Rend. Sem. Mat. Univ. Pol. Torino - Vol. 62, 3 (2004)

A. Campillo – J.I. Farrán

ADJOINTS AND CODES

Abstract. We discuss an algorithm to compute bases for the space $\mathcal{L}(G)$, provided G is a rational divisor over a non-singular absolutely irreducible algebraic curve. The algorithm is founded on the Brill–Noether algorithm by using the theory of Hamburger–Noether expansions, and it is given in terms of symbolic computation. As a byproduct, we introduce a method to compute the Weierstrass semigroup at P together with functions for each value in this semigroup, provided P is a rational point of this curve. This methods are nice applications of the classical adjuntion theory.

On the other hand, we discuss an alternative method for the computation of Weierstrass semigroups and the corresponding functions on the basis of the Abhankar-Moh theorem, for the case of a plane curve with only one branch at infinity. This method requires the precomputation of a certain integral basis. Both alternative methods be applied to the effective construction (Riemann–Roch problem) and decoding (Weierstrass semigroups) of Algebraic Geometry codes. The second method can be also regarded as a kind of adjunction procedure, and we compare both alternatives from a general point of view.

1. An introduction to Coding Theory and AG codes

The general problem of error-correcting codes is as follows: A message should be sent through a noisy channel and one wants, as far as possible, detect or even correct the (possibly) committed errors. For that, on has to encode the original message with enough redundancy, so that few errors can be detected of corrected in an algorithmic (and efficient) way.

For the sake of simplicity, we assume that the above situation can be described with mathematical terminology in the following way: The symbols of an information source are identified to the elements of a finite field $\mathbb{F}_q$, that is the *alphabet* which is used to write such information. If the message is a sequence of k such symbols, it can be considered as an element of the $\mathbb{F}_q$-vector space $\mathbb{F}_q^n$, whose *dimension* is k. Thus, an *encoding* against errors is an injective linear map

$$\mathbb{F}_q^k \hookrightarrow \mathbb{F}_q^n$$

and the image of such map is called (linear) code, that is, a linear subspace $C \subseteq \mathbb{F}_q^n$ of dimension k. The number n is called the *length* of the code, and then $n - k$ is the *redundancy*. If a vector $\mathbf{c}$ is in C, it is called a *codeword*, and if an error vector $\mathbf{e}$ is added, the received word is then $\mathbf{x} = \mathbf{c} + \mathbf{e}$.

From a theoretical point of view, the code C detects a configuration of errors $\mathbf{e}$ if the received word $\mathbf{x} = \mathbf{c} + \mathbf{e}$ is not a codeword. In practice, detecting errors is just checking if $\mathbf{x} \in C$ or not, and this is easily checkable by just using linear algebra. Note that there also exist non-linear codes in the literature, but this point (and many others) would be more complicated to solve.

From a computational point of view, let $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ be a vector basis of C, and

construct the *generator matrix* G of C whose rows are the vector of such a basis. The encoding process is just a matrix multiplication

$$\mathbf{m} \mapsto \mathbf{m} \cdot G$$

where $\mathbf{m}$ are the information symbols (the original message). In principle, checking if $\mathbf{x} \in C$ (error detection) would be just checking whether $\mathbf{x}$ is a linear combination of $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ or not. This is usually solved in the following way: Take a vector basis of the orthogonal space $C^{\perp}$ (with respect to the standard non-degenerated bilinear form on $\mathbb{F}_q^n$), namely $\{\mathbf{h}_1, \ldots, \mathbf{c}_{n-k}\}$, and construct the *parity-check matrix* H of C whose rows are these vectors. Thus

$$\mathbf{x} \in C \Leftrightarrow H \cdot \mathbf{x}^t = \mathbf{0}.$$

The problem of error correction is harder. In the origins of the Coding Theory the approach was purely probabilistic: For a given received word, one considers that the emitted word is the codeword that minimizes some conditioned probability (decoding by *maximum likelihood*). Nowadays, from the development of the Algebraic Coding Theory, the approach is more algorithmic: Find the "nearest" codeword to the received word $\mathbf{x}$, where proximity is defined from the metric given by the so-called **Hamming distance**

$$d(\mathbf{x}, \mathbf{y}) := \sharp\{i \mid x_i \neq y_i\}$$

Unfortunately, this is a computationally hard problem (NP), and finding a efficient solution for the decoding problem depends strongly on the structure of the concrete code, that is, on the way in which the code is constructed.

On the other hand, one should know how many errors are expected to be detected (or corrected) by the code. First, one defined the so-called minimum distance of the code as

$$d \equiv d(C) := \min\{d(\mathbf{c}, \mathbf{c}') \mid \mathbf{c}, \mathbf{c}' \in C, \quad \mathbf{c} \neq \mathbf{c}'\}$$

Then, one easily checks that

a) C detects any t errors, whenever $t < d$.

b) C corrects any t errors, whenever $2t < d$.

Note that "corrects" means just that the problem of finding the nearest word to a received word has a unique solution and it equals the emitted word, not that we are able to do it algorithmically. Finally, an important (but also hard) problem is to computed (or estimate) d. In principle, we are satisfied with just finding a good enough *designed distance* d^* such that $d \geq d^*$. You can see more details on Coding Theory for example in [14] or [12].

