On the efficient decoding of algebraic-geometric codes

Ruud Pellikaan*

Appeared in *Eurocode 92*, (P.Camion, P. Charpin and S. Harari eds.), Udine, CISM Courses and Lectures **339**, Springer, Wien, 1993, pp. 231–253.

Abstract

This talk is intended to give a survey on the existing literature on the decoding of algebraic-geometric codes. Although the motivation originally was to find an efficient decoding algorithm for algebraic-geometric codes, the latest results give algorithms which can be explained purely in terms of linear algebra. We will treat the following subjects:

1. The decoding problem
2. Decoding by error location
3. Decoding by error location of algebraic-geometric codes
4. Majority coset decoding
5. Decoding algebraic-geometric codes by solving the key equation
6. Improvements of the complexity

*Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

1 The decoding problem

Minimum distance decoding (MDD), also called *maximum likelihood decoding* (MLD) under some assumptions on the channel used, is an algorithm which has as input $(C, \mathbf{y})$, where C is a linear code, say given by a generator matrix and $\mathbf{y}$ a word of the same length as C and as output a word $\mathbf{c} \in C$, such that the distance between $\mathbf{y}$ and $\mathbf{c}$ is equal to the distance between $\mathbf{y}$ and C. It is shown by Berlekamp, McEliece and van Tilborg [3] that this problem is NP complete. One may object that this is not a relevant problem in practice, that is usually one knows a great deal in advance of the code C, or one has a lot of time to find the decoding algorithm itself, the essential point in practice is that with input $\mathbf{y}$ one wants to find (one of) the nearest codewords in a short time. So one considers *minimum distance decoding with preprocessing* (MDDP), but also this problem turns out to be hard, see Bruck and Naor [5]. If C is a linear code of minimum distance d and $\mathbf{y}$ a word of distance at most $\lfloor (d-1)/2 \rfloor$ to C, then the nearest codeword is unique. *Decoding up to half the minimum distance with preprocessing*, DHDP for short, has as input a word $\mathbf{y}$ and as output the nearest codeword in case $d(\mathbf{y}, C) \leq (d-1)/2$, and "the received word has more than $\lfloor (d-1)/2 \rfloor$ errors" otherwise. As far as I know it is not known how difficult DHDP is. In the sequel we show that a lot of codes, in particular algebraic-geometric codes, have an efficient algorithm which decodes up to half the minimum distance.

2 Decoding by error location

We have the standard bilinear form defined by $< \mathbf{a}, \mathbf{b} >= \sum_i a_i b_i$. If A is a subset of $\mathbf{F}_q^n$, then we define the dual $A^\perp$ of A in $\mathbf{F}_q^n$ with respect to the standard bilinear form by $A^\perp = \{\mathbf{b} | < \mathbf{a}, \mathbf{b} >= 0 \text{ for all } \mathbf{a} \in A\}$. So in this definition A need not to be linear but $A^\perp$ is always linear.

Let C be a linear code in $\mathbf{F}_q^n$. Suppose we have received a word $\mathbf{y}$ and we have some *erasures*, that is $\mathbf{y} = \mathbf{e} + \mathbf{c}$ for some error $\mathbf{e}$ and codeword $\mathbf{c} \in C$, and we know a set J with at most $d(C) - 1$ elements which contains the set I of *error positions*, that is $I = \{i | e_i \neq 0\}$. Then we can find the error values as follows. Let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be a basis of $C^\perp$. Consider the following linear equations

$$< \mathbf{x}, \mathbf{v}_i >=< \mathbf{y}, \mathbf{v}_i > \quad \text{for} \quad i = 1, \ldots, r,$$

$$x_j = 0 \quad \text{for } j \notin J.$$

Then $\mathbf{e}$ is the unique solution. It is clear that the error vector is a solution. If $\mathbf{x}$ is another solution, then $\mathbf{x} - \mathbf{e}$ has zero inner product with all the elements of $C^\perp$, so it is an element of C, moreover its weight is at most $|J| \leq d(C) - 1$, so it must be zero,

that is to say $\mathbf{x} = \mathbf{e}$. Thus we have shown that we can reduce error decoding to erasure decoding. We still have to find the error positions.

But if we want to decode all words with t errors, then there are $\binom{n}{t}$ possible t sets one has to consider, and this number grows exponentially with n. So we are looking for an efficient way to find the error positions. First we introduce some notation. We define the star multiplication $\mathbf{a} * \mathbf{b}$ of two elements $\mathbf{a}$ and $\mathbf{b}$ of $\mathbf{F}_q^n$ by coordinatewise multiplication, that is $(\mathbf{a} * \mathbf{b})_i = a_i b_i$. For two subsets A and B of $\mathbf{F}_q^n$ we denote the set $\{\mathbf{a} * \mathbf{b} | \mathbf{a} \in A, \mathbf{b} \in B\}$ by $A * B$. The following definition of a t-error correcting pair comes out of the blue for those who have never seen it before. But we shall explain why it works and afterwards show the ubiquity of error correcting pairs and that the examples are abundant.

We will follow our paper [41], an earlier version [39] was never published. The same kind of reasoning was found independently by Kötter [29].

Definition 2.1 Let C be a $\mathbf{F}_q$-linear code of length n. Let A and B be $\mathbf{F}_{q^e}$-linear codes of length n, then (A, B) is called a *t-error correcting pair* for C over $\mathbf{F}_{q^e}$, if

1. $C \subseteq (A * B)^\perp$
2. $k(A) > t$
3. $d(B^\perp) > t$
4. $d(A) + d(C) > n$

Remember that we have defined the dual for any subset of $\mathbf{F}_{q^e}^n$, so $(A * B)^\perp$ is well defined and remark that C is a subspace of $\mathbf{F}_q^n$, so we can write (1') $A * B \subseteq C^\perp$ instead of (1), if we mean by $C^\perp$ the dual of the subset C in $\mathbf{F}_{q^e}^n$ and not in $\mathbf{F}_q^n$.

Suppose (A, B) is a t-error correcting pair for C. We first define the vector space $K_{\mathbf{y}}$ of a received word $\mathbf{y}$, by

$$K_{\mathbf{y}} = \{\mathbf{a} \in A | < \mathbf{y}, \mathbf{a} * \mathbf{b} >= 0 \quad \text{for all} \quad \mathbf{b} \in B\}.$$

If $A * B \subseteq C^\perp$ and $\mathbf{y}$ is a word with error $\mathbf{e}$, then $K_{\mathbf{y}} = K_{\mathbf{e}}$. Remark that an element which is zero at the error positions is contained in $K_{\mathbf{y}}$. In more formal terms. Suppose J is a subset of $\{1, \ldots, n\}$. Define

$$A(J) = \{\mathbf{a} \in A | a_j = 0 \text{ for all } j \in J\}.$$

If $I = supp(\mathbf{e})$, then

$$A(I) \subseteq K_{\mathbf{y}}.$$

Because

$$< \mathbf{y}, \mathbf{a} * \mathbf{b} >=< \mathbf{e}, \mathbf{a} * \mathbf{b} >= \sum_{i \in I} a_i b_i e_i = 0,$$

since $a_i = 0$ for all $i \in I = supp(\mathbf{e})$, if $\mathbf{a} \in A(I)$. So this explains condition (1). Now assume $wt(\mathbf{e}) \leq t$. We need a nonzero element of $K_{\mathbf{y}}$, and the existence of such an element is ensured by assuming that the dimension of A is at least $t+1$, since $K_{\mathbf{y}}$ contains $A(I)$, which is the intersection of t subspaces of codimension 1 and therefore not zero. So this explains condition (2). $K_{\mathbf{y}}$ is the object we know and $A(I)$ is the one we want to know, since it gives an indication where the error positions are located. Condition (3) guarantees that

$$K_{\mathbf{y}} = A(I).$$

We have seen already one inclusion, now suppose $\mathbf{a} \in K_{\mathbf{y}}$, then

$$0 =< \mathbf{y}, \mathbf{a} * \mathbf{b} >=< \mathbf{e}, \mathbf{a} * \mathbf{b} >=< \mathbf{e} * \mathbf{a}, \mathbf{b} >,$$

for all $\mathbf{b} \in B$, so $\mathbf{e} * \mathbf{a} \in B^{\perp}$. But $wt(\mathbf{e} * \mathbf{a}) \leq wt(\mathbf{e}) \leq t < d(B^{\perp})$, hence $\mathbf{e} * \mathbf{a} = 0$, so a_i is zero at all places where e_i is not zero, that is at all $i \in I$, thus $\mathbf{a} \in A(I)$. So we compute $K_{\mathbf{y}}$ and take a nonzero element $\mathbf{a}$ in it; if $K_{\mathbf{y}}$ would be zero, then the received word has more than t errors. We search for the set of zeros of $\mathbf{a}$ and call it J. By the above we know that J contains I and condition (4) implies that J has at most $d(C) - 1$ elements, thus we can find the error values by solving the above mentioned linear equations over the subfield $\mathbf{F}_q$, which has a unique solution. In the algorithm we have to find $K_{\mathbf{y}}$ which is the zero space of a set of linear equations over $\mathbf{F}_{q^e}$ and to find the error values which is done by solving linear equations over $\mathbf{F}_q$, thus the complexity of the algorithm is at most of the order $\mathcal{O}((ne)^3)$, since a multiplication in $\mathbf{F}_{q^e}$ is of the order of e^3 multiplications in $\mathbf{F}_q$. Thus we have proved the following theorem.

