit, die anderen Werken wörtlich oder sinngemäß entnommen sind, sind unter Angabe der Quelle als Entlehnung kenntlich gemacht. Die Zeichnungen, Kartenskizzen, bildlichen Darstellungen, Statistiken und musik. Notenbeispiele sind von mir verfasst, soweit nicht als Entlehnung gekennzeichnet.

Berlin, den 23. März 2004