We now described shortly the construction of error-correcting codes from algebraic curves (see [12] or [16] for further details). Let $\tilde{\chi}$ be a non-singular projective algebraic curve defined over a finite field $\mathbb{F}$ such that $\tilde{\chi}$ is irreducible over $\overline{\mathbb{F}}$. In order to define the Algebraic Geometry codes (AG codes in short), take $\mathbb{F}$-rational points $P_1, \ldots, P_n$ of

the curve and a $\mathbb{F}$-rational divisor G (which can be assumed effective) having disjoint support with $D \doteq P_1 + \ldots + P_n$, and then consider the well-defined linear maps

$$\begin{array}{c} ev_D \ : \ \mathcal{L}(G) \longrightarrow \mathbb{F}^n \\ f \mapsto (f(P_1), \ldots, f(P_n)) \end{array} \quad \text{and} \quad \begin{array}{c} res_D \ : \ \Omega(G-D) \longrightarrow \mathbb{F}^n \\ \omega \mapsto (res_{P_1}(\omega), \ldots, res_{P_n}(\omega)) \end{array} .$$

One defines the linear codes

$$C_L \equiv C_L(D,G) \doteq Im(ev_D) \quad , \quad C_\Omega \equiv C_\Omega(D,G) \doteq Im(res_D).$$

The length of both codes is obviously n, and one has $(C_\Omega) = C_L^\perp$ by the *residues theorem*. On the other hand, given D and G as above there exists a differential form ω such that $C_L(D,G) = C_\Omega(D, D - G + (\omega))$ and thus it suffices to deal with the codes of type C_Ω.

Denote by $k(C)$ and $d(C)$ the dimension over $\mathbb{F}$ and the minimum distance of the linear code C respectively, $d(C)$ being the minimum number of non-zero entries of a non-zero vector of C. Goppa estimates for $k(C)$ and $d(C)$ are derived from the Riemann–Roch formula. In fact, if $2g - 2 < deg\, G < n$ then

$$(1) \ \left[\begin{array}{lcl} k(C_L) & = & deg\, G + 1 - g \\ d(C_L) & \geq & n - deg\, G \end{array} \right. \qquad (2) \ \left[\begin{array}{lcl} k(C_\Omega) & = & n - deg\, G + g - 1 \\ d(C_\Omega) & \geq & deg\, G + 2 - 2g \end{array} \right.$$

The main problem to solve for the construction of such codes consists in principle of computing bases for $\mathcal{L}(G)$, finding points (rational or not) of the curve and evaluating functions of $\mathcal{L}(G)$ at some rational points. In practice, there is no general method for computing bases for $\mathcal{L}(D)$, so that one must use a (possibly singular) plane model χ for the non-singular curve $\tilde{\chi}$, for which there exist classical methods for doing it (namely Brill–Noether [10] or Coates [6] algorithms). Thus, one must substitute the term "points" by the term "branches" (or "places", if one deals with the function field instead of the geometric object). In this way, the computational problems which are involved in the effective construction of AG codes can basically be reduced to the following ones:

(1) Find sufficiently many $\mathbb{F}$-rational places of $\tilde{\chi}$, so that $n > 2g - 2$. This can be done by means of Gröbner bases computation, after having a good description of the resolution of singularities of χ, what implies in particular the knowledge of all its closed singular points. One should eventually compute some extra closed points for auxiliary divisors.

(2) Find a basis for $\mathcal{L}(G)$ using the Brill–Noether method, what is our main task in the following section and what can be done effective with the aid of Hamburger–Noether expansions.

(3) Compute the order of a function at a rational point P and evaluate the function at this point when possible, what can be done from lazy parametrizations at the rational branch corresponding to P. More precisely, if $\phi = G/H$ is a quotient of homogeneous polynomials of the same degree in three variables,

and $(X(t), Y(t))$ is the rational parametrization obtained from the symbolic Hamburger–Noether expressions for the branch given by P, the order can be computed taking at P the corresponding local affine equation g/h of ϕ and doing the substitution

$$\frac{g(X(t), Y(t))}{h(X(t), Y(t))} = \frac{a_r t^r + \dots}{b_s t^s + \dots}$$

obtaining the order $r - s$ by lazy evaluation. Moreover, if ϕ is well-defined at P (what always happens in the applications to Coding Theory), then $r \geq s$ and $\phi(P) = a_s/b_s$.

(4) Find efficient decoding algorithms for AG codes. Roughly speaking, after solving the above problems (preprocessing) the complexity of decoding AG codes is reduced to that of solving linear systems. The idea is finding a set J of positions P_i containing the error positions, and then is J is "small enough", that is $\sharp J < d$, then the decoding is reduced to solving a certain linear system (see [11]).

We will see in the next section that everything can be reduced to deal with primitive rational parametrizations (introduced in [2]) above any singular point of the plane model (and eventually at some other points), and with Hamburger–Noether expansions, as a consequence.

An interesting case is when $G = mP$, P being an extra rational point of $\tilde{\chi}$. In this case the codes $C_m \doteq C_\Omega(D, mP)$ can be decoded by the majority scheme of the Feng and Rao algorithm [8], which is so far the most efficient method for the considered codes. In order to apply this decoding method, one has to fix for every non-negative integer i a function f_i in $\mathbb{F}(\tilde{\chi})$ with only one pole at P of order i for those values of i for which it is possible, i.e. for the integers in the Weierstrass semigroup $\Gamma = \Gamma_P$ of $\tilde{\chi}$ at P. In this way, the set $\{ f_i \mid i \leq m, \quad i \in \Gamma\}$ is actually a basis for $\mathcal{L}(mP)$. Thus, a second problem arises for AG codes: the computation of Γ and the corresponding functions f_i. This problem will be solved with two different approaches: either with the Brill–Noether algorithm, or with the the theory of approximate roots, due to Abhyankar and Moh [1], with the aid of some kind of "integral basis algorithm". On the other hand, the Goppa estimate can be substituted by the so-called Feng–Rao distance, depending on the combinatorics of the semigroup Γ, and whose computation is an additional (and non-trivial) problem (see the details in [8]).