Theorem 2.2 *If C is an $\mathbf{F}_q$-linear code of length n and (A, B) is a t-error correcting pair for C over $\mathbf{F}_{q^e}$, then one can correct all words with at most t errors with complexity $\mathcal{O}((ne)^3)$.*

Example 2.3 Reed-Solomon codes, see [34].
Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ be an n-tuple of n distinct elements of $\mathbf{F}_q$. Let

$$RS_k(\alpha) = \{(f(\alpha_1), \ldots, f(\alpha_n)) | f \in \mathbf{F}_q[x], \deg(f) < k\}.$$

Then $RS_k(\alpha)$ is an $[n, k, n-k+1]$ code, so it is an MDS code and its dual is an $[n, n-k, k+1]$ code. Let $A = RS_{t+1}(\alpha)$ and $B = RS_t(\alpha)$ and $C = RS_{2t}(\alpha)^{\perp}$. Then C has minimum distance $2t+1$. Now A is an $[n, t+1, n-t]$ code, hence $k(A) = t+1$ and $d(A) + d(C) = (n-t) + (2t+1) > n$. Furthermore $d(B^{\perp}) = t+1$. Finally, if $\mathbf{a} \in A$ and $\mathbf{b} \in B$, then there are polynomials f and h of degree at most t and $t-1$,

respectively, such that $f(\alpha_i) = a_i$ and $h(\alpha_i) = b_i$ for all i. Now fh has degree at most $2t-1$ and $a_i b_i = fh(\alpha_i)$, so $\mathbf{a} * \mathbf{b} \in C^\perp$. Thus $A * B \subseteq C^\perp$ and therefore (A, B) is a t-error correcting pair for C.

Example 2.4 BCH codes.
Let α be a primitive element of $\mathbf{F}_{q^e}$ and let $n = q^e - 1$. Let C be the BCH code with the defining set of zeros $\{0, 1, \ldots, \delta - 2\}$, that is to say C is the set of all $(c_0, \ldots, c_{n-1}) \in \mathbf{F}_q^n$ such that

$$\sum_{j=0}^{n-1} c_j \alpha^{ij} = 0 \quad \text{for all} \quad 0 \leq i \leq \delta - 2.$$

Then C is an $\mathbf{F}_q$-linear code of length n and designed minimum distance δ. Let $A = RS_{t+1}(1, \alpha, \ldots, \alpha^{n-1}))$ and $B = RS_t(1, \alpha, \ldots, \alpha^{n-1})$, where $t = \lfloor (\delta - 1)/2 \rfloor$. Then A and B are $\mathbf{F}_{q^e}$-linear codes of length n and (A, B) is a t-error correcting pair. This is seen as follows. Let $\mathbf{v}_i = (1, \alpha^i, \ldots, \alpha^{i(n-1)})$ for $0 \leq i \leq n-1$. Then $\mathbf{v}_0, \ldots, \mathbf{v}_t$ is a basis for A and $\mathbf{v}_0, \ldots, \mathbf{v}_{t-1}$ is a basis for B and $\mathbf{v}_i * \mathbf{v}_j = \mathbf{v}_{i+j}$, so the vector space generated by $A * B$ has basis $\mathbf{v}_0, \ldots, \mathbf{v}_{2t-1}$, and $2t - 1 \leq \delta - 2$, so $C \subseteq (A * B)^\perp$. The other 3 conditions hold in the same way as in the previous example. With respect to the above bases of A and B the vector space $K_{\mathbf{y}}$ is the nullspace of the matrix

$$\begin{pmatrix} S_0 & S_1 & S_2 & \ldots & S_t \\ S_1 & S_2 & S_3 & \ldots & S_{t+1} \\ S_2 & S_3 & S_4 & \ldots & S_{t+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_{t-1} & S_t & S_{t+1} & \ldots & S_{2t-1} \end{pmatrix},$$

where S_i is the syndrome $< \mathbf{y}, \mathbf{v}_i >$.

This line of reasoning for BCH codes was done for the first time by Peterson [43] in 1960 and independently by Arimoto [1] in 1961. Later improvements were found on the minimum distance of cyclic codes, known as the Hartmann-Tzeng, Roos and van Lint-Wilson bounds, see [35]. Feng and Tzeng [18, 19] and Bours et al. [4] gave algorithms decoding up to half the Hartman-Tzeng and in many cases of the Roos designed minimum distance. Duursma and Kötter [12] could explain the just mentioned algorithms by giving error correcting pairs for these cyclic codes, morover they gave error correcting pairs for all but four cases of the table of van Lint and Wilson of all binary cyclic codes of length at most 62.

Example 2.5 Classical Goppa codes.
Let $l = \{\alpha_1, \ldots, \alpha_n\}$ be an enumeration of l, a subset of $\mathbf{F}_{q^e}$ of n elements. Let h be

a polynomial in one variable X with coefficients in $\mathbf{F}_{q^e}$ such that its set of zeros is disjoint from l. Let $\Gamma(l,h)$ be the $\mathbf{F}_q$-linear code of all elements $\mathbf{c} \in \mathbf{F}_q^n$ such that

$$\sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} \text{ is divisible by } h.$$

It is wellknown [37, 12.3] that $\Gamma(l,h)$ has parity check matrix

$$\begin{pmatrix} h(\alpha_1)^{-1} & \dots & h(\alpha_n)^{-1} \\ \alpha_1 h(\alpha_1)^{-1} & \dots & \alpha_n h(\alpha_n)^{-1} \\ \vdots & \ddots & \vdots \\ \alpha_1^{r-1} h(\alpha_1)^{-1} & \dots & \alpha_n^{r-1} h(\alpha_n)^{-1} \end{pmatrix},$$

where r is the degree of the Goppa polynomial h. Then this code has designed minimum distance $r+1$. So if we take $n = q^m - 1$ and $\alpha_i = \alpha^i$ for $i = 0, \dots, n-1$ for some primitive element α of $\mathbf{F}_{q^e}$ and $h = X^{\delta-1}$, then we are back in example 2.4 of the BCH codes. In order to define the pair (A,B) we have to introduce Generalized Reed-Solomon codes. Let $\alpha = (\alpha_1, \dots, \alpha_n)$ be an n-tuple of n distinct elements of $\mathbf{F}_{q^e}$. Let $\mathbf{y}$ be an n-tuple of nonzero elements of $\mathbf{F}_{q^e}$. Define the k-dimensional Generalized Reed-Solomon code over $\mathbf{F}_{q^e}$ by

$$GRS_k(\alpha, \mathbf{y}) = \{(f(\alpha_1)y_1, \dots, f(\alpha_n)y_n) | f \in \mathbf{F}_{q^e}[x], \deg(f) < k\}.$$

Now let $A = RS_{t+1}(\alpha)$ and $B = GRS_t(\alpha, \mathbf{y})$, where $t = \lfloor r/2 \rfloor$ and $y_i = h(\alpha_i)^{-1}$. Then (A,B) is a t-error correcting pair for $\Gamma(l,h)$. Another way to decode these classical Goppa codes is by means of solving the so called key equation with Euclid's algorithm or the Berlekamp-Massey algorithm [2, 36] in the ring of polynomials $\mathbf{F}_{q^e}[X]$. We return to that in section 5.

An important class of codes which can be decoded by error location are algebraic-geometric codes which we will discuss in the next section.