We finally comment in few word why these codes are interesting. In fact, the so-called *main problem* in Coding Theory is that when one constructs a sequence of codes over $\mathbb{F}_q$ fixed

$$C_i \equiv [n_i, k_i, d_i]$$

with arbitrarily large length, that is

$$\lim_{i \to \infty} n_i = \infty$$

then in most examples one gets either

$$\lim_{i \to \infty} \frac{k_i}{n_i} = 0 \quad \text{or} \quad \lim_{i \to \infty} \frac{d_i}{n_i} = 0$$

what means that the performance of the codes tends to be bad when the length is arbitrarily large. Very few examples give both limits positive, and only with AG codes one systematically gets constructions whose asymptotic behaviour is excellent, in the sense that they go beyond the so-called Gilbert–Varshamov bound. The key point to do that is finding sequences of curves whose ratio n/g is as large as possible in an asymptotic way, that is achieving the so-called Drinfeld–Vlăduţ bound [5]. This problem is very hard, essentially because $n_i \to \infty$ implies $g_i \to \infty$, by using the Hasse–Weil bound, and it is connected with the problem of finding curves with many rational points (see [16] for further details).

2. An aproach to the Brill–Noether algorithm

For a given plane curve χ one can consider its normalization, that is the proper birational morphism

$$\mathbf{n} : \tilde{\chi} \to \chi$$

where $\tilde{\chi}$ is the curve obtained by gluing together the affine charts given by the normalization of the affine graded $\mathbb{F}$-algebras A_U for all affine charts U of χ. The curve $\tilde{\chi}$ can be regarded as the result of successive blowing-ups of all the closed points of χ which are singular, until we get a curve without singular points. This process can be represented by a combinatorial object called the *resolution forest* $\mathcal{T}_\chi$, consisting of one *weighted oriented tree* for each singular closed point of χ. Such a tree consists of the sequence of infinitely near points along with some weights giving the information about the successive field extensions and some kind of multiplicities. The upper extremal points of this forest are called branches.

The object $\mathcal{T}_\chi$ follows from the computation of Hamburger–Noether expansions, but what will be actually used in the sequel is just a primitive rational parametrization of each branch over every singular closed point of the curve χ, and the so-called adjunction divisor $\mathcal{A}$, which is nothing but the effective divisor given by the conductor ideal $\mathcal{C}_\chi$ on $\tilde{\chi}$. Let us write the adjunction divisor as

$$\mathcal{A} = \sum_{j=1}^{l} d_{q_j} q_j$$

where $q_1, \ldots, q_l$ denote the branches of $\mathcal{T}_\chi$. The coefficients d_{q_j} can be computed from $\mathcal{T}_\chi$ in different ways, but we will use a direct formula from the parametrizations of the branches. In fact, from the Hamburger–Noether expansion at a given rational branch q we can compute by lazy evaluation sufficiently many terms of a primitive rational parametrization $(X(Z_r), Y(Z_r))$ at q, and then the *Dedekind formula* allows us to compute the coefficient d_q of $\mathcal{A}$ at q as

$$d_q = ord_t\left(\frac{f_Y(X(t), Y(t))}{X'(t)}\right) = ord_t\left(\frac{f_X(X(t), Y(t))}{Y'(t)}\right)$$

where f is a local equation for χ at q (notice that either $X'(t) \neq 0$ or $Y'(t) \neq 0$). In

particular, one obtains the genus of χ by the formula $g = \dfrac{(m-1)(m-2)}{2} - \dfrac{1}{2}\deg \mathcal{A}$, where m denotes the degree of χ.

Our aim is to compute a basis for the vector space of finite dimension

$$\mathcal{L}(G) \doteq \{\phi \in \mathbb{F}(\tilde{\chi}) \mid (\phi) + G \geq 0\} \cup \{0\}$$

for an arbitrary $\mathbb{F}$-rational divisor G on $\tilde{\chi}$. A classical description of such space is derived from the Brill–Noether theorem as follows. Assume that χ is given by the homogeneous polynomial $F \in \mathbb{F}[X_0, X_1, X_2]$, and take a divisor G on $\tilde{\chi}$ that is rational over $\mathbb{F}$. Denote by $\mathcal{F}_n \subset \mathbb{F}[X_0, X_1, X_2]$ the set of forms of degree n, and consider $H_0 \in \mathcal{F}_n$ with $n \in \mathbb{N} \setminus \{0\}$, not divisible by F and satisfying

$$\mathbf{N}^* H_0 \geq G + \mathcal{A}$$

where $\mathbf{N} = i \circ \mathbf{n}$, $\mathbf{n}$ being the normalization of χ and i the embedding of χ in the projective plane, and where $\mathbf{N}^* H$ denotes the intersection divisor between the curve defined by the homogeneous polynomial H and χ. Then, the **Brill–Noether theorem** states that

$$\mathcal{L}(G) = \{\frac{h}{h_0} \mid H \in \mathcal{F}_n, \quad H \notin F \cdot \mathbb{F}[X_0, X_1, X_2] \text{ and } \mathbf{N}^* H + G \geq \mathbf{N}^* H_0\} \cup \{0\}$$

where h, h_0 denote respectively the rational functions H, H_0 restricted on χ.

This result allows us to compute a $\mathbb{F}$-basis of $\mathcal{L}(G)$ by means of the following

ALGORITH. (BRILL–NOETHER ALGORITHM)

For a given G, define $G_+ = \max\{G, 0\}$ and $J_+ = \mathcal{A} + G_+$.

(1) Take a large enough $n \in \mathbb{N}$ such that there exists $H_0 \in \mathcal{F}_n$ not divisible by F with $\mathbf{N}^* H_0 \geq J_+$.
(For instance $n > \max\left\{m - 1, \dfrac{m}{2} + \dfrac{deg\ J_+}{m} - \dfrac{3}{2}\right\}$, $m = deg\ F$ being the degree of χ).

(2) Compute a basis over $\mathbb{F}$ of the vector space

$$V = \{H \in \mathcal{F}_n \ : \ F|H \ \text{ or } \ \mathbf{N}^* H \geq J_+\} \cup \{0\}$$

(3) Compute a set of forms of $\mathcal{F}_n$ giving a basis over $\mathbb{F}$ of the vector space $V' = V/W$, where $W = \{A \in \mathcal{F}_n \ : \ F|A\} \cup \{0\}$.

(4) Choose $H_0 \in V \setminus W$ and compute the divisor $\mathbf{N}^* H_0$.