3 Decoding by error location of algebraic-geometric codes

In this section we treat algebraic-geometric codes, also called geometric Goppa codes [20, 21, 22, 23], which can be considered as a generalization of Reed-Solomon codes on the affine line to codes on arbitrary curves. We will not discuss the importance of these codes nor give an exact definition of it and prefer to discuss it by analogy with Reed-Solomon and classical Goppa codes and by an example. The part needed to

prove the existence of asymptotically good codes with parameters above the Gilbert-Varshamov bound is very difficult and takes say 5 years to grasp for an outsider. On the other hand for the construction of geometric Goppa codes on explicit curves one needs only the theory of algebraic curves over finite fields, in particular the theorem of Riemann-Roch, and this part a layman can learn in half a year. There are some introductory works on the subject, see [24, 32, 33, 34]. For a selfcontained treatment on algebraic curves over finite fields we recommend the first chapters of the books of Chevalley [8] or Moreno [38]. This suffices to understand the papers concerning the decoding of algebraic-geometric codes. For a more thourough treatment of algebraic curves and their codes we mention the book of Tsfasman and Vlăduţ [59] and the forthcomming book of Stichtenoth [57].

The projective line over $\mathbf{F}_q$ has $q+1$ *points* which are defined over $\mathbf{F}_q$. The field of rational functions on the projective line is denoted by $\mathbf{F}_q(X)$, the elements are quotients of polynomials in $\mathbf{F}_q[X]$. The points of the projective line, not equal to the point at infinity correspond one to one with monic linear polynomials. *Places* are generalizations of points and places which are not the point at infinity correspond one to one with monic irreducible polynomials in $\mathbf{F}_q[X]$, and the degree of a place is the degree of the corresponding irreducible polynomial. Places of degree 1, not equal to the point at infinity, are exactly the points of the affine line. For every place P we introduce a *valuation* v_P at the place on the field of rational functions, which gives the order of zero or pole at these places, that is to say

$$v_P(f) = m \quad \text{if and only if} \quad f = p^m \frac{a}{b},$$

where p is the irreducible polynomial corresponding to the place P and a and b are polynomials not divisible by p. For the point at infinity one can make a coordinate transformation $Y = 1/X$ and consider the valuation defined by the irreducible polynomial Y or equivalently

$$v_\infty(\frac{a}{b}) = \deg(b) - \deg(a),$$

where a and b are polynomials in $\mathbf{F}_q[X]$.

We introduce *divisors* as a way of bookkeeping of the number of zeros and poles of functions and other objects. A divisor is a formal sum of places with integer coefficients, such that at most finitely many coefficients are nonzero. One can add divisors coefficientwise. The *support* of a divisor G is the set of places with a nonzero coefficient in G. One has a partial order on the divisors by comparing them coefficientwise. Thus, if $G = \sum m_P P$ and $H = \sum n_P P$, then $G + H = \sum (m_P + n_P)P$, and $G \leq H$ if and only if $m_P \leq n_P$ for all P. A divisor is called *effective* if all the coefficients are not negative. The degree of a divisor G of the form $\sum m_P P$ is by definition $\deg(G) = \sum m_P \deg(P)$. A

rational function f has a divisor (f), which we call the *principal divisor* of the function, as follows

$$(f) = \sum v_P(f)P.$$

The principal divisor of a rational function f is the difference of two effective divisors, $(f) = (f)_0 - (f)_\infty$, where $(f)_0$ is the *divisor of zeros* and $(f)_\infty$ is the *divisor of poles* of f. Every rational function f has a factorization $f = \prod q_i^{e_i}$, where q_i is an irreducible polynomial and e_i an integer. Let Q_i be the place corresponding to q_i and P_∞ the point at infinity. It is not difficult to see that $(f) = \sum e_i Q_i - e_i \deg(Q_i) P_\infty$. Thus $\deg((f)) = 0$.

For every divisor G one defines the vector space $L(G)$ of rational functions with zeros and poles prescribed by G as follows

$$L(G) = \{f \in \mathbf{F}_q(X) | f = 0 \text{ or } (f) \geq -G\}.$$

So, if $G = mP_\infty$, then $L(G)$ is the vector space of all polynomials of degree at most m, it has basis $1, X, \ldots, X^m$ and dimension $m+1$. Let h be a Goppa polynomial. Let $h = \prod q_i^{e_i}$ be the factorization of h, and Q_i the place corresponding to q_i. If $G = \sum e_i Q_i - P_\infty$, then $L(G)$ is the vector space of rational functions of the form a/h, where a is a polynomial of degree at most $r-1$ and r the degree of h, thus $L(G)$ has a basis of the form $1/h, x/h, \ldots, x^{r-1}/h$ and dimension r. In both examples we have that the dimension of $L(G)$ is equal to $\deg(G) + 1$, this is indeed always the case for the projective line.

If $P = (\alpha : 1)$ and G is a divisor which does not have P in its support and $f \in L(G)$, then $v_P(f) \geq 0$, so f has not a pole at P, therefore one can evaluate f at P and get $f(\alpha) \in \mathbf{F}_q$. If $l = \{\alpha_1, \ldots, \alpha_n\}$ are n elements in $\mathbf{F}_q$ and $P_i = (\alpha_i : 1)$, then $P_1, \ldots, P_n$ are n distinct points on the projective line, that is places of degree 1. We denote the divisor $P_1 + \cdots + P_n$ by D. Let G be a divisor with has disjoint support with D. Then we can evaluate f at all $P_i's$ and consider the map

$$\mathrm{ev}_D : L(G) \longrightarrow \mathbf{F}_q^n,$$

defined by $\mathrm{ev}_D(f) = (f(P_1), \ldots, f(P_n))$. This map is $\mathbf{F}_q$-linear and injective if $\deg(G) < n$. We denote the image of this map by $C_L(D, G)$ which is an $\mathbf{F}_q$-linear $[n, m+1, n-m]$ code, where $m = \deg(G)$. In case $G = (k-1)P_\infty$ we get the Reed-Solomon code $RS_k(\alpha)$. Now we consider the classical Goppa code with Goppa polynomial $h \in \mathbf{F}_q[X]$ and locator set $l = \{\alpha_1, \ldots, \alpha_n\}$ in $\mathbf{F}_q$. So we do not consider the subfield subcode case for simplicity, that is $e = 1$ in $\mathbf{F}_{q^e}$. Let $h = \prod q_i^{e_i}$ be the factorization of h and Q_i the place corresponding to q_i. Define the divisor $G = \sum e_i Q_i - P_\infty$. Then $C_L(D, G)$ is the dual of the classical Goppa code $\Gamma(l, g)$, since the above mentioned parity check matrix of $\Gamma(l, g)$ is a generator matrix of $C_L(D, G)$.

For an arbitrary algebraic curve $\mathcal{X}$ over the field $\mathbf{F}_q$ one can consider its function field $\mathbf{F}_q(\mathcal{X})$, which is an finite extension of $\mathbf{F}_q(X)$. Similarly to the case of the projective line one can define points, places, valuations, divisors, the principal divisor of a function and the vector space $L(G)$ of a divisor G and the code $C_L(D, G)$. We still have that the degree of a principal divisor is zero, but for the dimension of $L(G)$ we do not have a simple formula. The Theorem of Riemann says that there exists a nonnegative integer l such that for every divisor G

$$\dim(L(G)) \geq \deg(G) + 1 - l,$$

and the smallest nonnegative integer with this property is called the *genus*, and is denoted by $g(\mathcal{X})$ or g. Thus the genus of the projective line is zero. If moreover $\deg(G) > 2g - 2$, then $\dim(L(G)) = \deg(G) + 1 - g$. If $2g - 2 < \deg(G) < n$, then $C_L(D, G)$ has dimension $\deg(G) + 1 - g$ and minimum distance at least $n - \deg(G)$.

The minimum distance of the dual of a code can be estimated in terms of the code itself, since we have the following property which is easy to verify

$$d(C^{\perp}) > t \quad \text{if and only if } \dim(C(I)) = k - |I| \text{ for all } I \subseteq \{1, \ldots, n\} \text{ and } |I| \leq t,$$

where $C(I)$ is the subcode of words in C which are zero at all $i \in I$, as defined in section 1. If $Q = \sum_{i \in I} P_i$, then $C_L(D, G)(I) = C_L(D, G - Q)$. The dimension of $C_L(D, G - Q)$ is $\deg(Q) = |I|$ less than the dimension of $C_L(D, G)$, if $|I| < \deg(G) - (2g - 2)$. Thus the dual of $C_L(D, G)$ has at least minimum distance $\deg(G) - 2g + 2$. Another way to get an estimate for the minimum distance of the dual code of $C_L(D, G)$ is with the help of *differentials*. This can be explained by the classical Goppa codes too. Think of differentials as objects of the form $f dh$, where f and h are rational functions and dh is the differential of h, such that the map which sends h to dh is $\mathbf{F}_q$-linear and the Leibniz rule $d(fh) = f dh + h df$ holds. One can talk about zeros and poles of differentials. We denote the set of differentials on $\mathcal{X}$ by $\Omega_{\mathcal{X}}$. At every place P there exists a *local parameter* that is a function u which has valuation 1 at P, and for every differential ω there exists a function f such that $\omega = f du$. The valuation $v_P(\omega)$ of ω at P is now by definition $v_P(f)$, so we say ω has a zero of order m if $m = v_P(f) > 0$ and ω has a pole of order m if $m = -v_P(f) > 0$. The divisor (ω) of ω is by definition $(\omega) = \sum v_P(\omega) P$. The divisor of a differential is called *canonical* and has always degree $2g - 2$. In the same way as we defined $L(G)$ for functions we now define the vector space $\Omega(G)$ of differentials with zeros and poles prescribed by G as follows

$$\Omega(G) = \{\omega \in \Omega_{\mathcal{X}} | \omega = 0 \text{ or } (\omega) \geq G\}.$$

Now one could have defined the genus as the dimension of the vector space of differentials without poles, that is of $\Omega(O)$, where O is the divisor with coefficient 0 at every

place. The Theorem of *Riemann-Roch* states that

$$\dim(L(G)) = \deg(G) + 1 - g + \dim(\Omega(G)).$$

If moreover P is a place of degree 1 and u is a local parameter of P, then f has a formal Laurent series $\sum_{i=m}^{\infty} a_i u^i$, where the coefficients a_i are in $\mathbf{F}_q$ and $m = v_P(\omega)$ and $a_m \neq 0$. The entry a_{-1} is called the *residue* of ω and denoted by $\mathrm{res}_P(\omega)$. If we have n points on the curve, that is n places of degree 1, and a divisor G with none of the $P_i's$ in its support, then we consider the map

$$\mathrm{res}_D : \Omega(G - D) \longrightarrow \mathbf{F}_q^n,$$

defined by $\mathrm{res}_D(\omega) = (\mathrm{res}_{P_1}(\omega), \ldots, \mathrm{res}_{P_n}(\omega))$. Its image we denote by $C_\Omega(D, G)$. If $2g-2 < \deg(G) < n$, then $C_\Omega(D, G)$ is a $\mathbf{F}_q$-linear code of dimension $n - \deg(G) - 1 + g$ and minimum distance at least $\deg(G) - 2g + 2$. We call $\deg(G) - 2g + 2$ the *designed minimum distance* of the code $C_\Omega(D, G)$ and denote it by d^*. Both constructions, that is by evaluating functions or by taking residues of differential forms are called *geometric Goppa* codes or *algebraic-geometric* codes, abreviated by AG codes. The fact that $C_L(D, G)$ and $C_\Omega(D, G)$ are dual codes is seen by the *residue theorem*, which states that the sum of all residues over all places of a fixed differential is zero.

For classical Goppa codes, where we do not consider the subfield subcode case for simplicity, that is $\mathbf{F}_{q^e} = \mathbf{F}_q$, it is as follows. Let $h = \prod q_i^{e_i}$ be the factorization of the Goppa polynomial h and Q_i the place corresponding to q_i. Define the divisor $G = \sum e_i Q_i - P_\infty$, as before. A differential ω is an element of $\Omega(G - D)$ if and only if

$$\omega = \sum_{i=1}^{n} \frac{c_i dX}{X - \alpha_i} \quad \text{and} \quad \sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} \quad \text{is divisible by } \ h.$$

Moreover $\mathrm{res}_D(\omega) = \mathbf{c}$ whenever ω is of the above form. Thus $C_\Omega(D, G) = \Gamma(l, h)$.

The function field of a curve is a finite extension of $\mathbf{F}_q(X)$, thus it is of the form $\mathbf{F}_q(X)[Y]/(F(X, Y))$, where we may assume that $F(X, Y) \in \mathbf{F}_q[X, Y]$ and is absolutely irreducible, that is irreducible in $k[X, Y]$, where k is the algebraic closure of $\mathbf{F}_q$. We say that $F(X, Y)$ is an *affine equation* of a *plane model* of the curve. If we homogenize F, that is to say we consider $\tilde{F}(X, Y, Z) = Z^m F(X/Z, Y/Z)$, where m is the degree of F, then we say that $\tilde{F}$ is an equation of a *projective plane model*. If $\tilde{F}$ together with all its partial derivatives have no common zero $(a, b, c) \in k^3$, except $(0, 0, 0)$, we say that the projective plane model has no *singularities*, in this case all the $(a : b : c) \in PG(2, q)$ such that $\tilde{F}(a, b, c) = 0$, are the points of the curve, that is places of degree 1, moreover the genus of the curve is equal to $(m - 1)(m - 2)/2$.

Example 3.1 Hermitian curves.
The Hermitian curve $H(q)$ is defined by the affine equation

$$U^m + V^m + 1 = 0$$

over $GF(q)$, where $q = (m-1)^2$, for the details we refer to [56, 58]. Let $a, b \in GF(q)$ such that $a^{m-1} + a = b^m = -1$ and $P = (1 : b : 0)$. Define $X = b/(V - bU)$ and $Y = UX - a$. Then we have another equation

$$X^m - Y^{m-1} - Y = 0$$

of the same curve. We prefer the last equation since it has exactly one point at infinity. This plane curve is nonsingular and has $q\sqrt{q}$ points.

Now we return to the decoding of AG codes. The condition $A * B \subseteq C^\perp$ is easy to fulfill for algebraic-geometric codes. Suppose we want to decode the code $C = C_\Omega(D, G)$, we remarked already that the dual of this code is $C_L(D, G)$. Let F be any divisor with disjoint support with D, then $G - F$ has also disjoint support with D. Moreover if $f \in L(F)$ and $h \in L(G - F)$, then $fh \in L(G)$. If $\mathbf{a} = \mathrm{ev}_D(f)$ and $\mathbf{b} = \mathrm{ev}_D(h)$, then $\mathbf{a} * \mathbf{b} = \mathrm{ev}_D(fh)$. Thus

$$C_L(D, F) * C_L(D, G - F) \subseteq C_L(D, G) = C_\Omega(D, G)^\perp.$$

With the above estimates for the dimension and minimum distance of AG codes we have the following proposition and theorem.

Proposition 3.2 *Let F and G be divisors with support disjoint from D.*
Let $A = C_L(D, F)$, $B = C_L(D, G - F)$ and $C = C_\Omega(D, G)$. Then
1. *$A * B \subseteq C^\perp$.*
2. *If $t + g \leq deg(F) < n$, then $k(A) > t$.*
3. *If $deg(G - F) > 2g - 2$, then $d(A) + d(C) > n$.*
4. *If $deg(G - F) > t + 2g - 2$, then $d(B^\perp) > t$.*

Theorem 3.3 *Every algebraic-geometric code $C_\Omega(D, G)$ of designed minimum distance d^* on a curve of genus g has a $\lfloor (d^* - 1 - g)/2 \rfloor$-error correcting pair whenever $deg(G) > 2g - 2$.*

After the invention of Goppa of AG codes and the work of Tsfasman, Vlăduţ and Zink [60], who proved that there exists curves over finite fields with many points, giving asymptotically good codes, at the beginning of the eighties, it took some time before Justesen, Larsen, Elbrønd Jensen, Havemose and Høholdt [26] found at the end of the eighties a generalization of Peterson's algorithm for AG codes on plane curves. This

was generalized to arbitrary curves by Skorobogatov and Vlăduţ [55]. Independently Krachkovskii proved the same result, but his work was only published as a preprint in Russian [30]. These papers are surveyed in this and the previous section. Independently Porter gave another decoding algorithm which we will discuss in section 5. The above algorithm is called the *basic algorithm* in [55, 59]. In the *modified algorithm* one does not look at one error locating pair only, but at a sequence $(C_L(D, iP), C_L(D, G - iP))$ of pairs, where P is a point not in the support of G nor D, and one looks at the smallest i such that the coresponding vector space $K_{\mathbf{y}}$ is not zero. With the modified algorithm $(d^* - 1)/2 - s$ errors can be decoded [26, 28, 9], where s is the so called *Clifford defect*, see Duursma [9]. For plane curves s is roughly equal to $g/4$, but only in very special cases s is equal to zero. It can be shown that the modified algorithm does not always decode up to half the minimum distance [28], but the number of error patterns where the algorithm does not decode the error is relatively small [28, 41]. It was my contribution [40] to show that, on so called *maximal curves*, there exist u divisors $F_1, \ldots, F_u$, where u is of the order $2g$, such that for every received word with at most $\lfloor (d^* - 1)/2 \rfloor$ errors, at least one of the pairs $(C_L(D, F_i), C_L(D, G - F_i))$ corrects the errors. This was generalized to all curves by Vlăduţ [61]. For these last results one has to introduce the Zeta function of a curve over a finite field and the Jacobian of the curve. It is a counting argument to show that certain divisors exist, but in order to find these divisors explicitly one has to know quite a lot about the Jacobian and not much is known for this purpose. Rotillon and Thiong Ly [46] considered the problem of finding these divisors for the Klein quartic, and Le Brigand [31] for hyperelliptic curves. Carbonne and Thiong Ly [7] found for some curves the minimal u such that the above procedure works. First Ehrhard [13] considered divisors F_i of the form lP_j. In all these attempts one runs the algorithm for a lot of error locating pairs in parallel. Later Ehrhard [15] gave an explicit solution of the problem of decoding up to half the minimum distance, now the divisor F_i is changed during the algorithm. It is probably not possible to give an explanation of this result purely in terms of linear algebra. Moreover the designed minimum distance should be at least $4g$. Feng, Rao and Duursma gave another solution to the problem of finding an efficient explicit algorithm which decodes up to half the designed minimum distance. Their work on majority coset decoding is treated in the next section.