(5) Compute a set of forms of $\mathcal{F}_n$ being linearly independent over $\mathbb{F}$ which generate (modulo W) the vector space of forms H satisfying $\mathbf{N}^* H \geq \mathcal{A} + R$ (or $H = 0$), where $R \doteq \mathbf{N}^* H_0 - J$ and $J = \mathcal{A} + G$.

(6) If $\{H_1, \ldots, H_s\}$ is the basis obtained in **(5)** and for $i = 0, 1, \ldots, s$ we denote by $h_i \in \mathbb{F}(\chi)$ the functions H_i restricted to χ, then

$$\mathcal{B} = \left\{ \frac{h_1}{h_0}, \ldots, \frac{h_s}{h_0} \right\}$$

is a basis of $\mathcal{L}(G)$ over $\mathbb{F}$.

Notice that if we want this algorithm to be effective we must solve the following related problems:

(a) Compute the intersection divisor $\mathbf{N}^* H$ of a homogeneous polynomial H and the curve χ, that is, the value $\upsilon_Q(H)$ at every rational branch Q of χ. This can be solved by means of a suitable combination of Gröbner bases computation and primitive rational parametrizations of such branches (again from Hamburger–Noether expansions).

(b) For a given rational divisor J and a suitable $n \in \mathbb{N}$, compute a basis over $\mathbb{F}$ for the vector space

$$V \equiv V(J, n) = \{H \in \mathcal{F}_n \ : \ F | H \ \text{ or } \ \mathbf{N}^* H \geq J\} \cup \{0\}$$

where J is either $J_+ = \mathcal{A} + G_+$ or $\mathcal{A} + R$, according to the Brill–Noether algorithm. In the same way, one must describe the space W appearing in the steps **(3)** and **(5)**, and compute the respective quotients of vector spaces by using standard techniques from Linear Algebra.

We show now how to impose the "adjunction conditions" defining the above "spaces of adjoints" $V(J, n)$. Thus, one starts from the homogeneous polynomial $F(\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2) \in \mathbb{F}[\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2]$ defining the absolutely irreducible curve χ in the projective plane, and we have the data of a divisor $J = \mathcal{A} + R$ that is rational over $\mathbb{F}$, involving a finite number of rational branches of χ (singular or not) along with their corresponding coefficients. Notice that R can be either G_+ or $\mathbf{N}^* H_0 - \mathcal{A} - G$ in the above algorithm.

We first take a value of n such that there exists an adjoint of degree n satisfying the first step of the Brill–Noether algorithm. Then, we have to study the conditions imposed on a form $H \in \mathcal{F}_n$ of degree n by the inequality $\mathbf{N}^* H \geq \mathcal{A} + R$, R being an extra effective divisor.

Now, assume again that from the Hamburger–Noether expansions we have computed, for every branch q which is involved in the support of either $\mathcal{A}$ or R, sufficiently many terms of a primitive rational parametrization $(X(t), Y(t))$, and consider then the coefficients d_q and r_q of $\mathcal{A}$ and R at q. Then the local condition at q imposed to H by the inequality $\mathbf{N}^* H \geq \mathcal{A} + R$ is given by

$$\mathrm{or}\, d_t h(X(t), Y(t)) \geq d_q + r_q$$

h being the local affine equation of H in terms of the coordinates X, Y at the corresponding point P under q. A suitable number of steps of the lazy evaluation (which is

desirable to be determined a priori) suffices to describe the first $d_q + r_q$ monomials of the Taylor expansion of $h(X(t), Y(t))$ as a function of the indeterminate coefficients of H, whose vanishing gives the required linear conditions at q.

REMARK 1. One needs the computation of successive symbolic extensions of $\mathbb{F}$ for obtaining the parametrizations of conjugate branches of χ. However, one should be able to apply the above conditions on only one of the conjugate branches and apply somehow the Galois group to obtain conjugate conditions at the same point (or at the conjugate points). This would save substitutions of type $h(X(t), Y(t))$ (and the corresponding translations to the origin) and hence time of computations.

3. Computing Weierstrass semigroups

As we have told before, the decoding procedure of Feng and Rao is just based on the computation of a basis for $\mathcal{L}(lP)$, P being a rational point of $\tilde{\chi}$, in the way that if $l \in \Gamma_P$, the Weierstrass semigroup Γ_P consisting of the Weierstrass non-gaps at P, then such a basis is obtained by adding to a basis of $\mathcal{L}((l-1)P)$ a function f_l with a unique pole at P of order l. What we are going to do now is to show how one can compute the semigroup Γ_P and the functions f_l in a quite general situation by using the theory of adjoints.

There are two ways to proceed (see the details in [4]). One way is to compute the functions f_l for all $l \leq \tilde{l}$, $\tilde{l}$ being the largest non-gap that is needed in the computations with the considered one-point code. The other way is to compute first a generator system for the Weierstrass semigroup (namely, the Ap´ery system related to the minimum element e in the semigroup, since then calculations in Γ_P would be very nice), which can be assumed to be contained in the set of the first $g + e$ non-gaps. Then one saves the functions only for all l in such a system and compute the functions for all the other by using the arithmetic of the semigroup. In this case, $\tilde{l}$ denotes the largest generator.

In both situations, one must compute first a basis $\{h_1, \ldots, h_s\}$ of $\mathcal{L}(\tilde{l}P)$ over $\mathbb{F}$. Then we propose a triangulation method which works by induction on the dimension s as follows:

(1) By computing first the pole orders $\{-v_P(h_i)\}$ at P, assume that the functions $\{h_i\}$ are ordered in such a way that these pole orders are increasing in i. Thus, the maximum non-gap l' such that $l' \leq \tilde{l}$ is just $l' = -v_P(h_s)$.