Any linear code is representable by means of a curve, see [42]. This is a nice mathematical result, but from a pratical coding point of view it puts matters upside down. If we want to construct good codes we have gained nothing by this knowledge. But it is hoped for that by this fact one can prove that any linear code of minimum distance d has a $\lfloor (d-1)/2 \rfloor$-error correcting pair. The property $A * B \subseteq C^{\perp}$ is difficult to fulfill for arbitrary linear codes and comes quite naturally in the geometric situation as the product of functions.

4 Majority coset decoding

In the following we explain the beautiful idea of Feng and Rao [16] which was treated in greater generality by Duursma [10, 11]. This can be explained purely in terms of linear algebra, in the same way as we abstracted from AG codes to linear codes in section 2.

Suppose we have a code C_1 for which we need a decoding algorithm, and a subcode C_2 for which we have a decoding algorithm. *Coset decoding* is an algorithm which has as input a word $\mathbf{y}_1$ such that $\mathbf{y}_1 \in \mathbf{e} + C_1$, and as output $\mathbf{y}_2$ such that $\mathbf{y}_2 \in \mathbf{e} + C_2$.

This was done for instance by Yaghoobian and Blake [62], where they decode Hermitian codes, but the complexity of the algorithm is in general rather great. Feng and Rao apply coset decoding to the code $C_1 = C_\Omega(D, mP)$, which has designed minimum distance d^* and its subcode $C_2 = C_\Omega(D, (m+g)P)$, which has designed minimum distance $d^* + g$, where P is a point not in D and g is the genus of the curve. We have seen in Theorem 3.3 that C_2 has a t-error correcting pair, where $t = \lfloor (d^* - 1)/2 \rfloor$. The coset decoding is done in several steps where the dimension of the subcode drops by one in every step. The whole setup is also used by Feng and Rao in another paper [17] to get an improved lower bound on the minimum distance of AG codes in case the redundancy is between g and $2g$. Furthermore it can be explained in terms of linear algebra only as we will do in the following. Instead of saying in the future that we are talking about three sequences of linear codes which satisfy conditions $(5), \ldots, (8)$, we prefer to give this notion a name, which is done in the following defnition.

Definition 4.1 An *array* of codes is a triple $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ of sequences of linear codes in $\mathbf{F}_q^n$, enumerated by $\mathcal{A} = (A_i)_{1 \leq i \leq u}$, $\mathcal{B} = (B_j)_{1 \leq j \leq v}$ and $\mathcal{C} = (C_r)_{w \leq r \leq l}$, such that the following holds:

5. $\dim(A_i) = i$, $\dim(B_j) = j$ and $\dim(C_r) = n - r$.
6. $A_i \subseteq A_{i+1}$, $B_j \subseteq B_{j+1}$ and $C_{r+1} \subseteq C_r$.
7. For every i and j there exists an r such that $A_i * B_j \subseteq (C_r)^\perp$. Thus for every i and j we define $r(i,j)$ to be the smallest index r such that $A_i * B_j \subseteq (C_r)^\perp$.
8. If $\mathbf{a} \in A_i \setminus A_{i-1}$ and $\mathbf{b} \in B_j \setminus B_{j-1}$ and $r = r(i,j) \geq w+1$, then $\mathbf{a} * \mathbf{b}$ is an element of $(C_r)^\perp \setminus (C_{r-1})^\perp$.

Remark that $r(i,j)$ is increasing, that is if $i \leq i'$ and $j \leq j'$, then $r(i,j) \leq r(i',j')$. Define the following set

$$N_r = \{(i,j) | 1 \leq i \leq u, 1 \leq j \leq v, r(i,j) = r+1\}$$

Let n_r be the number of elements of N_r. Define

$$d_r = \min\{n_{r'} | r \leq r' < l\} \cup \{d(C_l)\}.$$

Theorem 4.2 *For an array of codes we have that $d_r \leq d(C_r)$, for all $w \leq r \leq l$.*

For the proof we introduce bases. Let $\mathbf{a}_1, \ldots, \mathbf{a}_i$ be a basis of A_i and let $\mathbf{b}_1, \ldots, \mathbf{b}_j$ be a basis of B_j. Then $\mathbf{a}_i \in A_i \setminus A_{i-1}$ and $\mathbf{b}_j \in B_j \setminus B_{j-1}$, by conditions (5) and (6). Thus $\mathbf{a}_i * \mathbf{b}_j \in (C_r)^\perp \setminus (C_{r-1})^\perp$, where $r = r(i,j)$, by condition (7). The proof of the theorem goes by decreasing induction on r. If $r = l$, then the conclusion is obvious by the definition of $d_l = d(C_l)$. Suppose we have already proved $d(C_{r+1}) \geq d_{r+1}$. Now we prove that $d(C_r) \geq d_r$. If $\mathbf{y}$ is a nonzero word in C_{r+1}, then $wt(\mathbf{y}) \geq d_{r+1} \geq d_r$, by the induction hypothesis, and the definition of d_r. If $\mathbf{y} \in C_r \setminus C_{r+1}$, then look at the entries $S_{i,j}$ in the $u \times v$ matrix $\mathbf{S}$, where $S_{i,j} =< \mathbf{y}, \mathbf{a}_i * \mathbf{b}_j >$. If $r(i,j) \leq r$, then $S_{i,j} = 0$, by (7). If $r(i,j) = r+1$, then $S_{i,j} \neq 0$. Because $(C_{r+1})^\perp = (C_r)^\perp + < \mathbf{a}_i * \mathbf{b}_j >$, since $\mathbf{a}_i * \mathbf{b}_j \in (C_{r+1})^\perp \setminus (C_r)^\perp$ by the above remark and $\dim(C_r) = \dim(C_{r+1}) + 1$, by (5). Therefore $\mathbf{S}$ has an echelon form with nonzero pivots at all entries $S_{i,j}$ where $r(i,j) = r+1$. Thus the rank of the matrix $\mathbf{S}$ is at least equal to the number of elements of N_r, which we called n_r. On the other hand, we can decompose the same matrix in a product of three matrices $\mathbf{S} = \mathbf{AWB}^t$, where $\mathbf{A}$ is the matrix with $\mathbf{a}_1, \ldots, \mathbf{a}_u$ as rows, $\mathbf{B}$ is the matrix with $\mathbf{b}_1, \ldots, \mathbf{b}_v$ as rows and $\mathbf{W}$ is the $n \times n$ diagonal matrix with $\mathbf{y}$ on the diagonal, and zeros outside the diagonal. Hence $\mathrm{rank}(\mathbf{S}) \leq \mathrm{rank}(\mathbf{W}) = wt(\mathbf{y})$. Combining the two inequalities involving the rank of $\mathbf{S}$ we get $wt(\mathbf{y}) \geq n_r$, which is at least d_r, by definition. Thus $d(C_r) \geq d_r$, and we have proved the theorem by induction.

In order to be able to decode C_w up to $\lfloor (d_w - 1)/2 \rfloor$ errors, we need an extra condition.

Definition 4.3 An array of codes $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ of sequences of linear codes in $\mathbf{F}_q^n$, enumerated by $\mathcal{A} = (A_i)_{1 \leq i \leq u}$, $\mathcal{B} = (B_j)_{1 \leq j \leq v}$ and $\mathcal{C} = (C_r)_{w \leq r \leq l}$, is called a *$t$-error correcting array* for a code C in $\mathbf{F}_q^n$, if $C = C_w$ and $t \leq (d_w - 1)/2$, and $C_l = 0$ or there exists i and j such that (A_i, B_j) is a t-error correcting pair for C_r, where $r = r(i,j)$.