(2) Since $-v_P(h_s) = l'$, we set $f_{l'} \doteq h_s$. If any other h_j satisfies the same condition, there exists a non-zero constant λ_j in $\mathbb{F}$ such that $-v_P(h_j - \lambda_j h_s) < l'$; then we change such functions h_j by $g_j \doteq h_j - \lambda_j h_s$ and set $g_k \doteq h_k$ for all the others. The result now is obviously another basis $\{g_1, \ldots, g_s\}$ of $\mathcal{L}(\tilde{l}P) = \mathcal{L}(l'P)$ over $\mathbb{F}$ but with only one function $g_s = f_{l'}$ whose pole at P has maximum order l'.

(3) Since the functions g_i are linearly independent over $\mathbb{F}$ and $-v_P(g_i) < l'$ for $i < s$, one has obtained a basis $\{g_1, \ldots, g_{s-1}\}$ of $\mathcal{L}(l''P)$ over $\mathbb{F}$, where l'' denotes the

largest non-gap such that $l'' < l'$. But now the dimension drops to $s-1$ and we can continue by induction.

As a consequence, the above procedure computes all the poles up to $\tilde{l}$ and also provides us with a function f_l for each non-gap $l \leq \tilde{l}$. The above algorithm is implemented by Farr´an and Lossen in the library `brnoeth` [7], distributed with the computer algebra system SINGULAR [9].

EXAMPLE 1. Let χ be the Klein quartic over $\mathbb{F}_2$ given by the equation

$$F(\mathrm{X},\mathrm{Y},\mathrm{Z}) = \mathrm{X}^3\mathrm{Y} + \mathrm{Y}^3\mathrm{Z} + \mathrm{Z}^3\mathrm{X} = 0$$

whose adjunction divisor is $\mathcal{A} = 0$, since χ is non-singular. We are going to compute the Weierstrass semigroup at $P = (0:0:1)$, which is not the only point at infinity. Thus, by means of the Brill–Noether algorithm we first compute a $\mathbb{F}_2$-basis of $\mathcal{L}(7P)$

$$\{h_1 = 1, h_2 = \frac{\mathrm{Z}}{\mathrm{Y}}, h_3 = \frac{\mathrm{Z}(\mathrm{Y}^2+\mathrm{YZ}+\mathrm{Z}^2)}{\mathrm{X}^2\mathrm{Y}}, h_4 = \frac{\mathrm{Z}^2(\mathrm{Y}+\mathrm{Z})}{\mathrm{X}^2\mathrm{Y}}, h_5 = \frac{\mathrm{Z}^3}{\mathrm{X}^2\mathrm{Y}}\}.$$

By using Hamburger–Noether expansions at P, one computes the pole order of these functions at such point

$$-\upsilon_P(h_1) = 0, -\upsilon_P(h_2) = 3, -\upsilon_P(h_3) = -\upsilon_P(h_4) = -\upsilon_P(h_5) = 7.$$

Thus, we take $f_7 = h_5$ and replace $h_4 = h_4 + h_5 = \dfrac{\mathrm{Z}^2}{\mathrm{X}^2}$ and $h_3 = h_3 + h_5 = \dfrac{\mathrm{Z}(\mathrm{Y}+\mathrm{Z})}{\mathrm{X}^2}$. Now the pole orders are

$$-\upsilon_P(h_1) = 0, -\upsilon_P(h_2) = 3, -\upsilon_P(h_3) = -\upsilon_P(h_4) = 6$$

and then we take $f_6 = h_4$. Thus, by replacing $h_3 = h_3 + h_4 = \dfrac{\mathrm{YZ}}{\mathrm{X}^2}$ we obtain now three different pole orders

$$-\upsilon_P(h_1) = 0, -\upsilon_P(h_2) = 3, -\upsilon_P(h_3) = 5$$

and we can stop. In particular, we have computed the Weierstrass semigroup, since we know the three Weierstrass gaps $\{1, 2, 4\}$ (note that the genus of χ is $g = 3$).

4. Semigroups at infinity

Let $\tilde{\chi}$ be again a non-singular projective algebraic curve defined over a finite field $\mathbb{F}$ and which is absolutely irreducible. Let χ be now a plane model for $\tilde{\chi}$, and assume that the hypothesis

(H1) χ has a unique branch at infinity

is satisfied, i.e. there exist a birational morphism

$$\mathbf{n} : \tilde{\chi} \to \chi \subseteq \mathbb{P}^2$$

and a line $L \subset \mathbb{P}^2$ defined over $\mathbb{F}$ such that $L \cap \chi$ consists of only one point P and $\tilde{\chi}$ has only one branch at P. Notice that both P and the branch at P are defined over the underlying finite field $\mathbb{F}$, since χ does. Thus there is only one point of $\tilde{\chi}$ over P, which will be denoted by $\overline{P}$.

Set $\tilde{\Upsilon} = \tilde{\chi} \setminus \{\overline{P}\}$ and $\Upsilon = \chi \setminus \{P\}$. One has the two following additive subsemigroups of $\mathbb{N}$:

$$\Gamma_P \doteq \{-\upsilon_{\overline{P}}(f) \mid f \in \mathcal{O}_{\tilde{\chi}}(\tilde{\Upsilon})\}$$
$$S_P \doteq \{-\upsilon_{\overline{P}}(f) \mid f \in \mathcal{O}_{\chi}(\Upsilon)\}$$

Notice that Γ_P is just the Weierstrass semigroup of $\tilde{\chi}$ at $\overline{P}$ and it contains S_P , but they are different unless the curve χ is non-singular in the affine part. Moreover, $\mathbb{N} \setminus \Gamma_P$ has g elements, g being the genus of $\tilde{\chi}$, and $\Gamma_P \setminus S_P$, which is also finite, will be computed below.

The first question to solve is the description of the semigroup S_P . In order to do that, we state the Abhyankar–Moh theorem, where the additional hypothesis

(H2) *char* $\mathbb{F}$ does not divide either *deg* χ or $e_P(\chi)$

is assumed. This result provides us with a set of generators for S_P with nice arithmetic properties (see for example [1]).