Remark that we could have taken the codes A_i, B_j and C_r to be $\mathbf{F}_{q^e}$-linear as we have done in the definition of a t-error correcting pair. In that case we assume that C is in $\mathbf{F}_q^n$ and a subfield subcode of C_w.

Theorem 4.4 *If a code has a t-error correcting array, then it has a decoding algorithm of complexity $\mathcal{O}(n^3)$ which corrects t errors.*

Before we prove the theorem we introduce some definitions, see [16, 17]. Let $\mathbf{S}$ be a $(u \times v)$-matrix with entries $S_{i,j}, 1 \leq i \leq u, 1 \leq j \leq v$. Let $\mathbf{S}(i,j)$ be the $(i \times j)$-matrix with entries $S_{i',j'}, 1 \leq i' \leq i, 1 \leq j' \leq j$. Consider the folowing two conditions.

9. $\text{rank}(\mathbf{S}(i-1,j-1)) = \text{rank}(\mathbf{S}(i-1,j)) = \text{rank}(\mathbf{S}(i,j-1))$

10. $\text{rank}(\mathbf{S}(i-1,j-1)) = \text{rank}(\mathbf{S}(i,j))$

Clearly (10) implies (9); conversely if (9) holds, then there exists a unique value of $S_{i,j}$ such that (10) holds. We call (i,j) a *discrepancy* if (9) holds and (10) does not hold. Remark that (9) holds for (i,j) if and only if (10) holds for all $(i',j), 1 \leq i' < i$ and all $(i,j'), 1 \leq j' < j$. Thus in every row there is at most one discrepancy and the same holds for every column. The number of discrepancies of $\mathbf{S}$ is equal to the rank of $\mathbf{S}$.

Now we prove Theorem 4.4. Suppose $\mathbf{y}$ is a received word and has error $\mathbf{e}$ with respect to $C = C_w$ of weight at most $t \leq (d_w - 1)/2$. Let

$$S_{i,j} = < \mathbf{e}, \mathbf{a}_i * \mathbf{b}_j >$$

and let $\mathbf{S}$ be the corresponding matrix with entries $S_{i,j}$. The proof now goes by increasing induction on r. For all (i,j) such that $r(i,j) \leq w$ we know the syndrome $S_{i,j}$, since $\mathbf{y} \in \mathbf{e} + C_w$. Suppose we already know an element $\mathbf{y}_r \in \mathbf{e} + C_r$ and all the syndromes $S_{i,j}$ for (i,j) such that $r(i,j) \leq r < l$. Now we explain how to get an element $\mathbf{y}_{r+1} \in \mathbf{e} + C_{r+1}$ and the syndromes $S_{i,j}$ such that $r(i,j) \leq r+1$. We call a pair (i,j) a *candidate* if $r(i,j) = r+1$ and (9) holds. A candidate is called *true* if (10) holds, and *false* if (10) does not hold. We know the candidates but we do not know which candidates are true or false. We first prove the remarkable property that the number of true candidates, which we will denote by T, is greater than the number of false candidates, which we will denote by F. Afterwards we show how to compute a certain $\lambda \in \mathbf{F}_q$ for every candidate, which is the same for all true candidates and thus gives us $\mathbf{y}_{r+1}$ and new syndromes. We have seen already in the proof of Theorem 4.2 that the rank of $\mathbf{S}$ is at most $wt(\mathbf{e}) \leq t$. Denote the number of *known* discrepancies, that is discrepancies at entries (i,j) such that $r(i,j) \leq r$, by K. The other discrepancies are called *unknown*. Clearly we have that all false candidates are unknown discrepancies. Thus

$$K + F \leq \text{the number of all discrepancies} = \text{rank}(\mathbf{S}) \leq t.$$

Furthermore for every pair (i,j) such that $r(i,j) = r+1$, that is $(i,j) \in N_r$, which is not a candidate, there exists a known discrepancy in the same row or column, possibly in both. For every candidate, true or false, there exist no known discrepancy in the same row nor column. We called n_r the number of elements of N_r, thus

$$n_r \leq T + F + 2K.$$

If we combine the two inequalties above and use that $2t < d_w \leq d_r \leq n_r$, then we get

$$n_r \leq T + F + 2K \leq T + F + (2t - 2F) < T - F + n_r.$$

Therefore

$$F < T,$$

that is the number of true candidates is greater than the number of false candidates. We associate with each candidate an element $\lambda \in \mathbf{F}_q$ in such a way that all true candidates have the same λ, thus giving a way to know the true candidates, by majority as we explained above, and therefore the syndromes $S_{i,j}$ such that $r(i,j) = r+1$. For every candidate, that is for every (i,j) such that (9) holds, there is a unique $S'_{i,j} \in \mathbf{F}_q$ to fill at entry (i,j) in order that (10) holds. This entry $S'_{i,j}$ can be computed using the known syndromes, and is equal to $S_{i,j}$ if and only if (i,j) is a true candidate. The vector space C_r contains the vector space C_{r+1} and the quotient is one dimensional, so there exists a vector $\mathbf{c}_r$ such that $C_r =< \mathbf{c}_r > + C_{r+1}$. Therefore there exists a unique $\lambda_r \in \mathbf{F}_q$ such that $\mathbf{y}_r + \lambda_r \mathbf{c}_r$, which we will call $\mathbf{y}_{r+1}$, is an element of $\mathbf{e} + C_{r+1}$. If $r(i,j) = r+1$, then $C_{r+1}^{\perp} = C_r^{\perp} + < \mathbf{a}_i * \mathbf{b}_j >$. Thus $< \mathbf{c}_r, \mathbf{a}_i * \mathbf{b}_j >$ is not equal to zero. By taking the inner product with $\mathbf{a}_i * \mathbf{b}_j$ we get

$$< \mathbf{y}_r, \mathbf{a}_i * \mathbf{b}_j > + \lambda_r < \mathbf{c}_r, \mathbf{a}_i * \mathbf{b}_j >= S_{i,j}.$$

For every candidate (i,j) we compute $S'_{i,j}$ as explained above and the element $\lambda_{i,j}$ defined by

$$\lambda_{i,j} = \frac{S'_{i,j} - < \mathbf{y}_r, \mathbf{a}_i * \mathbf{b}_j >}{< \mathbf{c}_r, \mathbf{a}_i * \mathbf{b}_j >}.$$

If (i,j) is a true candidate, then $S'_{i,j} = S_{i,j}$ so $\lambda_{i,j} = \lambda_r$. Therefore all $\lambda_{i,j}$ are the same for all true candidates (i,j). Thus the λ which occurs most often among the $\lambda_{i,j}$ of the candidates is equal to λ_r. In this way we get $\mathbf{y}_{r+1} \in \mathbf{e} + C_{r+1}$ and the syndromes $S_{i,j}$ such that $r(i,j) \leq r+1$, that is one step further in the induction. Finaly $C_l = 0$ or there exists i and j such that (A_i, B_j) is a t-error correcting pair for C_r, where $r = r(i,j)$ and $t \leq (d_w - 1)/2$. In the first case we are done, since $\mathbf{y}_l = \mathbf{e}$ when $C_l = 0$, and in the second case we can apply the decoding algorithm of section 2 to the word $\mathbf{y}_r$ with the pair (A_i, B_j) for C_r, which gives $\mathbf{e}$ as output.

Theorem 4.5 *Every algebraic-geometric code $C_\Omega(D,G)$ of designed minimum distance d^* on a curve of genus g has a $\lfloor (d^*-1)/2 \rfloor$-error correcting array, whenever $2g-2 < deg(G) < n-g$ and n is smaller than the number of points on the curve.*

We give a scetch of the proof. The number of points on the curve is greater than n, so we can take a point P which is not in the support of D. Let a_P be the coefficient of P in G. There are increasing sequences $(\mu_i)_{i=1}^u$ and $(\nu_j)_{j=1}^v$ such that $A_i = C_L(D, \mu_i P)$ and $B_j = C_L(D, G + \nu_j P)$ and $C_r = C_\Omega(D, G + \nu_r P)$ define a $\lfloor (d^*-1)/2 \rfloor$-error correcting array. Conditions 5 and 6 are immediate by the appropriate choice of the sequences (μ_i) and (ν_i). Condition 7 is seen as follows. If $\mathbf{a} \in A_i \setminus A_{i-1}$ and $\mathbf{b} \in B_j \setminus B_{j-1}$, then

there exist functions $f \in L(\mu_i P) \setminus L((\mu_i - 1)P)$ and $h \in L(G + \nu_j P) \setminus L(G + (\nu_j - 1)P)$ such that $f(P_l) = a_l$ and $h(P_l) = b_l$ for all l. Thus f has pole order μ_i at P and h has pole order $a_P + \nu_j$ at P, thus fh has pole order $a_P + \mu_i + \nu_j$ at P. There exists an r such that $\mu_i + \nu_j = \nu_r$, so $fh \in L(G + \nu_r P) \setminus L(G + (\nu_r - 1)P)$. Furthermore $\mathrm{ev}_D(fh) = \mathbf{a} * \mathbf{b}$ and the map ev_D is injective on $L(G + \nu_r P)$ since $\deg(G) < n - g$, so $\mathbf{a} * \mathbf{b} \in C_r^{\perp} \setminus C_{r-1}^{\perp}$.

In the papers of Feng and Rao only codes of the form $C_\Omega(D, mP)$ are considered, so there it suffices to take $B_i = A_i$. In Duursma's approach arbitrary divisors G are allowed and one can take B_i different from A_i. The restriction that the number of points on the curve is greater than n remains.

5 Decoding algebraic-geometric codes by solving the key equation

Around the same time as Justesen et al. [26], Porter [44] found another decoding algorithm, which is a generalization of solving the key equation of classical Goppa codes by Euclid's algorithm in the ring of polynomials in one variable. The thesis of Porter contained some mistakes but these were corrected in [45]. Ehrhard [13, 14] also showed that the results of Porter are correct, moreover he proved the equivalence of the modified algorithm and the decoding by solving the key equation.

One can view the ring of polynomials in one variable as the ring of rational functions on the projective line with only poles at the point at infinity. The ring of polynomials in one variable is replaced by the ring $K_\infty(P)$ of rational functions on the curve with only poles at a fixed point P, where P is not equal to one of the points used to construct the geometric Goppa code.

Now we scetch the generalization of the classical Goppa codes. Let l be a subset of $\mathbf{F}_q$ and h a Goppa polynomial. Let $l = \{\alpha_1, \ldots, \alpha_n\}$. Suppose h is not zero at α_i, for all i. We already mentioned that the classical Goppa code $\Gamma(L, h)$ is defined by

$$\Gamma(l, h) = \{\mathbf{y} | \sum \frac{y_i}{X - \alpha_i} \equiv 0 \pmod{h}\}.$$

If we let $\varepsilon_i = dX/(X - \alpha_i)$, and take for P the point at infinity on the projective line and for E the divisor of zeros of h, then $\Gamma(l, h) = C_\Omega(D, E - P)$ and

$$\mathbf{y} \in \Gamma(L, h) \text{ if and only if } \sum \frac{y_i}{X - \alpha_i} dX \in \Omega(E - P - D)$$

First it is shown that for arbitrary AG codes one may assume that the divisor G in the definition of the code $C_\Omega(D,G)$ is of the form $E-\mu P$, where E is an effective divisor and μ a positive integer. Secondly one can show that there always exist n independent differentials $\varepsilon_1,\ldots,\varepsilon_n \in \Omega(-D-\mu P)$ such that $\mathrm{res}_{P_i}(\varepsilon_j)$ is 1 if $i=j$ and 0 otherwise, and for every differential $\omega \in \Omega(E-\mu P-D)$ we have

$$\omega = \sum \mathrm{res}_{P_i}(\omega)\varepsilon_i.$$

If we let $\varepsilon(\mathbf{y}) = \sum y_i\varepsilon_i$, then the map ε is the right inverse of res_D and

$$\varepsilon(\mathbf{y}) \in \Omega(E-\mu P-D) \text{ if and only if } \mathbf{y} \in C_\Omega(D, E-\mu P).$$

Now suppose that E is the divisor of zeros of a function $h \in K_\infty(P)$ which is not zero at P_i for all i. We want to define the *syndrome* of a received word. In order to represent the syndrome as a rational function one proves the existence of a particular differential η first. For classical Goppa codes one takes $\eta = dX$. The syndrome $S(\mathbf{y})$ of a received word $\mathbf{y}$ is now defined as follows.

$$S(\mathbf{y})\eta = \sum y_i \frac{h(P_i)-h}{h(P_i)}\varepsilon_i.$$

The syndrome is an element of the ring $K_\infty(P)$, and if E is the divisor of zeros of $h \in K_\infty(P)$, then

$$\mathbf{y} \in C_\Omega(D, E-\mu P) \text{ if and only if } S(\mathbf{y}) \equiv 0 \pmod{h}.$$

For simplicity we assume that η is a differential such that $(\eta) = (2g-2)P$. Now one searches for *solutions* of the *key equation*, that is for pairs (f,r) with $f,r \in K_\infty(P)$ such that there exists an $a \in K_\infty(P)$ with the property

$$fS(\mathbf{y}) = r + ah.$$

A solution is called *valid* if moreover $\deg(r) - \deg(f) \leq 2g-2+\mu$. A valid solution (f,r) is called *minimal* if $\deg(f)$ is minimal among all the degrees of f' such that (f',r') is a valid solution. If the received word $\mathbf{y}$ has at most $(d^*-1)/2 - s$ errors, where s is the Clifford defect [9], then a minimal valid solution (f,r) has the property that $\mathrm{res}_{P_i}(r\eta/f)$ is the error value at place i for all i. Shen [52, 53, 54] computed explicit formulas for the differentials $\varepsilon_1,\ldots,\varepsilon_n$ and the syndromes of codes on Hermitian curves.

Euclid's algorithm gives in the case of classical Goppa codes a sequence of solutions of the key equation, and the first valid solution in this sequence is also a minimal valid solution. The ring $K_\infty(P)$ is not a Euclidean ring whenever the genus is not zero. The sequence of solutions in the Euclidean algorihm is replaced by an algorithm giving the so called *subresultant sequence*, see Porter [44] and Shen [51, 54].

6 Improvements of the complexity

It is not before one has decoding algorithms as fast as Euclid's algorithm or the Berlekamp-Massey algorithm [2, 36] solving the key equation, that AG codes will be implemented for practical purposes.

For polynomials in one variable division with rest gives Euclid's algorithm. Buchberger's algorithm [6] computing Gröbner bases is a generalization of this to polynomials in several variables. Sakata [47, 48] gave a generalization of the Berlekamp-Massey algorithm to several variables. Using Sakata's ideas Justesen, Larsen, Elbrønd Jensen and Høholdt [27] could improve the complexity of their algorithm for codes on plane curves from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{7/3})$. Dahl Jensen [25] generalized this and obtained complexity $\mathcal{O}(n^{3-\frac{2}{r+1}})$ for curves in projective r-space. In analogy with the work Justesen et al. [27] following Sakata, Shen [53, 54] obtained the same complexity for codes on Hermitian curves using Porter's algorithm.

Up to now nobody succeeded in obtaining the same complexity for decoding AG codes as Euclid's algorithm, that is $\mathcal{O}(n^2)$.

Acknowledgement I want to thank I.M. Duursma and G.-L. Feng for many discussions explaining the mechanism of majority coset decoding.

References

[1] S. Arimoto, Encoding and decoding of p-ary group codes and the correction system, Information Processing in Japan **2** (1961), 320-325. in Japanese.

[2] E.R. Berlekamp, Algebraic coding theory, McGraw-Hill, New York 1968.

[3] E.R. Berlekamp, R.J. McEliece and H.C.A. van Tilborg, On the inherent intractibility of certain coding problems, IEEE Trans. Inform. Theory **24** (1978), 384-386.

[4] P. Bours, J.C.M. Janssen, M. van Asperdt and H.C.A. van Tilborg, Algebraic decoding beyond e_{BCH} of some binary cyclic codes, when $e > e_{BCH}$, IEEE Trans. Inform. Theory **36** (1990), 214-222.

[5] J. Bruck and M. Naor, The hardness of decoding linear codes with preprocessing, IEEE Trans. Inform. Theory **36** (1990), 381-385.

[6] B. Buchberger, Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, PhD Thesis, University of Innsbruck, Austria, 1965.

[7] Ph. Carbonne and A. Thiong Ly, Minimal exponent for Pellikaan's decoding algorithm, Proceedings Eurocode 92.

[8] C. Chevalley, Introduction to the theory of algebraic functions in one variable, Math. Surveys VI, Providence, AMS 1951.

[9] I.M. Duursma, Algebraic decoding using special divisors, to appear in IEEE Trans. Inform. Theory.

[10] I.M. Duursma, Majority coset decoding, to appear in IEEE Trans. Inform. Theory.

[11] I.M. Duursma, On the decoding procedure of Feng and Rao, Proceedings ACCT-3, Voneshta Voda, June 1992.

[12] I.M. Duursma and R. Kötter, On error locating pairs for cyclic codes, preprint October 1992.