THEOREM 1 (ABHYANKAR–MOH). *Assumed that (H1) and (H2) are satisfied by* χ*, then there exist an integer* h *and a sequence of integers* $\delta_0, \ldots, \delta_h \in S_P$ *which generate* S_P *such that:*

(I) $d_{h+1} = 1$ *and* $n_i > 1$ *for* $2 \leq i \leq h$*, where* $d_i \doteq gcd(\delta_0, \ldots, \delta_{i-1})$ *for* $1 \leq i \leq h+1$ *and* $n_i \doteq d_i/d_{i+1}$ *for* $1 \leq i \leq h$.

(II) $n_i\delta_i$ *is in the semigroup generated by* $\delta_0, \ldots, \delta_{i-1}$ *for* $1 \leq i \leq h$.

(III) $n_i\delta_i > \delta_{i+1}$ *for* $1 \leq i \leq h-1$.

Such semigroups are a particular case of telescopic semigroups, and their main arithmetic property is that they are free, i.e. every $n \in S_P$ can be easily written in an unique way in the form

$$n = \sum_{i=0}^{h} \lambda_i \delta_i$$

with $\lambda_0 \geq 0$ and $0 \leq \lambda_i < n_i$ for $1 \leq i \leq h$.

Now we will say how to obtain these generators of S_P in a constructive way together with functions in $B \doteq \mathcal{O}_{\chi}(\Upsilon)$ having poles of order equal to those generators (and hence one will have functions in B with poles of order any element in S_P by using the arithmetic properties of such generators). For it, we need first the concept of approximate root.

DEFINITION 1. *Let S be a ring, $G \in S[\mathrm{Y}]$ a monic polynomial of degree e and $F \in S[\mathrm{Y}]$ a monic polynomial of degree n with $e|n$. Then G will be called an approximate b-th root of F if $deg\,(F - G^b) < n - e = e\,(b-1)$.*

Now the main remark is that for every monic polynomial $F \in S[\mathrm{Y}]$ of degree n and for every b divisor of n which is a unit in S, there exists a unique approximate b-th root of F, and it can be computed very efficiently by solving a triangular (non-linear) system.

Thus, let the affine plane model of the curve given by the equation

$$F = F(\mathrm{X}, \mathrm{Y}) = \mathrm{Y}^m + a_1(\mathrm{X})\,\mathrm{Y}^{m-1} + \ldots + a_m(\mathrm{X})$$

and suppose that $char\,\mathbb{F}$ satisfies the assumption of the Abhyankar–Moh theorem. Up to a change of variables in the form $\mathrm{X}' = \mathrm{X} + \mathrm{Y}^n$, $\mathrm{Y}' = \mathrm{Y}$, we can actually assume that $char\,\mathbb{F}$ does not divide the total degree m of χ. On the other hand, denote the approximate d-th root of F with respect to the coefficient ring $S = \mathbb{F}[\mathrm{X}]$ by $app(d, F)$. Thus, the so called *algorithm of approximate roots* computes the generators given by the Abhyankar–Moh theorem as follows:

$$F_0 = \mathrm{X}\,,\ \delta_0 = d_1 = m\,,\ F_1 = \mathrm{Y}\,,\ \delta_1 = deg_{\mathrm{X}} Res_{\mathrm{Y}}(F, F_1)$$

$$n > 1 \ \Rightarrow \ \left[\begin{array}{lcl} d_n & = & g\,c\,d\ (\delta_0, \delta_1, \ldots, \delta_{n-1}) \\ F_n & = & app(d_n, F) \\ \delta_n & = & deg_{\mathrm{X}} Res_{\mathrm{Y}}(F, F_n) \end{array}\right.$$

The procedure stops at the first $h \geq 1$ with $d_{h+1} = d_{h+2}$, what happens just when $d_{h+1} = 1$, since the point at infinity is unibranch. In fact, assumed that there is only one point at infinity, then the algorithm succeeds (i.e. gets to the end and the obtained semigroup satisfies the three properties of the Abhyankar–Moh theorem) if and only if that point is unibranch.

As a consequence, the generators of S_P given by the Abhyankar–Moh theorem and the corresponding functions can be easily computed in terms of approximate roots of F and resultants of polynomials. In particular, we can compute a rational function with an only pole at P of order n for every $n \in S_P$. In fact, if $n = \sum_{i=0}^{h} \lambda_i \delta_i$ with $\lambda_0 \geq 0$ and $0 \leq \lambda_i < n_i$ for $1 \leq i \leq h$, then $f_n = \prod_{i=0}^{h} F_i^{\lambda_i}$ is the searched function, where F_i are the polynomials which are obtained in the algorithm of approximate roots.

Now the remaining part of the method is the computation of $\Gamma_P \setminus S_P$ with the corresponding functions, what can be done effective by means of the following result (see [3]).

LEMMA 1 (TRIANGULATION). *Let A and B be the respective affine coordinate $\mathbb{F}$-algebras of $\tilde{\Upsilon}$ and Υ, i.e. $A = \mathcal{O}_{\tilde{\chi}}(\tilde{\Upsilon})$ and $B = \mathcal{O}_{\chi}(\Upsilon)$; then one has*

$$\sharp(\Gamma_P \setminus S_P) = dim_{\mathbb{F}}(A/B)$$

Proof. Take a basis $\{h_1, \ldots, h_l\}$ of A/B over $\mathbb{F}$. We show a *triangulation procedure* to find the values in $\Gamma_P \setminus S_P$ as well as functions which provide these values.

Set $B^i \doteq B + \mathbb{F}h_1 + \ldots + \mathbb{F}h_i$, for $0 \leq i \leq l$; we will proceed by induction, so let $0 \leq i < l$ and suppose we have found functions $g_1, \ldots, g_i$ which are linearly independent over $\mathbb{F}$ with

$$\begin{gathered}\Gamma_P^i \doteq S_P \cup \{-v_{\overline{P}}(g_1), \ldots, -v_{\overline{P}}(g_i)\} \subseteq \Gamma_P \\ -v_{\overline{P}}(g_j) \notin \Gamma_P^{i-1} \\ B + \mathbb{F}g_1 + \ldots + \mathbb{F}g_i = B^i\end{gathered}$$

Now look at h_{i+1}; if $-v_{\overline{P}}(h_{i+1}) \notin \Gamma_P^i$, then set $g_{i+1} = h_{i+1}$ and go on.

Otherwise, there exists $f \in B^i$ with

$$\begin{gathered}v_{\overline{P}}(h_{i+1}) = v_{\overline{P}}(f) \\ -v_{\overline{P}}(h_{i+1} - f) < -v_{\overline{P}}(h_{i+1})\end{gathered}$$

Thus we can repeat the process with $h_{i+1} - f$ replacing to h_{i+1}; since $h_{i+1} \notin B^i$, one obtains in a finite number of steps a function g_{i+1} such that

$$g_{i+1} \equiv h_{i+1} \pmod{B^i} \quad \text{and} \quad -v_{\overline{P}}(g_{i+1}) \notin \Gamma_P^i$$

At the end of the procedure l different elements in $\Gamma_P \setminus S_P$ will be added, and then $\sharp(\Gamma_P \setminus S_P) \geq dim_{\mathbb{F}}(A/B)$. The equality follows immediately from the formula $A = B^l = B + \mathbb{F}g_1 + \ldots + \mathbb{F}g_l$. □

REMARK 2. The only non-trivial part of this algorithm is how to obtain the initial "integral basis" to start the triangulation procedure. In principle, there are several ways to proceed:

i) Try to implement any of the known *integral basis algorithms*, basically based on generalizations of algorithms coming from algebraic number theory, which have been adapted to the function field of the curve. They actually compute a basis of A as B-module, but it is not very difficult to derive a basis of A/B over $\mathbb{F}$. However, the problem is interesting itself, since this algorithm does not work correctly for positive characteristic in most of the computer algebra systems (namely, MapleV or AXIOM). For example, in [13] a very efficient algorithm is developed for characteristic zero, by using Puiseux expansions with MapleV, but it seems that it has not been possible (as far as we know) to generalize it to any characteristic, probably because of the use of either such expansions or MapleV. We are now testing and speeding-up a new algorithm for SINGULAR with the use of Hamburger–Noether expansions instead, what solves the above problem.

ii) Try to take advantage of the *normalization* library of SINGULAR, which describes A as a function of some added ring variables, and try to eliminate somehow such variables. This has been also implemented in SINGULAR, though the algorithm mentioned in **i)** seems to be more efficient for plane curves, since the complexity lies on the computation of some Gröbner bases.

iii) Try to design a new algorithm with computes A in geometric terms from the Hamburger–Noether expansions of the plane curve. This should be possible to do in principle, since in some sense the data of A is equivalent to the resolution of singularities of the affine part of the curve, but the way to implement it by an algorithm is not clear for us yet. The idea is basically that

$$A/B = \bigoplus \overline{\mathcal{O}}_{\Upsilon,Q}/\mathcal{O}_{\Upsilon,Q}$$

where the sum runs over the set of all (closed) singular points in the affine part; thus, the Hamburger–Noether expansions would provide local bases at each Q, and the *Chevalley's principle* (independence of valuations) should allow us to glue up such local basis into a global one. This idea is still to be developed.

5. Examples and conclusions

The choice of the method to use in order to compute a Weierstrass semigroup depends on the situation. In fact, the Brill-Noether method works in a general situation, but the implementation is complicate and it does not give a nice description of the semigroup (namely, an Ap´ery system in order to calculate the Feng–Rao distance of certain one-point AG code).

On the other hand, the Abhyankar-Moh method gives such a description of Γ_P and the algorithm works in a very simple way (see [3]), but it requires some additional hypothesis on the plane model: it must have an only rational branch P at infinity which is defined over the base field $\mathbb{F}$ and the characteristic of $\mathbb{F}$ must not divide at the same time to the degree of the plane model and the multiplicity of P, what is frequently satisfied, but not always. If moreover the plane model has no other singular points at the affine part the curve, the algorithm of approximate roots directly yields the Weierstrass semigroup, and then the algorithm can be very easily implemented (for instance, such a programme takes a few lines in AXIOM code).

Anyway, the complement of this semigroup requires the previous computation of a certain integral basis (see remark 2), what is essentially equivalent to the desingularization of the affine part of the plane model, but what follows from such basis by means of a simple triangulation procedure, with the additional advantage of a nice description of the obtained semigroup. We will briefly illustrate these ideas with two examples.

EXAMPLE 2. Consider the affine plane curve $F(\mathrm{X},\mathrm{Y}) = \mathrm{Y}^9+\mathrm{Y}^8+\mathrm{XY}^6+\mathrm{X}^2\mathrm{Y}^3+\mathrm{Y}^2+\mathrm{X}^3$ defined over $\mathbb{F}_2$, with only one branch at infinity $P = (1:0:0)$. The algorithm of approximate roots yields

$$F_0 = \mathrm{X}\,,\ \delta_0 = d_1 = 9\,,\ F_1 = \mathrm{Y}$$

$$\delta_1 = deg_{\mathrm{X}} Res_{\mathrm{Y}}(F,\mathrm{Y}) = 3\,,\ d_2 = g\,c\,d\,(9,3) = 3$$

$$F_2 = app(3,F) = \mathrm{Y}^3+\mathrm{Y}^2+\mathrm{Y}+\mathrm{X}+1$$

$$\delta_2 = deg_{\mathrm{X}} Res_{\mathrm{Y}}(F,F_2) = 8\,,\ d_3 = g\,c\,d\,(9,3,8) = 1$$

thus $h = 2$ and $S_P = \langle 9, 3, 8\rangle$.