[13] D. Ehrhard, Über das Dekodieren Algebraisch-Geometrischer Codes, PhD Thesis, University of Düsseldorf, July 1991.

[14] D. Ehrhard, Decoding algebraic-geometric codes by solving a key equation, in the Proceedings AGCT-3, H. Stichtenoth and M.A. Tsfasman (eds.), Luminy 1991, Springer Lect. Notes. **1518** (1992), 18-25.

[15] D. Ehrhard, Achieving the designed error capacity in decoding algebraic-geometric codes, to appear in IEEE Trans. Inform. Theory.

[16] G.-L. Feng and T.R.N. Rao, Decoding of algebraic geometric codes up to the designed minimum distance, to appear in IEEE Trans. Inform. Theory.

[17] G.-L. Feng and T.R.N. Rao, A novel approach for construction of algebraic-geometric codes from affine plane curves, University of Southwestern Louisiana, preprint 1992.

[18] G.-L. Feng and K.K. Tzeng, A generalization of the Berlekamp-Massey algorithm for multisequence shift register synthesis with application to decoding cyclic codes, IEEE Trans. Inform. Theory **37** (1991), 1274-1287.

[19] G.-L. Feng and K.K. Tzeng, Decoding cyclic and BCH codes up to the actual minimum distance using nonrecurrent syndrome dependence relations, IEEE Trans. Inform. Theory **37** (1991), 1716-1723.

[20] V.D. Goppa, Codes associated with divisors, Probl. Peredachi Inform. **13**(1) (1977), 33-39. Translation: Probl. Inform. Transmission **13** (1977), 22-26.

[21] V.D. Goppa, Codes on algebraic curves, Dokl. Akad. Nauk SSSR **259** (1981), 1289-1290, Translation: Soviet Math. Dokl. **24** (1981), 170-172.

[22] V.D. Goppa, Algebraico-geometric codes, Izv. Akad. Nauk SSSR **46** (1982), Translation: Math. USSR Izvestija **21** (1983), 75-91.

[23] V.D. Goppa, Codes and information, Russian Math. Surveys **39** (1984), 87-141.

[24] V.D. Goppa, Geometry and codes, Mathematics and its Applications **24**, Kluwer Acad. Publ., Dordrecht, 1991.

[25] C. Dahl Jensen, Codes and geometry, PhD Thesis, Technical University of Denmark, May 1991,

[26] J. Justesen, K.J. Larsen, H.Elbrønd Jensen, A. Havemose and T. Høholdt, Construction and decoding of a class of algebraic geometric codes, IEEE Trans. Inform. Theory **35** (1989), 811-821.

[27] J. Justesen, K.J. Larsen, H.Elbrønd Jensen and T. Høholdt, Fast decoding of codes from algebraic plane curves, IEEE Trans. Inform. Theory **38** (1992), 111-119.

[28] J. Justesen, H.Elbrønd Jensen and T. Høholdt, On the number of correctable errors for some AG-codes, to appear in IEEE Trans. Inform. Theory.

[29] R. Kötter, A unified description of an error locating procedure for linear codes, Proceedings ACCT-3, Voneshta Voda, June 1992.

[30] V. Yu. Krachkovskii, Decoding of codes on algebraic curves, Odessa, preprint 1988 in Russian.

[31] D. Le Brigand, Decoding of codes on hyperelliptic curves, Proceedings Eurocode 90, G.D. Cohen and P. Charpin (eds.), Lect. Notes in Comp. Sc. **514** (1991), 126-134.

[32] J.H. van Lint, Algebraic geometric codes, in Coding Theory and Design Theory, part I, D. Ray-Chaudhuri (ed.), IMA Volumes Math. Appl. **21**, Springer-Verlag, Berlin etc. (1990),

[33] J.H. van Lint and G. van der Geer, Introduction to coding theory and algebraic geometry, DMV Seminar **12**, Birkhäuser Verlag, Basel Boston Berlin, 1988.

[34] J.H. van Lint and T.A. Springer, Generalized Reed-Solomon codes from algebraic geometry, IEEE Trans. Inform. Theory **33** (1987), 305-309.

[35] J.H. van Lint and R.M. Wilson, On the minimum distance of cyclic codes, IEEE Trans. Inform. Theory **32** (1996), 23-40.

[36] J.L. Massey, Shift-register synthesis and BCH decoding, IEEE Trans. Inform. Theory **15** (1969), 122-127.

[37] F.J. McWilliams and N.J.A. Sloane, The theory of error-correcting codes, North-Holland Math. Library **16**, North-Holland, Amsterdam, 1977.

[38] C. Moreno, Algebraic curves over finite fields, Cambridge Tracts in Math. **97**, Cambridge Un. Press, 1991.

[39] R. Pellikaan, On decoding linear codes by error correcting pairs, preprint Eindhoven University of Technology, 1988.

[40] R. Pellikaan, On a decoding algorithm for codes on maximal curves, IEEE Trans. Inform. Theory **35** (1989), 1228-1232.

[41] R. Pellikaan, On the decoding by error location and the number of dependent error positions, Discrete Math. **106/107** (1992), 369-381.

[42] R. Pellikaan, B.-Z. Shen and G.J.M. van Wee, Which linear codes are algebraic-geometric ?, IEEE Trans. Inform. Theory. IT-**37** (1991), 583-602.

[43] W.W. Peterson, Encoding and error-correction procedures for the Bose-Chauduri codes, IEEE Trans. Inform. Theory **6** (1960), 459-470.

[44] S.C. Porter, Decoding codes arising from Goppa's construction on algebraic curves, Thesis, Yale University, dec. 1988.

[45] S.C. Porter, B.-Z. Shen and R. Pellikaan, On decoding geometric Goppa codes using an extra place, IEEE Trans. Inform. Theory. IT-**38** (1992), 1663-1676.

[46] D. Rotillon and J.A. Thiong Ly, Decoding codes on the Klein quartic, Proceedings Eurocode 90, G.D. Cohen and P. Charpin (eds.), Lect. Notes in Comp. Sc. **514** (1991), 135-150.

[47] S. Sakata, On determining the independent point set for doubly periodic arrays and encoding two-dimensional cyclic codes and their duals, IEEE Trans. Inform. Theory **27** (1981), 556-565.

[48] S. Sakata, Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array, Journal of Symbolic Computation, **5** (1988), 321-337.

[49] S. Sakata, Extension of the Berlekamp-Massey algorithm to N dimensions, Information and Computation, **84** (1990), 207-239.

[50] S. Sakata, Decoding binary 2-D cyclic codes by the 2-D Berlekamp-Massey algorithm, IEEE Trans. Inform. Theory **37** (1991), 1200-1203.

[51] B.-Z. Shen, Solving a congruence on a graded algebra by a subresultant sequence and its application, to appear in Journ. of Symbolic Computation.

[52] B.-Z. Shen, On encoding and decoding of the codes from Hermitian curves, to appear in Cryptography and Coding III, the IMA Conference Proceedings Series, M. Ganley (ed.), Oxford University Press.

[53] B.-Z. Shen, Constructing syndromes for the codes from Hemitian curves and a decoding approach, preprint Eindhoven University of Technology, 1992.

[54] B.-Z. Shen, Algebraic-geometric codes and their decoding algorithm, PhD Thesis, Eindhoven University of Technology, September 1992.

[55] A.N. Skorobogatov and S.G. Vlăduţ, On the decoding of algebraic-geometric codes, IEEE Trans. Inform. Theory **36** (1990), 1051-1060.

[56] H. Stichtenoth, A note on Hermitian codes over GF(q^2), IEEE Trans. Inform. Theory **34** (1988), 1345-1348.

[57] H. Stichtenoth, Algebraic function fields and codes, to appear in Universitext, Springer-Verlag, 1993.

[58] H.J. Tiersma, Codes comming from Hermitian curves, IEEE Trans. Inform. Theory **33** (1987), 605-609.

[59] M.A. Tsfasman and S.G. Vlăduţ, Algebraic-geometric codes, Mathematics and its Applications **58**, Kluwer Acad. Publ., Dordrecht, 1991.

[60] M.A. Tsfasman, S.G. Vlăduţ and T. Zink, Modular curves, Shimura curves and Goppa codes, better than Varshamov-Gilbert bound, Math. Nachrichten **109** (1982), 21-28.

[61] S.G. Vlăduţ, On the decoding of algebraic-geometric codes over GF(q) for $q \geq 16$, IEEE Trans. Inform. Theory **36** (1990), 1461-1463.

[62] T. Yaghoobian and I.F. Blake, Hermitian codes as generalized Reed-Solomon codes, Designs, Codes and Cryptogrphy **2** (1992), 15-18.