On the other hand, according to the lemma 1, take a $\mathbb{F}_2$-basis for A/B

$$h_1 = \frac{Y(1+Y^6)}{X+Y^3} \quad h_2 = \frac{Y(1+Y^6)}{(X+Y^3)(Y^2+Y+1)}$$

$$h_3 = \frac{X^2+Y^6}{Y^2+Y+1} \quad h_4 = \frac{Y^2(1+Y^3)(Y^2+Y+1)}{X+Y^3}$$

The values at P of this functions are $-v_P(h_1) = 13 \notin S_P$, $-v_P(h_2) = 7 \notin \Gamma_P^1$, $-v_P(h_3) = 10 \notin \Gamma_P^2$ and $-v_P(h_4) = 13 \in \Gamma_P^3$. Then change h_4 by

$$g_4 = h_4 + h_1 = \frac{Y(1+Y^3)(Y^2+Y+1)}{X+Y^3}$$

and now $-v_P(g_4) = 10 \in \Gamma_P^3$, so still one has to take the function

$$g_4 = h_4 + h_1 + h_3 = \frac{Y(1+Y^3)(Y^4+Y^2+1) + (X+Y^3)^3}{(X+Y^3)(Y^2+Y+1)}$$

and now $-v_P(g_4) = 4 \notin \Gamma_P^3$. Hence, the Weierstrass semigroup at P is

$$\Gamma_P = \{0, 3, \mathbf{4}, 6, \mathbf{7}, 8, 9, \mathbf{10}, 11, 12, \mathbf{13}, 14, \ldots\}$$

Unfortunately, there are examples where this method cannot be applied (consider again the example 1), and then the Brill-Noether is the only way to compute Γ_P and the functions, even though it does not help (in general) to compute the Feng–Rao distance.

We finally remark that the philosophy of the Abhyankar–Moh theorem and the computation of an integral basis is somehow the same as the adjunction theory, that is, searching for polynomial curves passing through some points with designed multiplicities. This is actually what is done inside the integral basis algorithm given in [13], taking the *discriminant* instead of the *conductor*. On the other hand, the approximate roots computed by the Abhyankar–Moh method are curves with maximal contact order at the only point at infinity (see [15]).

References

[1] ABHYANKAR S.S., *On the semigroup of a meromorphic curve*, Intl. Symp. on Algebraic Geometry, Kyoto (1977), 249–414.

[2] CAMPILLO A. AND CASTELLANOS J., *Curve singularities*, preprint, Univ. Valladolid (1997).

[3] CAMPILLO A. AND FARRÁN J.I, it Computing Weierstrass semigroups and the Feng–Rao distance from singular plane models, Finite Fields and their Applications **6** (2000), 71–92.

[4] CAMPILLO A. AND FARRÁN J.I, *Symbolic Hamburger–Noether expressions of plane curves and applications to AG codes*, Mathematics of Computation **71** (2002), 1759–1780.

[5] DRINFELD V.G. AND VLĂDUŢ S.G., *Number of points of an algebraic curve*, Funktsional'-nyi Analiz i Ego Prilozhenia **17** (1983), 53–54.

[6] DUVAL D., *Diverses questions relatives au calcul formel avec des nombres algbriques*, Ph.D. thesis, Universit´e de Grenoble, Grenoble 1987.

[7] FARRÁN J.I. AND LOSSEN CH., "`brnoeth.lib`", A SINGULAR 2.0 library for the Brill–Noether algorithm, Weierstrass semigroups and AG codes (*Software* and *Reference Manual*, 16 pp.), in SINGULAR 2.0, A Computer Algebra System for Polynomial Computations, G.-M. Greuel, G. Pfister y H. Sch¨onemann, Centre for Computer Algebra, University of Kaiserslautern (2001), available via `http://www.singular.uni-kl.de/`.

[8] FENG G.L. AND RAO T.R.N., *Decoding algebraic-geometric codes up to the designed minimum distance*, IEEE Trans. Inform. Theory **39** (1993), 37–45.

[9] GREUEL G.-M., PFISTER G. AND SCHÖNEMANN H., "SINGULAR 2.0.3", *A computer algebra system for polynomial computations*, Centre for Computer Algebra, University of Kaiserslautern (2002), available via `http://www.singular.uni-kl.de/`.

[10] HACHÉ G. AND LE BRIGAND D., *Effective construction of Algebraic Geometry codes*, IEEE Trans. Inform. Theory **41** (1995), 1615–1628.

[11] HØHOLDT T. AND PELLIKAAN R., *On the decoding of algebraic-geometric codes*, IEEE Trans. Inform. Theory **41** (1995), 1589–1614.

[12] HØHOLDT T., VAN LINT J.H. AND PELLIKAAN R., *Algebraic Geometry codes*, in: "Handbook of Coding Theory" vol.1, (Eds. Pless V., Huffman W.C. and Brualdi R.A.), Elsevier, Amsterdam 1998, 871–961.

[13] VAN HOEIJ M., *An algorithm for computing an integral basis in an algebraic function fi eld*, J. Symbolic Computation **18** (1994), 353–363.

[14] VAN LINT J.H., *Introduction to Coding Theory*, Springer-Verlag, 1982.

[15] PINKHAM H., *Séminaire sur les singularités des surfaces (Demazure-Pinkham-Teissier)*, cours donn´e au Centre de Math. de l'École Polytechnique, (1977-1978).

[16] TSFASMAN M.A. AND VLĂDUŢ S.G., *Algebraic-geometric codes*, Math. and its Appl. vol. 58, Kluwer Academic Pub., 1991.

AMS Subject Classification: 14Q05, 14G50, 68W30, 94B27.

Antonio CAMPILLO, Departamento de Álgebra, Geometr´ıa y Topologia, Facultad de Ciencias, Universidad de Villadolid, 47005 Villadolid, SPAIN
e-mail: `campillo@cpd.uva.es`

Jos´e Ignacio FARRÁN, Departamento de de Matimatica Aplicada a la Ingenier´ıa, Universidad de Villadolid, 47011 Villadolid, SPAIN
e-mail: `ignfar@wmatem.eis.uva.es`

Lavoro pervenuto in redazione il 16.04.2003.